ELMIRA HAJIMANI

INTELLIGENT SUPPORT SYSTEM FOR CVA DIAGNOSIS BY CEREBRAL COMPUTERIZED TOMOGRAPHY

UNIVERSIDADE DO ALGARVE

Faculdade de Cîencias e Tecnologia

2016

ELMIRA HAJIMANI

INTELLIGENT SUPPORT SYSTEM FOR CVA DIAGNOSIS BY CEREBRAL COMPUTERIZED TOMOGRAPHY

Doutoramento em Engenheira Informática

(Especialidade em Inteligência Artificial)

Trabalho efeuado sob a orientação de:

António Eduardo de Barros Ruano e Maria da Graça Ruano

UNIVERSIDADE DO ALGARVE

Faculdade de Cîencias e Tecnologia

2016

INTELLIGENT SUPPORT SYSTEM FOR CVA DIAGNOSIS BY CEREBRAL COMPUTERIZED TOMOGRAPHY

Declaração de autoria de trabalho

Declaro ser a autora deste trabalho, que é original e inédito. Autores e trabalhos consultados estão devidamente citados no texto e constam da listagem de referências incluída.

_____

Elmira Hajimani

To my love, Hamid

To my parents, Zari and Majid

To my sister, Armita

Acknowledgement

First and foremost, I would like to express my deepest thanks and appreciation to my supervisors, Prof. António Ruano and Prof. Graça Ruano, for having provided me with their expertise, valuable advice, direction and suggestions during the past four years. Without their inspiration, support and encouragement, this dissertation would have been impossible.

I would particularly like to acknowledge the Erasmus Mundus EMIIY scholarship program for its financial support of this thesis. I also would like to express my gratitude to Prof. Hamid Shahbazkia as the coordinator of EMIIY scholarship program for all his kind supports.

I wish to acknowledge Dr. Luis Cerqueira, from Centro Hospitalar de Lisboa Central, Portugal, for marking the CT exams, Mr. Sergio Silva, in CSI Laboratory, for his precious help in solving implementation issues and Dr. Carina Ruano for her helpful suggestions while we were designing our web based tool for marking CVA lesions in CT images. I am also grateful to the staff of mobility office at the University of Algarve for all their supports.

My sincere appreciation goes to my dear friend, Dr. Eslam Nazemi, for his continuous encouragement and support during the past 10 years of my academic life. Dr. Nazemi brought me to the world of academic research and supervised my B.Sc. and M.Sc. dissertations.

I would like to express my deepest gratitude to my beloved husband and my great colleague, soon to be Dr. Hamid Reza Khosravani, for his endless love, unconditional support, for all the joyful moments we built together and for long hours of scientific discussions.

Last but not least, my special thanks goes to my beloved parents and my sister who taught me how to think and how to love.

Abstract

The Cerebral Vascular Accident (CVA) is one of the major causes of death in USA and developed countries, immediately following cardiac diseases and tumors. The increasing number of CVA's and the requirement of short time diagnosis to minimize morbidity and mortality encourages the development of computer aided diagnosis systems. Early stages of CVA are often undetected by human eye observation of Computer Tomographic (CT) images, thus incorporation of intelligent based techniques on such systems is expected to highly improve their performance.

This thesis presents a Radial Basis Functions Neural Network (RBFNN) based diagnosis system for automatic identification of CVA through analysis of CT images. The research hereby reported included construction of a database composed of annotated CT images, supported by a web-based tool for Neuroradiologist registration of his/her normal or abnormal interpretation of each CT image; in case of an abnormal identification the medical doctor was indicted by the software application to designate the lesion type and to identify the abnormal region on each CT's slice image.

Once provided the annotated database each CT image processing considered a pre-processing stage for artefact removal and tilted images' realignment followed by a feature extraction stage.

A large number of features was considered, comprising first and second order pixel intensity statistics as well as symmetry/asymmetry information with respect to the ideal mid-sagittal line of each image.

The policy conducted during the intelligent-driven image processing system development included the design of a neural network classifier. The architecture was determined by a Multi Objective Genetic Algorithm (MOGA) where the classifier structure, parameters and image features (input features) were chosen based on the use of different (often conflicting) objectives, ensuring maximization of the classification precision and a good generalization of its performance for unseen data

Several scenarios of choosing proper MOGA's data sets were conducted. The best result was obtained from the scenario where all boundary data points of an enlarged dataset were included in the MOGA training set.

Confronted with the NeuroRadiologist annotations, specificity values of 98.01% and sensitivity values of 98.22% were obtained by the computer aided system, at pixel level. These values were achieved when an ensemble of non-dominated models generated by MOGA in the best scenario, was applied to a set of 150 CT slices (1,867,602 pixels).

Present results show that the MOGA designed RBFNN classifier achieved better classification results than Support Vector Machines (SVM), despite the huge difference in complexity of the two classifiers. The proposed approach compares also favorably with other similar published solutions, both at lesion level specificity and at the degree of coincidence of marked lesions.

Keywords: Neural Networks; Symmetry features; Multi-Objective Genetic Algorithm; Intelligent support systems; Cerebral Vascular Accident.

Resumo

Os Acidentes Vasculares Cerebrais (AVC) são uma das maiores causas de morte nos EUA e em países desenvolvidos, imediatamente a seguir a condições cardícas e a tumores. O aumento do número de AVCs e o requisito de um diagnóstico rápido, necessário para minimizar a morbilidade e a mortalidade, encoraja o desenvolvimento de sistemas de ajuda ao diagnóstico. Os CVAs, num estado inicial não conseguem muitas vezes serem detetados pelo observação humana de imagens de Tomografia Computorizada (TC); a incorporação de técnicas baseadas em inteligência computacional poderá contribuir para melhorar a performance desses sistemas.

Esta tese apresenta um sistema de diagnóstico baseado em Redes Neuronais de Função de Base Radial (RNFBR) para a identificação automática de AVCs através da análise de imagens de TC. A investigação reportada nesta Tese incluiu a construção de uma base de dados de imagens de TC anotadas, suportadas por uma ferramenta baseada na Web que permite que os Neuroradiologistas registem as lesões por si identificadas, bem como o tipo de lesão e a região do cérebro onde a mesma se localiza.

Após criação da base de dados anotada, as imagens de TC são submetidas a um passo de pré-processamento, incluindo remoção de artefactos e realinhamento das imagens inclinadas,de modo a poder-se posteriormente proceder à extração de características.

Um grande número de características de entrada são considerados nesta abordagem, compreendendo estatísticas de primeira e segunda ordem dos pixéis da imagem, , bem como informações de simetria ou assimetria em relação à linha média sagital ideal.

Para a conceção de um classificador da rede neuronal, é utilizada uma abordagem baseada num Algoritmo Genético Multi-Objectivo (MOGA) para determinar a arquitetura do classificador, os seus parâmetros, bem como as características de entrada utilizadas, utilizando para tal diferentes objetivos, muitas vezes conflituosos entre si, aumentando a precisão de classificação sem no entanto comprometer a sua generalização para dados não consierados no projeto da rede.

Foram realizados vários cenários em MOGA. O melhor resultado foi obtido do cenário no qual no conjunto de treino de MOGA foram incorporados os vértices do fecho convexo de um conjunto alargado de pixéis.

Confrontando com as anotações do Neuroradiologista, foi sistema obteve valores de especificidade de 98,01% e sensibilidade de 98,22%, ao nível do pixel. Estes resultados foram obtidos por um conjunto de modelos não-dominados gerados pelo MOGA no melhor cenário, num conjunto de 150 imagens TC (1,867,602 pixels).

Esta abordagem compara-se muito favoravelmente com outras soluções semelhantes publicadas, tanto em especificidade ao nível da lesão, como no grau de coincidência de lesões marcadas. Comparando os resultados da classificação neuronal com Máquinas de Vetor de Suporte (SVM), é evidente que, apesar da enorme complexidade do modelo SVM, a precisão do modelo neuronal é superior à do modelo SVM.

Palavras-chave: Redes Neuronais; Algoritmo Genético Multi-Objectivo; Acidentes Vasculares Cerebrais

Contents

List of Tables

List of Figures

List of Algorithms

List of Acronyms

ABR	Area of Bleeding Region
ADH	Amplitude Distribution Histograms
AER	Area of Edema Region
AMM	Adaptive Mixtures Method
ANNs	Artificial Neural Networks
BP	Back Propagation
CAD	Computer Aided Diagnosis
CAROI	Circular Adaptive Region of Interest
CLBP	Completed Local Binary Pattern
CLBP_C	Completed Local Binary Pattern - Center
CLBP_M	Completed Local Binary Pattern - Magnitude
CLBP_S	Completed Local Binary Pattern -Sign
CRLBP	Completed Robust Local Binary Pattern
CSF	Cerebrospinal Fluid
CT	Computerized Tomography
CVA	Cerebral Vascular Accident
DFP	Davidson–Fletcher–Powell
DOT	Diffuse Optical Tomography
EDA	Exploratory Data Analysis
EM	Expectation Maximization

ESMF	Edge-based Selective Median Filter
FCM	Fuzzy C-Means
fMRI	functional Magnetic Resonance Imaging
FN	False Negative
FP	False Positive
GA	Genetic Algorithm
GLCM	Gray Level Co-occurrence Matrix
GM	Gray Matter
HU	Hounsfield Units
IDM	Inverse Difference Moment
k-NN	k-Nearest Neighbors
LBP	Local Binary Pattern
LM	Levenberg-Marquardt
LTP	Local Ternary Pattern
LRHGE	Long Run High Gray Level Emphasis
LRLGE	Long Run Low Gray Level Emphasis
MLP	Multi-Layer Perceptron
MOGA	Multi Objective Genetic Algorithm
MR	Magnetic Resonance
MRI	Magnetic Resonance Imaging
PACS	Picture Archiving and Communication System

PCC	Pearson Correlation Coefficient
PET	Positron Emission Tomography
PSM	Power Spectrum Method
RBFNN	Radial Basis Functions Neural Network
SMOTE	Synthetic Minority Over-sampling Technique
SMOTE-N	Synthetic Minority Over-sampling Technique Nominal
SMOTE-NC	Synthetic Minority Over-sampling Technique Nominal Continuous
SR1	Symmetric Rank one
SRHGE	Short Run High Gray Level Emphasis
SRLGE	Short Run Low Gray Level Emphasis
SUS	Stochastic Universal Sampling
SVM	Support Vector Machine
WM	White Matter

1. Introduction

The Cerebral Vascular Accident is the third cause of death in USA, immediately following cardiac diseases and tumors. In the USA, from the 700.000 CVA cases, 600.000 are ischemic and 100.000 hemorrhagic. 175.000 CVAs are fatal, and the rest reduces patients' morbidity, involving additional expenses for the National Health Systems [1]. In Portugal the CVA is the first cause of death, and several studies point out a prognosis of more than 80 CVA occurrences per day for the next 10 years.

Computerised Tomography (CT) is one of the imaging equipments for diagnosis which benefited more from technological improvements. Because of that, and due to the quality of the diagnosis produced, it is one of the most used equipments in clinical applications. For CVA diagnosis, CT is the elected imaging equipment, as the majority of hospitals have CTs, but no Magnetic Resonance (MR) equipment. In those where MR is available, it is typically used only at 1/3 of the day, due to the need for specialized personnel, which is lacking. However, within the first few hours after symptom onset, the interpretation of CT images can be difficult due to the inconspicuousness of the lesions. Quick diagnosis becomes even more difficult when the CT technician is not familiar with image post-processing protocols.

These facts constitute the motivation to create an intelligent application capable of assisting the CT technician on triggering a pathologic occurrence and enabling a better performance of CVA detection.

1.1 Objectives

This PhD aimed to construct a prototype of an automatic support system for CVA diagnosis in CTs, by:

1. Providing a platform to enlarge the existing "storage" of diagnosed CT scans, and implementing a proper database.

2. To account for the variability found in CT scans, carefully reviewing the features used for classification.

3. Using the available multi-objective evolutionary methodology for designing neural classifiers to select the most relevant features [2]. The referred system allows, besides designing the classifier topology and determining its parameters, to perform feature selection, according to different objectives and priorities. In contrast with the existing approaches found in the literature, where features are taken solely from one specific category (i.e., first order or second order statistics), our system will pick up the most important features from the union of these sets as well as incorporating some symmetry features.

4. Performing medical validation of the prototype system.

1.2 Major contributions

A web-based tool was developed [3] in order to be able to register the opinion of Neuroradiologists for each CT image. Using this tool, a database of CT images was created for Neuroradiologists to remotely analyze and mark the images either as normal or abnormal. For the abnormal ones, the doctor is able to identify the lesion type and the abnormal region on each CT's slice image.

A thorough review has been done on the features used by other works (i.e., Please refer to section 3.7). Table 5.1 provide a list of features that are used in this work. These features can be grouped into three main categories:

a) First order statistics which estimate properties of individual pixel values, ignoring the spatial interaction between image pixels.

b) Second order statistics which estimate properties of two pixel values occurring at specific locations relative to each other.

c) Features related with differences in symmetry across the ideal midsagittal plane.

To our knowledge, none of existing classifiers learn about the asymmetry caused by lesions in intracranial area. In this work, a group of symmetry features that were proposed in [4], are going to be used along with other statistical features to add the ability of detecting asymmetries (with respect to ideal mid-sagittal line) to the designed classifier.

Several experiments were conducted in MOGA and the corresponding obtained models were evaluated using a set of 1,867,602 pixels. In some experiments, active learning approach is applied

to design the subsequent experiment. To construct the dataset of the conducted experiments, Approxhull [5, 6] is used to incorporate convex points in the training set. This will help MOGA to see the whole range of the data where the classifier is going to be used.

The best result is obtained from an ensemble of preferable models of the experiment whose training set contained all convex points of the 1,867,602 pixels together with some random normal and abnormal pixels. Values of specificity of 98.01% (i.e., 1.99 % False Positive) and sensitivity of 98.22% (i.e., 1.78% False Negative) were obtained at pixel level, in a set of 150 CT slices (1,867,602 pixels).

Comparing the classification results with SVM, shows that, despite the huge complexity of SVM model, the accuracy of the ensemble of preferable models is superior to that of SVM model.

The present approach compares favorably with other similar (although with not the same specifications) published approaches [7, 8], achieving, on the one hand, improved sensitivity at lesion level, and, on the other hand, superior average difference and degree of coincidence between lesions marked by the doctor and marked by the automatic system.

As a result of the research work developed under this PhD thesis the following publications were produced:

• E. Hajimani, M. G. Ruano, and A. E. Ruano, "An Intelligent Support System for Automatic Diagnosis of Cerebral Vascular Accidents from Brain CT Images," submitted to Computer Methods and Programs in Biomedicine (May 2016)

• M. G. Ruano, E. Hajimani, and A. E. Ruano, "A Radial Basis Function Classifier for the Automatic Diagnosis of Cerebral Vascular Accidents," presented at the Global Medical Engineering Physics Exchanges/Pan American Health Care Exchanges (GMEPE / PAHCE), Madrid, Spain, 2016.

• E. Hajimani, M. G. Ruano, and A. E. Ruano, "MOGA design for neural networks based system for automatic diagnosis of Cerebral Vascular Accidents," in 9th IEEE International Symposium on Intelligent Signal Processing (WISP), 2015, pp. 1-6.

• E. Hajimani, A. Ruano, and G. Ruano, "The Effect of Symmetry Features on Cerebral Vascular Accident Detection Accuracy," presented at the RecPad 2015, the 21th edition of the Portuguese Conference on Pattern Recognition, Faro, Portugal, 2015.

• E. Hajimani, C. A. Ruano, M. G. Ruano, and A. E. Ruano, "A software tool for intelligent CVA diagnosis by cerebral computerized tomography," in 8th IEEE International Symposium on Intelligent Signal Processing (WISP), 2013, pp. 103-108.

1.3 Thesis structure

This thesis is organized in 6 chapters. Chapter 2 gives a brief overview on the theoretical background that is needed to develop this work. This includes a review on artificial neural networks and learning algorithms, Support Vector Machines, Multi-Objective Genetic Algorithm, active learning, Approxhull and existing solutions for dealing with the challenge of classifying imbalanced datasets.

Chapter 3 gives the necessary background information about Cerebral Vascular Accident, medical imaging techniques and the state of the art for automatic segmentation of lesions from brain tissues in medical images. A review on textural feature extraction methods is also presented in this chapter.

To train, test and validate the neural network models for classifying pathologic areas within brain CT images, it is necessary to acquire the opinion of Neuroradiologists, and use it as the gold standard. Chapter 4 presents our developed web-based tool to collect this information in an accurate and convenient way. Having used our developed web-based tool to obtain the opinion of the Neuroradiologist about existing CT images, we are now able to construct our dataset.

Chapter 5 starts with describing how we produced our datasets from the CT images previously marked by Neuroradiologist. To obtain the best possible RBFNN classifier, several scenarios were conducted in MOGA which are explained in chapter 5. This chapter also shows the results obtained, including visualization of the estimated abnormal regions in CT images, and compares the proposed approach with support vector machines and two other Computer Aided Diagnosis (CAD) systems. Finally, a discussion on the discrimination power of the most frequent features in preferable models of the best scenario is performed.

Conclusion and future works are presented in chapter 6.

2. Intelligent systems background

This chapter aims to review the basic concepts of intelligent data driven modeling techniques that are used for developing the presented intelligent support system for automatic diagnosis of Cerebral Vascular Accident from brain Computed Tomography images. Intelligent data driven modeling can be thought as the use of a collection of approaches, mainly artificial neural networks, fuzzy rule-based systems and evolutionary algorithms to build models, calibrate them and optimize their structures. For building models, data driven approaches use available data to develop relationships between the input and output variables involved in the actual process. The presented work uses a combination of neural networks and genetic algorithms methods to build the proposed system.

The chapter is organized as follows: Section 2.1 gives a brief overview on Artificial Neural Networks. In this section, after providing a taxonomy of existing neural network topologies, a more detailed discussion is done on Multi-Layered Perceptron and Radial Basis Functions Neural Networks. An overview on different learning algorithms with the focus on supervised algorithms is done afterwards. Section 2.1 continues with the presentation of three learning strategies for Radial Basis Functions Neural Networks. Termination criteria for the training process are discussed in the last part of section 2.1. A brief description of Support Vector Machines is presented in section 2.2 since we have compared our work with this method in later chapters. Multi-Objective Genetic Algorithm, as a framework to determine the architecture of the classifier, its corresponding parameters and input features according to the multiple objectives imposed and their corresponding restrictions and priorities, is discussed in section 2.3. Section 2.4 discusses active learning as a way of choosing the most informative data samples from a pool of data. Approxhull, as a data selection approach for selecting the most suitable data to be incorporated in the training set, is presented in section 2.5. Section 2.6 overviews potential solutions for dealing with the challenge of classifying imbalanced datasets.

2.1 Artificial Neural Networks

Artificial Neural Networks were initially developed as an attempt to mimic the behavior of human brain. As we know, human brain can be divided into regions, each of which specialized in different functions. But the interesting point is that one region of the brain has the capacity to process information of a modality normally associated with another region [9, 10]. This fact came from the

experiments in which neuroscientists rerouted retinal signals to a part of the brain which is responsible to process the auditory signals and concluded that the auditory cortex was learning how to process the visual signals. In another similar experiment, the retinal signals were rerouted, this time, to the somatosensory cortex which is responsible to process the sense of touch. The result was the same; after a while the somatosensory neurons were learning how to process new types of signals. Artificial Neural Networks are also providing the capability of designing algorithms that are applicable to many different areas just by tuning some parameters based on the corresponding context. These algorithms can be used for statistical analysis and data modeling in many different areas such as medical diagnosis, financial market prediction, energy consumption, face, speech and text recognition and many more. Fig. 2.1 provides a taxonomy of neural network architectures [11].

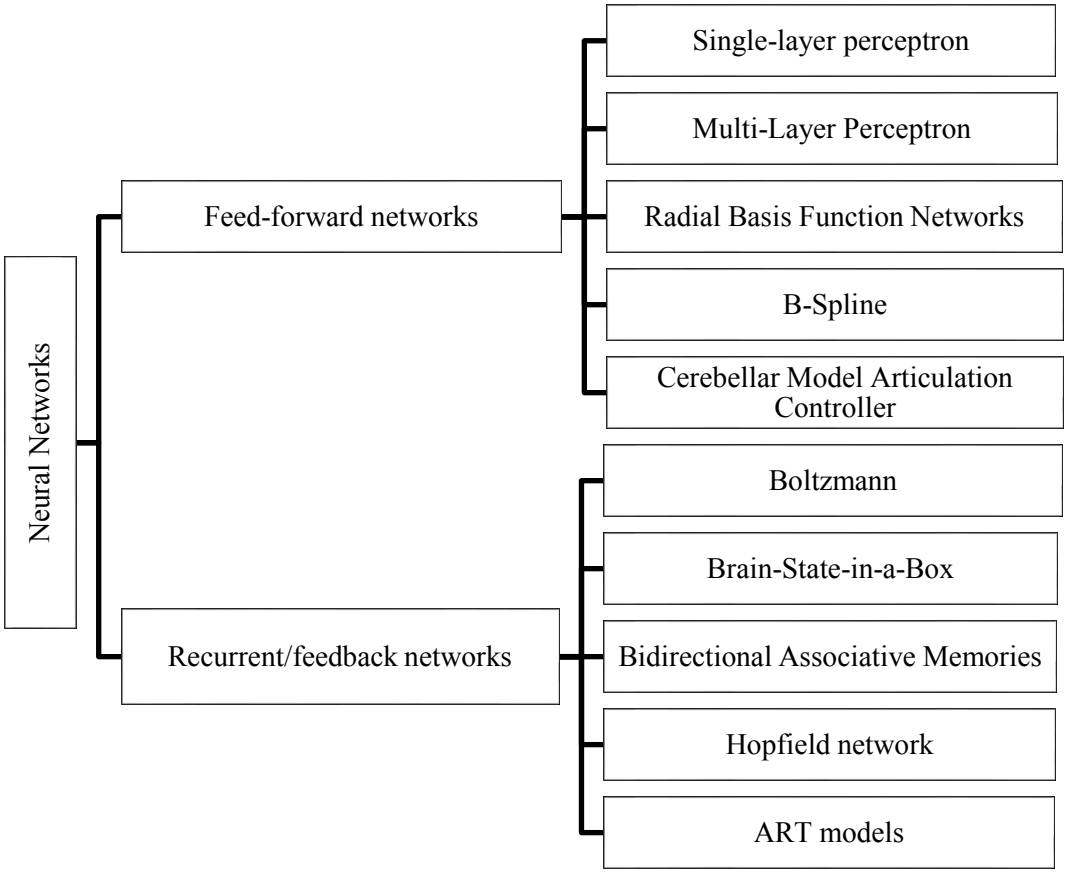


Fig. 2.1 A taxonomy of neural network architectures [11]

In the following subsections we are focusing on Multi-Layer Perceptron and Radial Basis Function Networks.

2.1.1 Multi-Layer Perceptron

Multi-Layer Perceptron (MLP) is one of the most well-known models that can be used for solving classification, pattern recognition and forecasting problems. As it can be seen in Fig. 2.2, a set of sensory units constitute the input layer. Input features are then passed to neurons belonging to one or more hidden layers. These hidden neurons with smooth (i.e., differentiable everywhere), nonlinear activation functions help the network to learn meaningful relations from the input vector to the output vector. Bounded functions like sigmoid or hyperbolic tangent are used as the activation function of the neurons in hidden layers. Eq. 2.1 shows one example (a sigmoidal function) of activation functions that can be used for the neurons in hidden layers.

$$\varphi_i^l(\mathbf{w}^l, \mathbf{x}) = \frac{1}{1+e^{-\left(b_i^l + \sum_{j=1}^{n_{l-1}} w_{i,j}^l \varphi_j^{l-1}(\mathbf{w}^{l-1}, \mathbf{x})\right)}} \tag{2.1}$$

Where φ_i^l is the output of the i^{th} neuron at hidden layer l (containing n_l hidden neurons), and b_i^l is its bias. If $l = 1$ (the first hidden layer) then $\varphi_j^{l-1} = x_j$, i.e., it is the j^{th} input. $w_{i,j}^l$ denotes the weight connecting the j^{th} neuron in layer l-1 with the i^{th} neuron in layer l.

The bias can be seen as another weight connecting the i^{th} neuron with a fixed value of 1. In this case, the last equation can be expressed as:

$$\varphi_i^l(w^l, \mathbf{x}) = \frac{1}{1+e^{-\left(w_{i,n_l+1}^l + \sum_{j=1}^{n_{l-1}} w_{i,j}^l \varphi_j^{l-1}(\mathbf{w}^{l-1},\ \mathbf{x})\right)}} \tag{2.2}$$

The output of the network is a linear combination of activation functions of the last hidden layer:

$$y_o = b_o^L + \sum_{k=1}^{n_L} w_{o,k}^L \varphi_k^L \tag{2.3}$$

In the last equation y_o represents the o^{th} output, and L is the number of hidden layers.

Using the same reasoning as above, equation (2.3) can be represented as:

$$y_o = w_{o,n_L+1}^L + \sum_{k=1}^{n_L} w_{o,k}^L \varphi_k^L \tag{2.4}$$

MLP has a fully connected structure [12] which means each neuron in any layer is connected to all neurons of the previous layer by a weighted link. The number of hidden layers and neurons in

those layers should be selected in a way that helps the training algorithm to converge to its optimum, while avoiding overmodelling due to a larger number of neurons than needed [13].

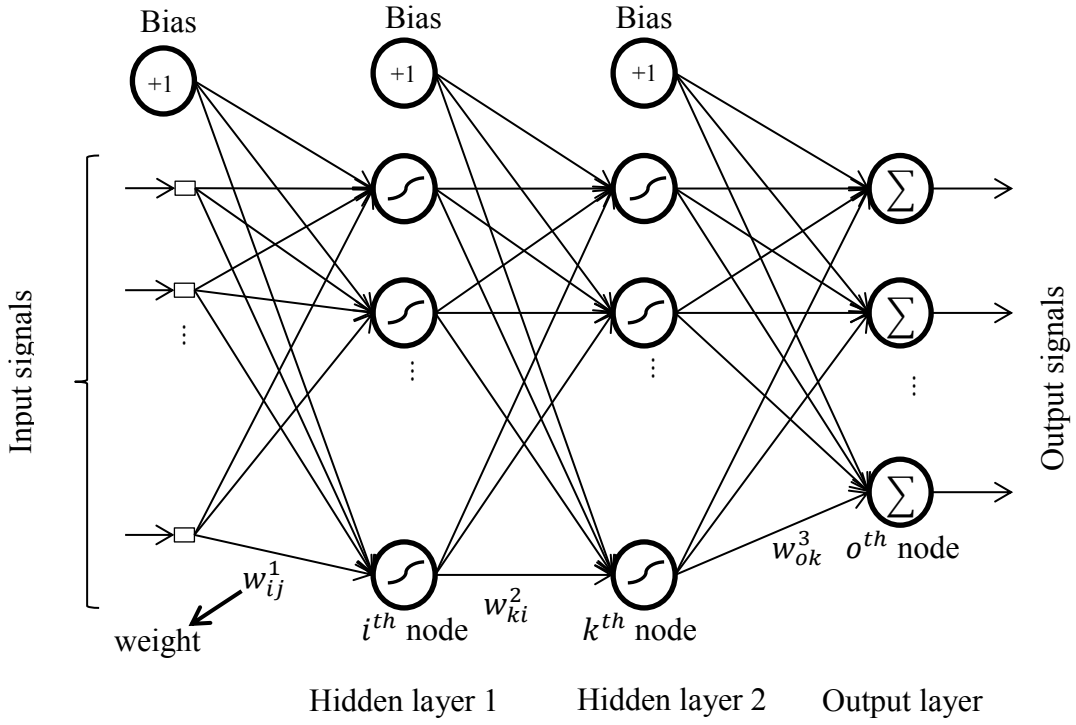


Fig. 2.2 Multi-layer Perceptron with two hidden layers.

2.1.2 Radial Basis Functions Network

Radial Basis Functions Neural Network is another type of feed forward networks which can be used for pattern discrimination and classification, interpolation, prediction and time series problems [14]. It has the advantages of fast learning, high accuracy and strong self-adapting ability [15]. As it can be seen from Fig. 2.3, the structure of RBFNN involves three layers. The first layer is composed of input features. Each feature in the first layer is directly connected to all neurons of the hidden layer without any weight. Each neuron in the hidden layer implements one radial basis function and provides a nonlinear transformation for the input space. The Gaussian function is the most used activation function and is shown in eq. (2.5).

$$\varphi_i(\mathbf{x}, \mathbf{c}_i, \sigma_i) = e^{-\frac{\|\mathbf{x}-\mathbf{c}_i\|^2}{2\sigma_i^2}} \tag{2.5}$$

Where the activation of the radial basis function φ_i is localized around center $\mathbf{c}_i$ and its localization degree is limited by σ_i [16], and $\mathbf{x}$ is the input data sample. A localized representation of information that is done by hidden-layer neurons helps the training process not only to reduce the output error for the current data sample $\boldsymbol{x}$, but also to minimize disturbance to those already learned [17].

The output of the network is a linear combination of outputs from the hidden-layer nodes which is shown in eq. (2.6).

$$y(\mathbf{x}) = w_{\mathrm{n+1}} + \sum_{i=1}^{n} w_i \varphi_i(\mathbf{x}) \tag{2.6}$$

Where n is the number of neurons in hidden layer, $w_{\mathrm{n+1}}$ is the bias term and w_i are weights for the output linear combiner.

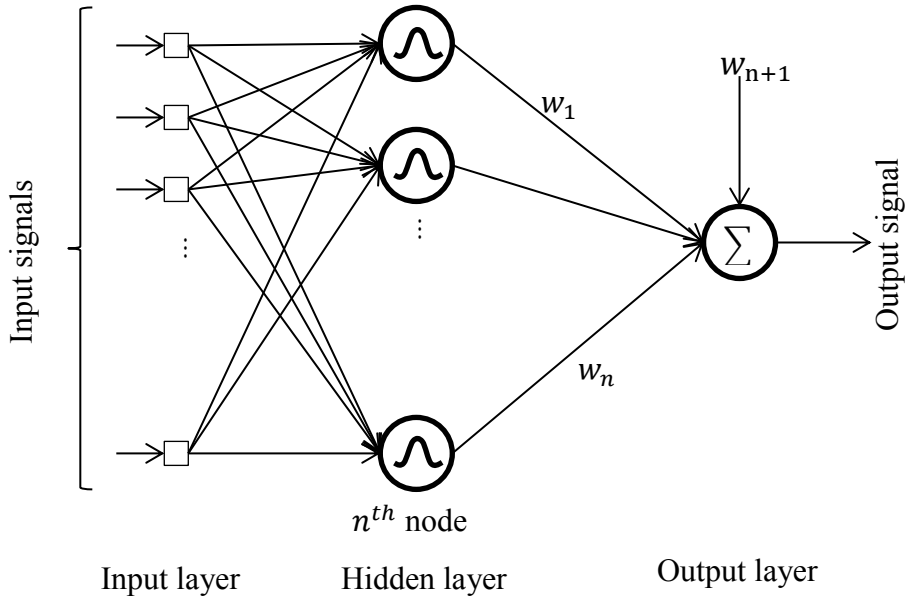


Fig. 2.3 Radial Basis Functions Neural Network with one hidden layer.

2.1.3 Learning Algorithms

As described in [11], there are three different points of view that can be used in categorizing learning algorithms. The first one is the mechanism that is used for learning which can be supervised, unsupervised, a combination of supervised and unsupervised and reinforcement type of learning. In supervised learning, each data sample that is fed to the learning algorithm is

previously labeled by a supervisor so the learning algorithm can compare its response with the actual label. In unsupervised type of learning, the data samples are not labeled by a supervisor so the learning algorithm has to look for the similarities within the data samples and determine which of them can form a group together. There is also a possibility to combine supervised and unsupervised methods to learn the parameters of a model. An example of this approach is discussed in section 2.1.3.3 for learning the parameters of an RBFNN model. Reinforcement learning is a kind of trial and error way of learning in which the learning algorithm interacts with the environment and learns from the consequences of its previous action. The algorithm is assigned a numerical value describing the amount of its success after doing each action. In fact, the algorithm learns how to select the action which maximizes its accumulated reward points.

The second point of view classifies learning algorithms based on the time that the parameters of the system are updated. If the parameter update occurs after seeing all the data samples, the learning algorithm acts in an offline manner. On the contrary, if parameter updates happens on arrival of each new data sample, we will have an online learning.

The third aspect categorizes learning algorithms based on whether parameter updates is done in a deterministic or stochastic way. Boltzmann learning rule is an example of stochastic learning approach. Fig. 2.4 provides a schematic diagram of the taxonomy of learning algorithms from the three different points of view.

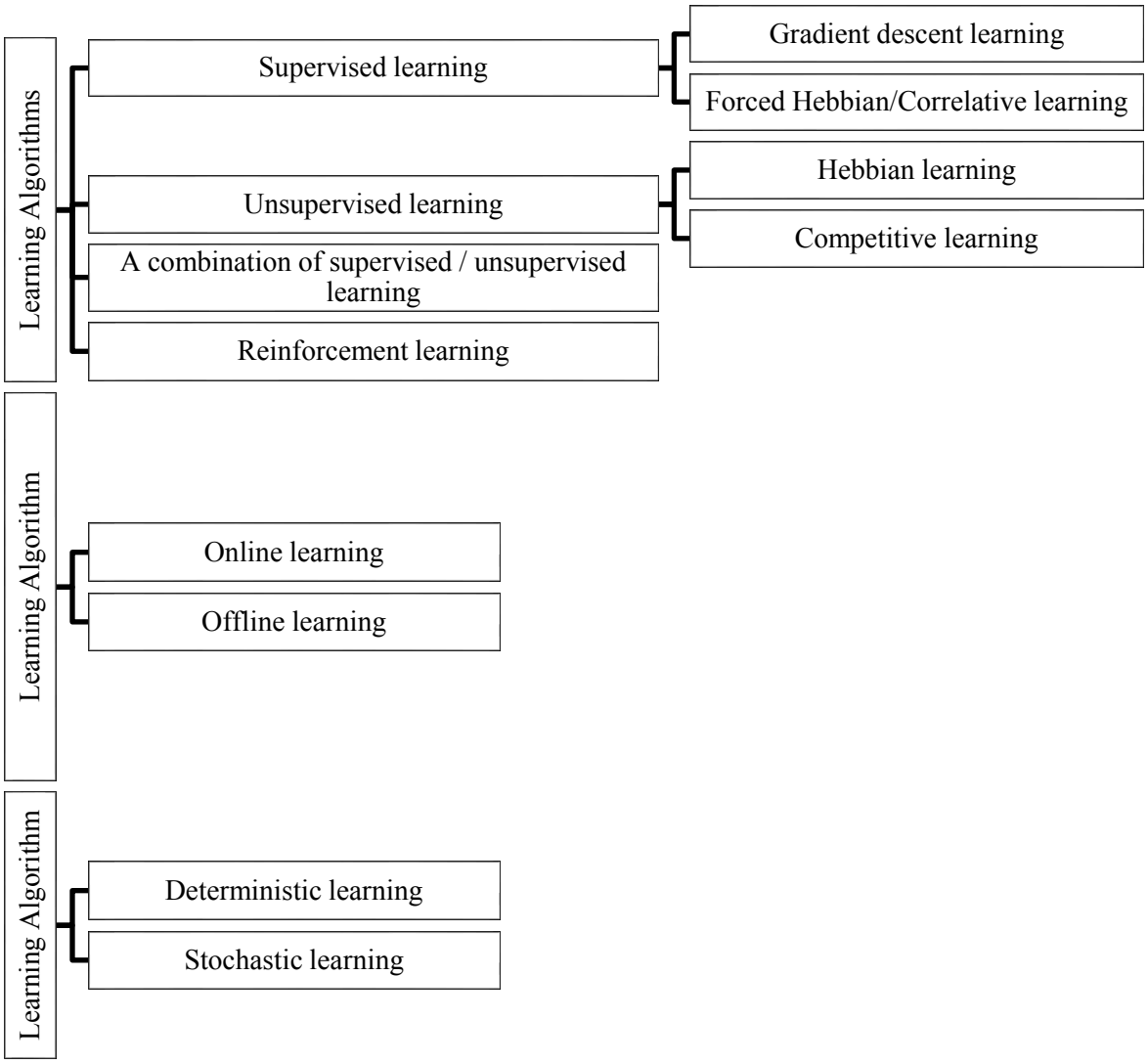


Fig. 2.4 A taxonomy of learning algorithms from three different points of view [11]

2.1.3.1 Supervised Learning

In supervised algorithms we need to provide both the input patterns and their corresponding desired outputs to the algorithm during the training process. The aim of the training is to infer a mapping function from the input space to the desired output space by minimizing the output error. A possible approach to minimize the output error is using the method below described.

2.1.3.1.1 Steepest Descent

Steepest descent is a gradient descent based method. This method is one of the simplest and the most fundamental minimization methods for unconstrained optimization. Given a cost function $\Omega(w_0, w_1, \dots, w_n)$, the steepest descent method will start by initializing $w_0, w_1, \dots, w_n$ to some random values and tries to minimize the cost function by updating parameters' values through $t = 1,2, \dots, T$ iterations. The update of parameter w_k is done by subtracting its current value from the gradient of cost function with respect to w_k as stated in eq. (2.7).

$$w_k^{(t+1)} = w_k^{(t)} - \alpha \frac{\partial}{\partial w_k^{(t)}} \Omega\left(w_0^{(t)}, w_1^{(t)}, \dots, w_n^{(t)}\right), \quad k = 0,1,2, \dots, n \tag{2.7}$$

Where α is the learning rate and defines the step size the algorithm is taking towards the local minima of the cost function. , Gradient descent cannot guarantee finding the global minimum and the result is strongly dependent on the initial values of the parameters as can be seen in Fig. 2.5.

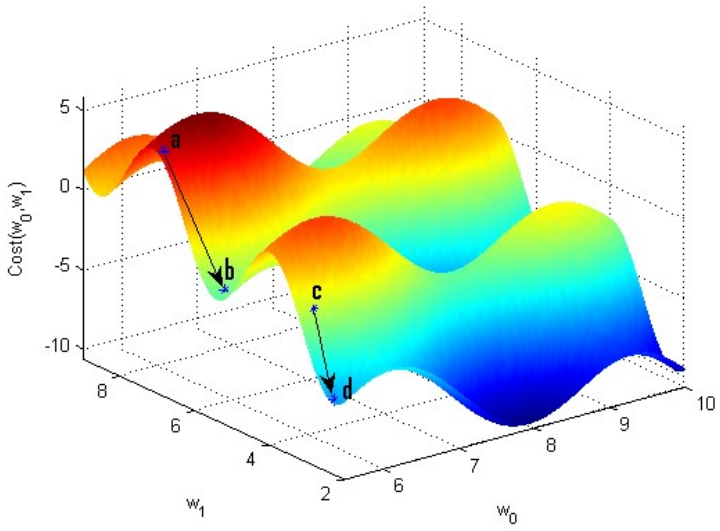


Fig. 2.5 The dependency of gradient descent on the initial parameters' value

2.1.3.1.1.1 Back Propagation technique

For training the MLP neural network, one can use the Back Propagation (BP) technique. BP uses the steepest descent method for training process. The aim is to find linear weights **w** which minimize the cost function stated in eq. (2.8).

$$\Omega(\mathbf{w}) = \frac{1}{2m} \times \sum_{i=1}^{m} (\mathrm{y}(\mathbf{x}_i, \mathbf{w}) - t_i)^2 \quad (2.8)$$

Where m is the number of training data samples, $\mathrm{y}(\mathbf{x}_i, \mathbf{w})$ is the output of MLP neural network for input data sample $\mathbf{x}_i$, parameterized by the weights and t_i is the target value for i^{th} training data sample. BP algorithm first initializes $\mathbf{W}$ parameters with random values and continue to update these parameters until the performance of the network is satisfactory. To update the weight $w_{ab}^{l\ (j)}$ (i.e., which connects neuron a in layer l to neuron b in layer -1) in j^{th} iteration eq. 2.9 will be used.

$$w_{ab}^{l\ (j)} = w_{ab}^{l\ (j-1)} - \alpha \frac{\partial}{\partial w_{ab}^{l\ (j-1)}} \Omega(\mathbf{w}) \quad (2.9)$$

Where α is the learning rate and $\frac{\partial}{\partial w_{ab}^{l\ (j-1)}} \Omega(\mathbf{w})$ is the gradient of cost function with respect to w_{ab}^{l} in $(j-1)^{th}$ iteration. The gradient value for all weights will be calculated using the following procedure:

Given a training example$(\mathbf{x}_i, t_i)$, a "forward pass" will be run to compute all the activations throughout the network, including the output value $\mathrm{y}(\mathbf{x}_i, \mathbf{w})$. Then, for each neuron in layer l, it is measured how much this neuron is responsible for any errors at the output in iteration $(j-1)$. It is done by calculating the error term $\boldsymbol{\delta}^{l\ (j-1)}$. $\boldsymbol{\delta}^{L+1\ (j-1)}$ (please remember that L is the number of hidden layers) which can be directly obtained by calculating the difference between the network's activation and the true target value. For the hidden units, $\boldsymbol{\delta}^{l\ (j-1)}$ will be computed based on a weighted average of the error terms of the nodes in layer $l+1$ as stated in eq. (2.10).

$$\boldsymbol{\delta}^{l\ (j-1)} = \left(\mathbf{w}^{l\ (j-1)}\right)^T \boldsymbol{\delta}^{l+1\ (j-1)} .* g'\left(\mathbf{z}^{l\ (j-1)}\right), \quad 2 \le l \le \mathrm{L} \quad (2.10)$$

Where $\mathbf{w}^{l\ (j-1)}$ is a vector of linear weights in iteration $(j-1)$ which connects layer l to layer $l-1$; g' is the derivative of sigmoid function g shown in eq. 2.11 and $\mathbf{z}^{l\ (j-1)}$ can be calculated from eq. 2.12

$$g'(z) = g(z)(1 - g(z)) \text{ where } g(z) = \frac{1}{1+e^{-z}} \quad (2.11)$$

$$\mathbf{z}^{l\ (j-1)} = \begin{cases} \mathbf{w}^{1\ (j-1)}\mathbf{x}_i & , \quad l = 2 \\ \mathbf{w}^{l-1\ (j-1)} g\left(\mathbf{z}^{l-1\ (j-1)}\right), & 3 \le l \le \mathrm{L} \end{cases} \quad (2.12)$$

This procedure is repeated for all data samples. The gradient of cost function with respect to each weight w_{ab}^{l} in iteration $(j-1)$ will be obtained as stated in eq. (2.13).

$$\frac{\partial}{\partial w_{ab}^{l\ (j-1)}}\Omega(\mathbf{w}) = \frac{1}{m} \times \sum_{i=1}^{m} \varphi_{b}^{l\ (j-1)}(\mathbf{x}_i)\delta_{a}^{l+1\ (j-1)}(\mathbf{x}_i) \tag{2.13}$$

Where $\varphi_{b}^{l\ (j-1)}(\mathbf{x}_i)$ is the output of neuron b in layer l for input pattern $\mathbf{x}_i$ in iteration $(j-1)$.

2.1.3.1.2 Newton's Method

Steepest descent method may take a large number of iterations to converge. Newton's method gives a much faster solution to find the parameter values for which the cost function $\Omega(\mathbf{w})$ is minimum. Newton's method starts by initializing $\mathbf{w}$ to some random values where $\mathbf{w}$ is a $n \times 1$ vector of parameters. It then approximates the cost function by a quadratic function in the current location using the second-order Taylor expansion. The next step is to minimize this model and obtain the next parameter values. Fig. 2.6 shows one step of Newton's method. Eq. (2.14) states the quadratic estimation of cost function $\Omega(\mathbf{w})$ around point $\mathbf{w}^{(t)} = (w_0^{(t)}, w_1^{(t)}, \dots, w_n^{(t)})$.

$$\Omega(\mathbf{w}) \approx \Omega\left(\mathbf{w}^{(t)}\right) + \nabla_{\mathbf{w}^{(t)}} \times (\mathbf{w} - \mathbf{w}^{(t)})^T + \frac{1}{2} \times (\mathbf{w} - \mathbf{w}^{(t)})^T \mathbf{H}^{(t)} (\mathbf{w} - \mathbf{w}^{(t)}) \tag{2.14}$$

Where $\nabla_{\mathbf{w}^{(t)}}$ and $\mathbf{H}^{(t)}$ are the gradient and the second partial derivative of the cost function at point $\mathbf{w}^{(t)}$, respectively. The second partial derivative of cost function, $\mathbf{H}^{(t)}$, is also called the Hessian matrix. The estimated function depicted in (2.14) will be minimized when eq. (2.15) is satisfied.

$$\nabla_{\mathbf{w}^{(t)}} + \mathbf{H}^{(t)}\left(\mathbf{w} - \mathbf{w}^{(t)}\right) = 0 \tag{2.15}$$

Solving eq. (2.15) will give us the Newton's method parameter update which can be seen in eq. (2.16).

$$\mathbf{w} = \mathbf{w}^{(t)} - (\mathbf{H}^{(t)})^{-1} \nabla_{\mathbf{w}^{(t)}} \tag{2.16}$$

For Newton's method to work, the Hessian $\mathbf{H}^{(t)}$ has to be a positive definite matrix for all t which in general cannot be guaranteed [12]. Moreover, within each iteration of Newton's method we need to calculate the inverse of Hessian matrix which is of order $O(n^3)$, being n the number of parameters. These limitations stimulated the development of alternatives to Newton's method, the Quasi-Newton methods. These methods are general unconstrained optimization methods, and

therefore do not make use of the special structure of nonlinear least square problems [11]. Two other methods that exploit this structure but assume that the problem is of type non-linear least square are the Gauss-Newton and the Levenberg-Marquardt methods, which will be presented in sections Section 2.1.3.1.4 and Section 2.1.3.1.5 respectively.

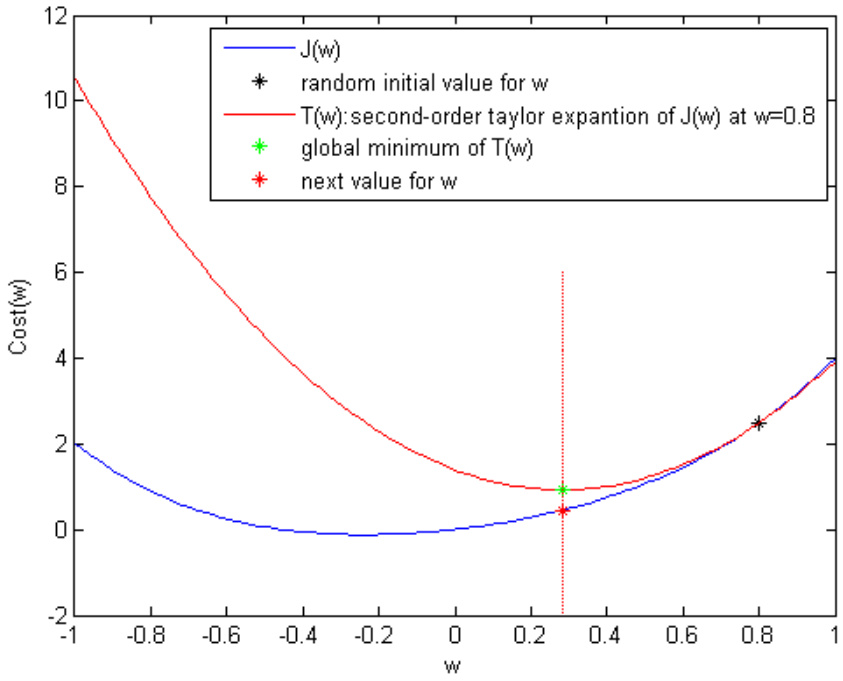


Fig. 2.6 Example of one step of Newton's method

2.1.3.1.3 Quasi -Newton methods

In quasi-Newton methods the Hessian matrix does not need to be computed. Instead, the Hessian matrix is updated by calculating the change of gradient between the current and previous iteration. There are different methods for updating Hessian matrix such as Davidson–Fletcher–Powell formula (DFP), SR1 formula (Symmetric Rank one), the BHHH method, the BFGS method and its low-memory extension, L-BFGS [18]. It is stated in [11] that BFGS update rule, shown in eq. (2.17), is the most effective for a general unconstrained method.

$$\mathbf{H}_{BFGS}^{(t+1)} = \mathbf{H}^{(t)} + \left(1 + \frac{(\mathbf{q}^{(t)})^T \mathbf{H}^{(t)} \mathbf{q}^{(t)}}{(\mathbf{s}^{(t)})^T \mathbf{q}^{(t)}}\right) \frac{\mathbf{s}^{(t)} (\mathbf{s}^{(t)})^T}{(\mathbf{s}^{(t)})^T \mathbf{q}^{(t)}} - \left(\frac{\mathbf{s}^{(t)} (\mathbf{q}^{(t)})^T \mathbf{H}^{(t)} + \mathbf{H}^{(t)} \mathbf{q}^{(t)} (\mathbf{s}^{(t)})^T}{(\mathbf{s}^{(t)})^T \mathbf{q}^{(t)}}\right) \qquad (2.17)$$

Where $\mathbf{s}^{(t)} = \mathbf{w}^{(t+1)} - \mathbf{w}^{(t)}$ and $\mathbf{q}^{(t)} = \nabla_{\mathbf{w}^{(t+1)}} - \nabla_{\mathbf{w}^{(t)}}$.

2.1.3.1.4 Gauss-Newton method

Since the calculation of Hessian Matrix and its inverse can be problematic and expensive, Gauss-Newton method uses another estimation of Hessian matrix in Newton's update rule formula, eq. (2.16). To estimate the Hessian matrix, Gauss-Newton method assumes that the problem is a non-linear least square problem. The cost function of such problems is stated in eq. (2.18).

$$\Omega(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{m} \mathbf{e}_i^2(\mathbf{w}) \ , \mathbf{w} = (w_0, w_1, \dots, w_n) \tag{2.18}$$

Where m is the number of data samples and $\boldsymbol{e}_i(\mathbf{w})$ is the error of the network parameterized by $\mathbf{w}$ while feeding i^{th} input pattern. The elements of first-order partial derivative of $\Omega(\mathbf{w})$ is computed as depicted in eq. (2.19).

$$\nabla_{w_j} = \sum_{i=1}^{m} \mathbf{e}_i \frac{\partial \mathbf{e}_i}{\partial \boldsymbol{w}_j} = \sum_{i=1}^{m} \mathbf{e}_i \mathbf{J}_{ij} \ \ , j = 0,1, \dots, n \tag{2.19}$$

Eq. (2.20) shows the matrix notation of eq. (2.19) in t^{th} iteration

$$\nabla_{\mathbf{w}^{(t)}} = (\mathbf{J}^{(t)})^T \mathbf{e}^{(t)} \tag{2.20}$$

Where $\mathbf{J}_{ij}$s are the elements of Jacobean matrix $\mathbf{J}$. The elements of Hessian matrix can also be computed by eq. (2.21).

$$\mathbf{H}_{jk} = \sum_{i=1}^{m} \left(\frac{\partial \mathbf{e}_i}{\partial \mathbf{w}_j} \frac{\partial \mathbf{e}_i}{\partial \mathbf{w}_k} + \mathbf{e}_i \frac{\partial^2 \mathbf{e}_i}{\partial \mathbf{w}_j \partial \mathbf{w}_k} \right) , \ \ j, k = 0,1, \dots, n \tag{2.21}$$

Gauss-Newton method ignores the second term in eq. (2.21) and approximate the elements of Hessian matrix as stated in eq. (2.22)

$$\mathbf{H}_{jk} \approx \sum_{i=1}^{m} \left(\frac{\partial \mathbf{e}_i}{\partial \mathbf{w}_j} \frac{\partial \mathbf{e}_i}{\partial \mathbf{w}_k} \right) = \sum_{i=1}^{m} \mathbf{J}_{ij} \mathbf{J}_{ik} \ , \ \ j, k = 0,1, \dots, n \tag{2.22}$$

Eq. (2.23) shows the matrix notation of eq. (2.22) in t^{th} iteration.

$$\mathbf{H}^{(t)} = (\mathbf{J}^{(t)})^T \mathbf{J}^{(t)} \tag{2.23}$$

By replacing eqs (2.20) and (2.23) in eq. (2.16), the Gauss-Newton update rule will be obtained as depicted in eq. (2.24).

$$\mathbf{w} = \mathbf{w}^{(t)} - ((\mathbf{J}^{(t)})^T \mathbf{J}^{(t)})^{-1} (\mathbf{J}^{(t)})^T \mathbf{e}^{(t)} \tag{2.24}$$

2.1.3.1.5 Levenberg-Marquardt

As previously stated, Gauss-Newton method is faster than the steepest descent but one of the problems associated with Gauss-Newton method [19] is that there is no guarantee for $(\mathbf{J}^{(t)})^T \mathbf{J}^{(t)}$ to be invertible as we need to calculate the inverse of $(\mathbf{J}^{(t)})^T \mathbf{J}^{(t)}$ in each iteration t. In order to guarantee that $(\mathbf{J}^{(t)})^T \mathbf{J}^{(t)}$ is invertible, it must be a nonsingular matrix. A nonsingular matrix is a square matrix whose determinant is nonzero. To guarantee the nonsingularity of $(\mathbf{J}^{(t)})^T \mathbf{J}^{(t)}$, Jacobean matrix $\mathbf{J}$ must have row rank n; that means the n rows of matrix $\mathbf{J}$ should be linearly independent.

Levenberg-Marquardt algorithm blends Gauss-Newton and steepest descent methods to take the advantage of Gauss-Newton speed while handling the situations at which the divergence happens. To blend the two methods, Levenberg-Marquardt introduces the update rule as shown in eq. (2.25).

$$\mathbf{w} = \mathbf{w}^{(t)} - ((\mathbf{J}^{(t)})^T \mathbf{J}^{(t)} + \delta \mathbf{I})^{-1} (\mathbf{J}^{(t)})^T \mathbf{e}^{(t)} \tag{2.25}$$

Where $\mathbf{I}$ is an identity matrix and δ is a scalar value. Please note that by adding diagonal matrix $\delta \mathbf{I}$ to term $(\mathbf{J}^{(t)})^T \mathbf{J}^{(t)}$, Levenberg-Marquardt algorithm avoids rank deficiency of $\mathbf{J}$ in iteration t and guarantees the inevitability of term $(\mathbf{J}^{(t)})^T \mathbf{J}^{(t)} + \delta \mathbf{I}$.

Levenberg-Marquardt changes the value of δ based on the current situation. Considering eq. (2.25), if we set the value of δ quite small and near to zero, the effect of second-derivative elements increases and the algorithm behaves as the Gauss-Newton approach does. On the other hand, if we increase the value of δ, the effect of second-derivative elements can be neglected and the algorithm follows the steepest descent approach. In this situation, Levenberg-Marquardt starts with rapid Gauss-Newton approach by assigning a small value to δ.

If the error goes down following an update, it means that the quadratic assumption on $\Omega(\mathbf{w})$ (i.e., the cost function) is valid; so we accept the new values for the parameters and continue with Gauss-Newton approach for the next iteration. The algorithm even makes δ smaller usually by a factor of 10 for the next iteration [20].

On the contrary, if the error goes up following an update (i.e., divergence happened maybe because of taking a very big step while we were near the minimum location so we simply jumped over the minimum), Levenberg-Marquardt resets parameters to their previous values and increases the value of δ (i.e., usually by a factor of 10 [20]) to enhance the influence of steepest descent part.

2.1.3.2 Improving the performance of the training algorithms for nonlinear least square problems by separating linear and nonlinear parameters

As stated in [21], considering a nonlinear least square problem, in order to improve the performance of the training algorithms, one can exploit the separability of the model parameters into linear and nonlinear. Suppose that $\mathbf{u}$ and $\boldsymbol{\upsilon}$ are vectors of linear and nonlinear parameters, respectively. The output of the models can be represented as eq. (2.26).

$$\mathbf{y} = \boldsymbol{\phi}(\boldsymbol{\upsilon})\mathbf{u} \tag{2.26}$$

Where $\boldsymbol{\phi}$ represents the output matrix of the last nonlinear hidden layer (possibly with a column of ones to consider the model output bias). When eq. (2.26) is replaced in eq. (2.18), we have:

$$\Omega(\mathbf{u}, \boldsymbol{\upsilon}) = \frac{1}{2}\sum_{i=1}^{m} \mathbf{e}_i^2(\mathbf{u}, \boldsymbol{\upsilon}) = \frac{\|\mathbf{t}-\boldsymbol{\phi}(\boldsymbol{\upsilon})\mathbf{u}\|_2^2}{2} \tag{2.27}$$

Where $\mathbf{t}$ is a vector of target values. For any value of $\boldsymbol{\upsilon}$, the minimum of cost function Ω with respect to $\mathbf{u}$ can be obtained using the least squares solutions, here determined with application of pseudo-inverse:

$$\hat{\mathbf{u}}(\boldsymbol{\upsilon}) = \boldsymbol{\phi}(\boldsymbol{\upsilon})^{+}\mathbf{t} \tag{2.28}$$

By replacing eq. (2.28) in eq. (2.27) a new criterion is obtained, as shown in eq. (2.29), which only depends on the nonlinear parameters.

$$\psi(\boldsymbol{\upsilon}) = \frac{\left\|\mathbf{t}-\boldsymbol{\phi}(\boldsymbol{\upsilon})\boldsymbol{\phi}(\boldsymbol{\upsilon})^{+}\mathbf{t}\right\|_2^2}{2} \tag{2.29}$$

To minimize eq. (2.29), its gradient with respect to $\boldsymbol{\upsilon}$ must be computed. It can be proved [22] that the gradient of ψ can be determined, computing first the optimal value of the linear parameters, using eq. (2.28), replacing this in the model, and subsequently performing the usual calculation of the gradient (only for the partition related with the nonlinear parametrs). Using the criterion stated

in eq. (2.29) presents some advantages, comparing with the use of the criterion depicted in eq. (2.27) [21]:

- It lowers the dimensionality of the problem;
- When the Levenberg-Marquardt is used, each iteration is computationally cheaper;
- Usually a smaller number of iterations is needed for convergence to a local minimum, since:
    1. The initial value of (2.29) is much lower than the one obtained with (2.27);
    2. Eq. (2.29) usually achieves a faster rate of convergence than (2.27).

2.1.3.3 Three learning strategies for training RBFNN

There are three different learning strategies to determine the parameters of a RBF neural network, depending on the approach that is considered for determining the centers and spreads of the radial-basis functions of the network [12, 23, 24].

In one approach, the centers can be selected from the training data set in a random manner. The spreads can then be calculated using eq. (2.30) [25].

$$\sigma = \frac{d_{max}}{\sqrt{2n}} \tag{2.30}$$

Where d_{max} is the maximum Euclidean distance between centers and n is the number of centers. If the training data are distributed in a representative manner for the problem at hand, this could be a wise approach. The linear parameters $\mathbf{u} = [u_1, u_2, \cdots, u_n, \alpha_0]$ are then found using eq. (2.31).

$$\mathbf{u} = \emptyset^{+}\mathbf{t} \tag{2.31}$$

Where $\mathbf{t}$ is a vector of target values and $\emptyset^{+}$ is the pseudo-inverse of the hidden neurons' output matrix.

Another approach is a combination of supervised and unsupervised methods which is also called self-organized selection of centers. In this approach, the locations of the centers are found by a clustering technique like k-means clustering (please see Algorithm 2.1).

Algorithm 2.1 Using k-means clustering to find RBFNN centers [21]

1. Initialization - Choose random values for the centers; they must be all different
$j = 1$
2. While *go_on*
   2.1. Sampling - Find a sample vector $\mathbf{x}(j)$ from the input matrix
   2.2. Similarity matching - Find the center (out of m_1) closest to $\mathbf{x}(j)$. Let its index be $k(x)$:
$$k(x) = \arg\min_i \left\|\mathbf{x}_j - \mathbf{c}_i\right\|_2, i = 1,2,\dots,m_1 \quad (2.35)$$
   2.3. Updating - Adjust the centers of the radial basis functions according to:
$$\mathbf{c}_i[j+1] = \begin{cases} \mathbf{c}_i[j] + \eta(\mathbf{x}(k) - \mathbf{c}_i), & i = k(x) \\ \mathbf{c}_i[j] & , otherwise \end{cases} \quad (2.36)$$
   2.4. $j = j + 1$

End

The spreads can then be calculated using eq. (2.30) or other heuristic methods that are listed in the following [21]:

1. The empirical standard deviation
$$\sigma_i = \sum_{j=1}^{n} \sqrt{\frac{\left\|\mathbf{c}_i - \mathbf{x}_j\right\|_2^2}{n}} \quad (2.32)$$
   Where n denotes the number of patterns assigned to cluster i
2. The k-nearest neighbors heuristic considers the k nearest centers to the center $\mathbf{c}_i$, the spread associated with this center is obtained by eq. (2.33)
$$\sigma_i = \frac{\sum_{j=1}^{k}\left\|\mathbf{c}_i - \mathbf{c}_j\right\|_2}{k\sqrt{2}} \quad (2.33)$$
3. Maximum distance between patterns
$$\sigma = \frac{\max_{i,j=1,2,\dots,m}\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2}{2\sqrt{2}} \quad (2.34)$$
   Where m is the total number of patterns. In this method all centers will have the same spread.

After identifying the centers and spreads of the radial basis functions, the linear weights can be obtained using a linear least square strategy.

In the last approach, considering that our problem is a nonlinear least square problem, both linear weights and non-linear parameters centers and the spread are computed as described in section 2.1.3.2.

2.1.3.4 Termination criterion for training process

By training process, we are aiming to construct a function based on the available training set $(\mathbf{x}_1, y_1), \ldots (\mathbf{x}_n y_n)$, for the purpose of estimating y at future observations $\mathbf{x}$. The training process is usually terminated when a certain, user-specified level of accuracy is obtained. There are mainly three approaches for this purpose. The first approach is to define a fixed number of iterations for the training process. The main shortage of this approach is that, while making this decision, it does not use any information about how well the parameters are adapted to the training data and how well they are generalized. The second approach is to check whether conditions mentioned in eq. (2.37) to eq. (2.39) are simultaneously met by the end of each iteration [21]. This approach is commonly used in nonlinear optimization problems.

$$\Omega[k-1] - \Omega[\mathrm{k}] < \theta[\mathrm{k}], \tag{2.37}$$

$$\|\mathbf{w}[k-1] - \mathbf{w}[k]\| < \sqrt{\tau_f}.(1 + \|\mathbf{w}[k]\|) \tag{2.38}$$

$$\|\mathbf{g}[k]\| \leq \sqrt[3]{\tau_f}.(1 + |\Omega[k]|) \tag{2.39}$$

Where

$$\theta[\mathrm{k}] = \tau_f.(1 + \Omega[k]) \tag{2.40}$$

τ_f is a measure of the desired number of correct digits in the training criterion, $k-1$ and k denote two consecutive iterations and $\mathbf{w}$, $\mathbf{g}$ and Ω are referring to weights, gradient and cost function respectively. As stated in [21], if a small value for τ_f is specified, overfitting problems can happen. The term overfitting refers to a situation in which our model is too much adjusted to the training data at hand. Since it is normal for training data to have noise, the overfitted model learns this noise. Such a model does not achieve a convenient generalization and will not perform well for the unseen data. Fig. 2.7 shows an overfitted and a model with good generalization. To avoid this situation, the third approach, which is called early stopping, can be used.

In early stopping, the dataset is split into two parts, namely training and test sets. The model is trained using data samples in the training set. By the end of each iteration, the cost function is evaluated for both training and test sets. If the parameter updates are reducing the cost for both training and test data samples, the training process is going well. On the contrary, if we are still having reduction on the cost of training data samples but the cost of test data samples is increasing, it means that the model is over-trained. At this point, early stopping will stop the training process. Fig. 2.8 visualizes this situation.

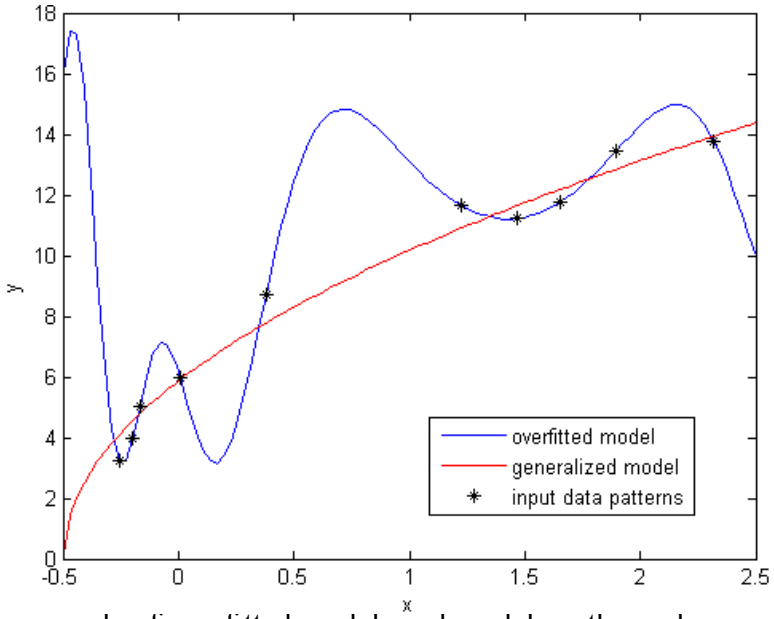


Fig. 2.7 An example of overfitted models and models with good generalization

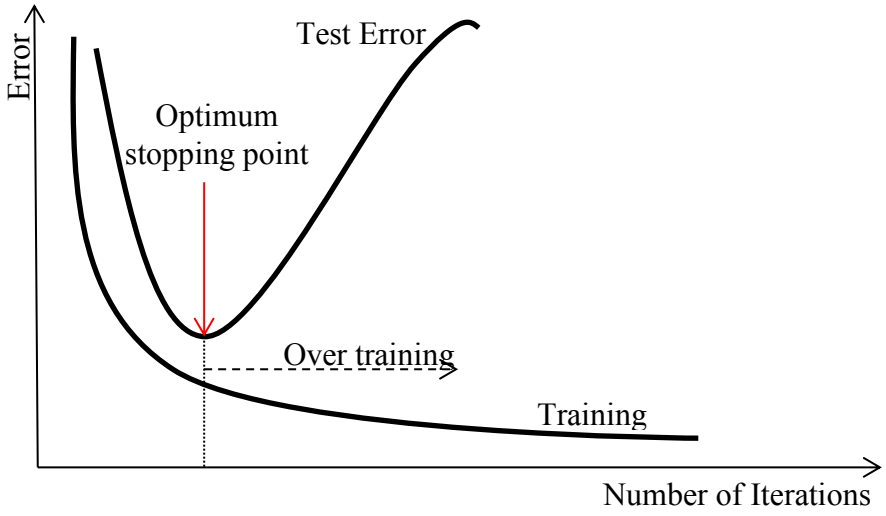


Fig. 2.8 Early stopping approach stops the training process at the optimum point to have a generalized model.

2.2 Support Vector Machines

Support Vector Machine was introduced as a machine learning method by Cortes and Vapnik [26]. Given a two class classification problem, the main idea was to nonlinearly map the input data samples to a higher dimensional feature space in which they are linearly separable, and then try to find the best separating hyper plane between the data samples of the two classes. In fact, a good separation is achieved by the hyper plane that has the largest distance to the nearest training data sample of any class. This distance is called maximal margin and those data samples that reside on the margin are called support vectors. In general, the larger the margin the lower the generalization error of the classifier is (Please see Fig. 2.9).

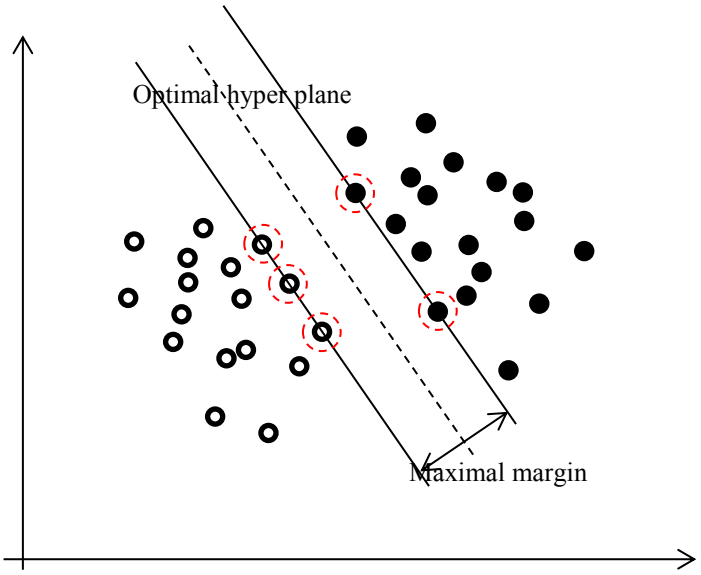


Fig. 2.9 An example of a separable problem in a 2 dimensional space. The support vectors, marked with red circles, define the margin of largest separation between the two classes [26].

The determination of the large margin hyper plane is performed, in SVMs, by solving a constrained Quadratic Problem [27]. Making use of KuhnTucker theory, the Lagrangian stated in eq. (2.41) must be maximized with respect to α_i subject to the constraints depicted in eq. (2.42).

$$L = \sum_{i=1}^{N} \alpha_i - \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{2.41}$$

$$\sum_{i=1}^{N} \alpha_i \, y_i = 0 \quad and \quad 0 \leq \alpha_i \leq C \tag{2.42}$$

Where N is the number of data samples in the training set; α_is are the Lagrange multipliers; $\mathbf{x}_i$ is the i^{th} training data sample; $y_i \in \{-1, +1\}$ determines the class to which $\mathbf{x}_i$ belongs; $K(\mathbf{x}_i, \mathbf{x}_j)$ are the inner-product kernels shown in eq. (2.43) and C is the penalty parameter to control the sensitivity of SVM to possible outliers. In other words, C controls the relative importance of maximizing the margin and satisfying the margin constraint for each data point.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{z=1}^{m} \phi_z(\mathbf{x}_i)\phi_z(\mathbf{x}_j) \tag{2.43}$$

In eq. (2.43) m is the dimension of the higher dimensional feature space where the transformed training data samples can be linearly separated. Some common kernel functions are shown in eqs (2.44)-(2.46).

- Homogeneous polynomial $K(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i}.\mathbf{x_j})^d$ (2.44)
- Inhomogeneous polynomial $K(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i}.\mathbf{x_j} + \mathbf{1})^d$ (2.45)
- Gaussian radial basis function $K(\mathbf{x_i}, \mathbf{x_j}) = exp(-\frac{\|\mathbf{x_i}-\mathbf{x_j}\|^2}{2\sigma^2})$ (2.46)

The decision function can be written as depicted in eq. (2.47).

$$f(\mathbf{x}) = sign(\sum_{i \in SV} y_i \alpha_i^* K(\mathbf{x}, \mathbf{x}_i) - \theta) \tag{2.47}$$

Where α_i^*s are the solution of the constrained maximization problem and SV represent the indexes of the support vectors and θ is a threshold value [28]. For a more detailed explanation of SVM please refer to [29].

2.3 Multi-Objective Genetic Algorithm

2.3.1 Genetic Algorithm

The Genetic Algorithm (GA) which was introduced by John Holland and his students aims to find an optimum solution within a search space, by mimicking the natural process of evolution. The natural process of evolution is based on two main principles: 1- competition or survival of the fittest and 2- child's inheritance of the parents' genetic makeup [30]. Genetic Algorithm starts its work by producing a number of potential solutions which is called the initial population. The initial

population is then evolved over a number of generations. Each potential solution (i.e., also called individual) is then evaluated and assigned a measure of fitness because the strategy is to use the elites to produce the next generation. To mimic this behavior, the first step is encoding the problem at hand in a way that each potential solution could be represented as a chromosome. To do that, we can see each problem as a black box with a series of input parameters and one output parameter. Input parameters control the behavior of the system. The output parameter is computed by an evaluation function and indicates how well a particular combination of input parameter settings can solve the optimization problem [31]. Each input parameter is considered as a gene and a particular combination of genes can produce a chromosome which is a potential solution to the problem at hand. The values of the genes (i.e., input parameters) are normally selected from a discrete domain.

After defining the genetic representation of the solution domain, the individuals of the first generation will be produced, typically in a random manner allowing the whole range of the search space to be incorporated. Within each generation, a proportion of the existing population should be selected to breed the next generation. The main idea is to allow the genes of best individuals to pass to the next generation. This is done through assigning a fitness value to each individual using a fitness function. Having identified the fitness values, one can use different techniques for the selection process, including: roulette wheel (or Fitness Proportionate Selection), Stochastic Universal Sampling (SUS), Tournament selection and Truncation selection.

In the roulette wheel method, first the fitness value of each individual is normalized by dividing its corresponding value by the sum of all fitness values. Then, the accumulated fitness value of each individual is calculated. The accumulated fitness value of each individual is the sum of its own fitness value and the fitness values of all previous individuals. The next step is to produce a random number R in range $[0\ 1]$. The selected individual is the first one whose accumulated normalized value is greater than R. This procedure is repeated until a number of desired individuals is selected. Fig. 2.10 shows a roulette wheel of four individuals. The accumulated normalized value of each individual is written inside the corresponding portion.

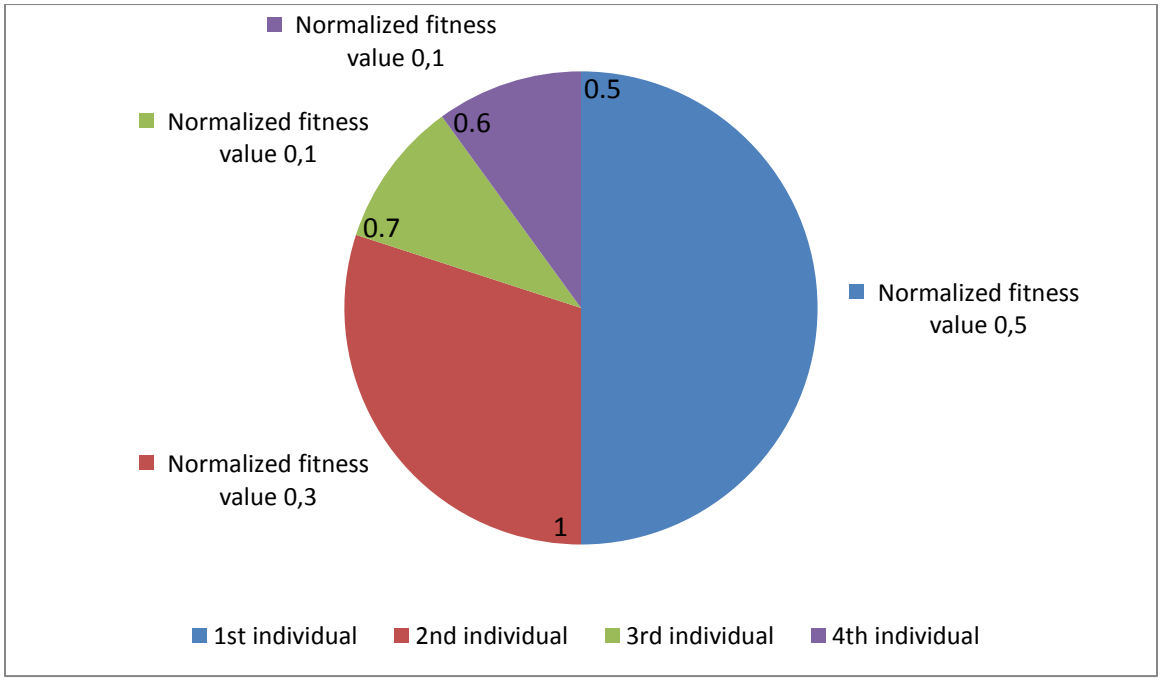


Fig. 2.10 Roulette wheel of 4 individuals. The accumulated normalized value of each individual is written inside the corresponding portion.

SUS is an improved version of roulette wheel. In this method, instead of having one pointer on the roulette wheel, there are multiple, equally spaced pointers. The number of pointers is equal to the number of individuals that should be selected. As a result, all individuals will be selected simultaneously. The position of the first pointer is determined by producing a random number.

In Tournament selection, several groups of individuals are randomly selected from the population. The size of each group is determined by a parameter named "tournament size". Within each group, the individual which has the highest fitness value wins the tournament and is selected as one of the parents of next generation.

Truncation selection, first orders the individuals by their fitness value and then selects p proportion of the fittest individuals (e.g., $p = \frac{1}{2}, \frac{1}{3}, etc.$).

Having selected the parents, we have now a pool of good parents to produce the next generation. A combination of genetic operators including crossover and mutation is used for generating new individuals. The main idea behind the crossover is to combine useful segments of different parents

and obtain a new individual which benefits from advantageous gene combinations of both parents. There are different techniques for doing crossover; three of them are presented in Fig. 2.11.

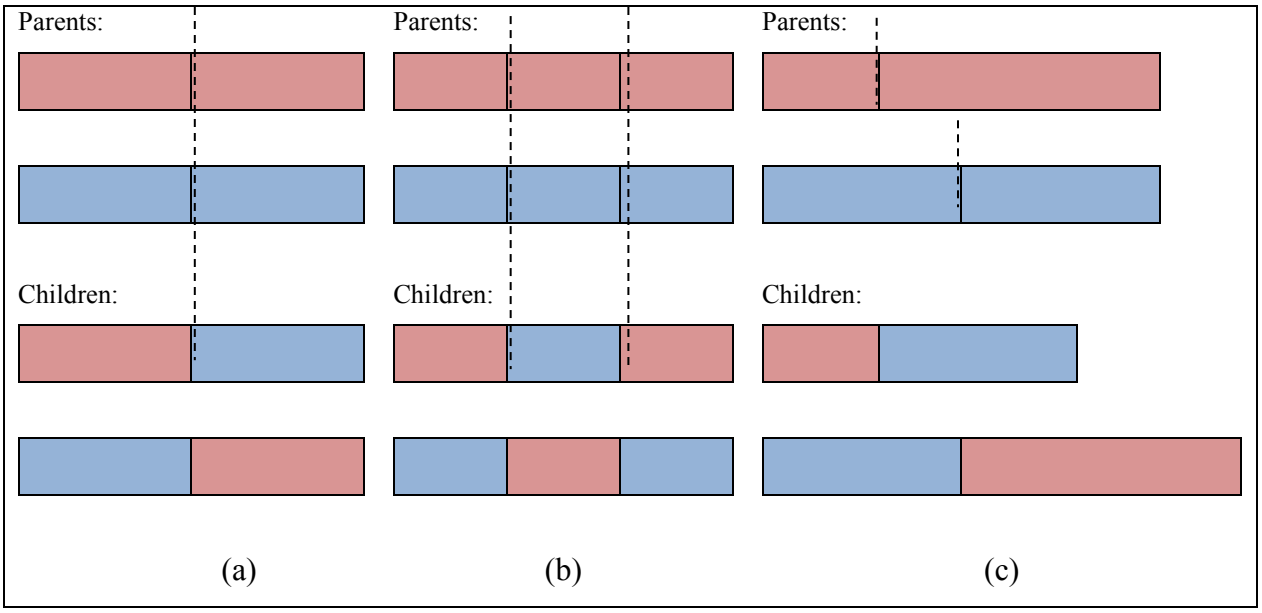


Fig. 2.11 Three different methods of crossover; (a) One-point crossover; (b) Two-point crossover; (c) Cut and splice crossover.

Mutation alters one or more gene values in a chromosome. Applying this operator may result to have a totally different chromosome. The genetic operators "crossover" and "mutation" are done with respect to crossover and mutation probability parameters.

After crossover and mutation, the generated offspring should be inserted into the population. The original genetic algorithm implements a generational replacement model, where the old population is all unconditionally replaced by the new one. Some other strategies propose that the offspring should have the possibility to compete with at least some of their parents. In [32] the different approaches to select the parents to replace, the different ways to reinsert, and additional considerations on this topic are discussed.

Common terminating criteria for Genetic Algorithm are [33]:

1. An upper limit on the number of generations is reached
2. An upper limit on the number of evaluations of the fitness function is reached
3. The chance of achieving significant changes in the next generations is very low.

In order to set the upper limit for the first two criteria, we should be able to estimate a reasonable maximum search length which needs some knowledge about the problem at hand. On the contrary, the third option does not need any information about the problem. The third termination criterion has two variations: genotypical and phenotypical termination criteria. The genotypical approach terminates the GA when the current population meets certain convergence level with respect the chromosomes in the population. This criterion checks if a certain percentage of genes in the population has converged. The convergence of a gene to a certain value is determined by the GA designer by defining a threshold that should be reached. For example, if 90% of the chromosomes have the same value x in a given gene, it is said that the gene has converged to x. Then when a certain percentage of genes, say 80%, have converged, the algorithm stops. The phenotypical termination criterion checks the progress achieved within the last n generations. The progress can be expressed in terms of the average fitness value of the last n generations. If the average is beyond a predefined threshold ε, the algorithm terminates.

2.3.2 Multi Objective optimization using Genetic Algorithms

As stated in [34] there are many problems in the engineering domain that need to optimize several non-commensurable, competing objectives at the same time. This kinds of problems does not have usually, a unique, perfect solution. Instead, they have a set of non-dominate solutions which is also called the Pareto-optimal set.

Assuming a minimization problem, vector $\boldsymbol{u} = (u_1, \cdots, u_n)$ is said to dominate $\boldsymbol{v} = (v_1, \dots, v_n)$ if and only if $\boldsymbol{u}$ is partially less than $\boldsymbol{v}$ ($\boldsymbol{u} <_p \boldsymbol{v}$) [32]. The formal notation is shown in eq. (2.48)

$$\forall\, i \in \{1, \dots, n\}\ , u_i \leq v_i \ \ \wedge \ \ \ni i \in \{1, \dots, n\} : \ u_i < v_i \tag{2.48}$$

As a result, vectors $\boldsymbol{u} = (u_1, \cdots, u_n)$ and $\boldsymbol{v} = (v_1, \dots, v_n)$ are said to be non-dominate to each other if neither $\boldsymbol{v}$ dominates $\boldsymbol{u}$ nor $\boldsymbol{u}$ dominates $\boldsymbol{v}$.

The population-based behavior of GA gives us the power of doing a parallel search within the space defined by the objectives. Having produced each generation, we have a population of solutions to the problem at hand. The most challenging part is how to assign the fitness value to reflect to what extent one solution has already optimized the defined objectives. Using the weighted sum approach is the simplest way. In this approach, a weight is assigned to each objective prior to the search

procedure. The weights express the relative importance among objectives. This approach has some problems: inappropriate weights may result in a wrong search. On the other hand, it is only possible to determine appropriate weights after search, which means multiple optimizer runs will be necessary to find the good weights. Moreover, small changes in weights may lead to a large change in objective values and vice versa. Pareto-based fitness assignment is another approach which was first proposed by Goldberg and then modified by Fonseca and Fleming [32]. In this method, the individuals are ranked according to the number of individuals by which they are dominated. For example, if an individual is non-dominated, its corresponding rank is 0 and if an individual is dominated by three other individuals, its corresponding rank will be 3. Fig. 2.12 visualizes the Pareto ranking notion.

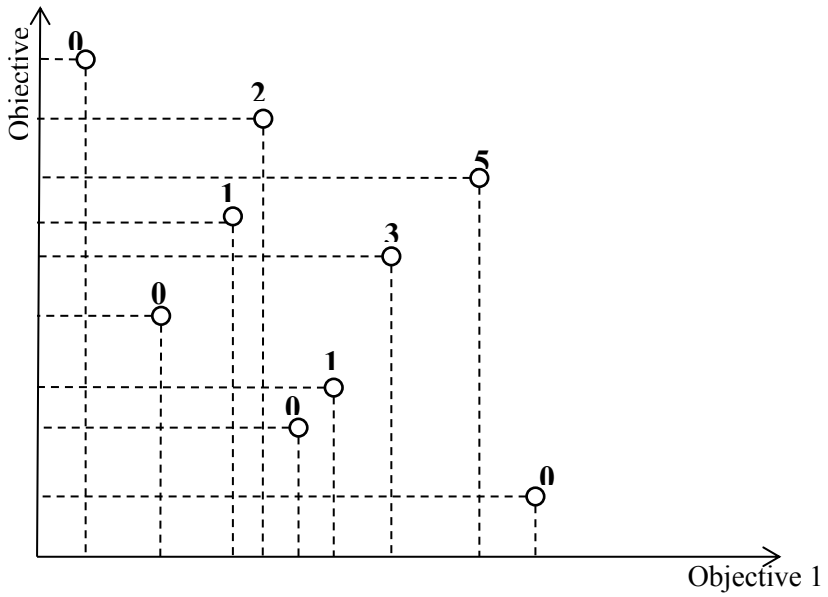


Fig. 2.12 Pareto ranking [34, 35]

If there exists any preference such as assigning different priorities to each objective or defining a desired level of performance for each objective (i.e., goals), the ranking technique is slightly modified to take the goals and priorities into account. Suppose that g_1 and g_2 are the corresponding goals of objectives 1 and 2. In the case that both objectives have the same priorities, the individuals who satisfied the goals are assigned a rank equal to the number of individuals by which they are dominated. The individuals which do not meet the goals are penalized by assigning a higher rank. Fig. 2.13 illustrates this situation.

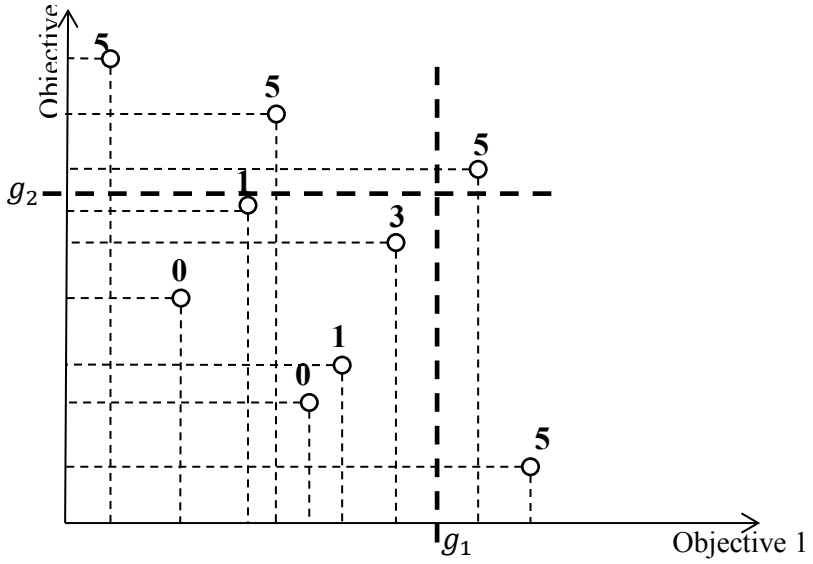


Fig. 2.13 Pareto ranking in the case that both objectives have equal priorities. Both objectives should meet the defined goals [32, 36].

Fig. 2.14 illustrates a situation in which objective 2 has a higher priority than objective 1. In this case, individuals which do not meet goal g_2 are the worst, independently of their performance according to objective 1. Ones g_2 is met, individuals are ranked based on how well they optimized objective 1 [32, 36].

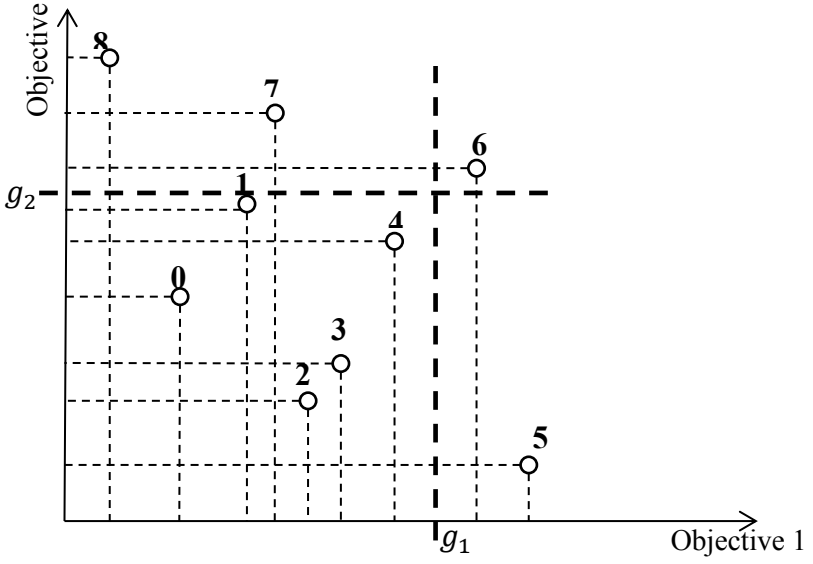


Fig. 2.14 Pareto ranking in the case that objective 2 has higher priority than objective 1. Both objectives should meet the defined goals [32, 36].

Having ranked the individuals, Multi Objective Genetic Algorithm assigns a fitness value to each individual based on its corresponding rank. To do that, the individuals are sorted based on ranks and the fitness is assigned by interpolating from the best individual (i.e., rank=0) to the worst according to a linear or exponential function. Finally, a single value of fitness is calculated for the individuals with the same rank by the means of averaging. Assigning the average value to those with the same rank will guarantee the same probability of being selected as the parent of next generation [32, 34, 37].

2.3.3 RBFNN structure determination using Multi Objective Genetic Algorithm

The identification of Neural Network structure and parameters (i.e., which need to be determined from data) is often done iteratively in an ad-hoc fashion focusing on the parameters identification. This is because the number of possibilities for selection of inputs and model structure are commonly very large. Moreover, the design criteria may include multiple conflicting objectives, leading the model identification problem to a multi-objective combinatorial optimization character.

In order to identify the best possible RBF neural network structure and parameters, this work uses the multi-objective neural network models identification method as presented in [2, 35]. This method also helps us to handle the conflicting objectives we have at the same time. For instance, we want to decrease model complexity and enhance the accuracy of the classification at the same time. Another example is our desire to have not only a very small amount of error in the training set, but also a model with good generalization.

Multi-Objective Genetic Algorithm first finds a non-dominated set of individuals through n number of generations and then selects preferable individuals from the non-dominated or preferable set.

In order to be able to use Genetic Algorithm approach for finding the best possible model structure and its corresponding parameters, each possible topology for the neural network needs to be formulated as a chromosome. To do that, the number of neurons in the hidden layer is considered as the first component of the chromosome and the remaining components are the indices of arbitrary number of features selected from the preliminary feature space. Fig. 2.15 shows the topology of the chromosome. The algorithm starts its work by producing a pre-defined number of chromosomes as the first generation. The method then needs a mechanism to compare the

individuals and select the best ones with respect to pre-defined objectives. The objectives can be selected from a set obj as described in eq. (2.49).

$$obj = \left\{RMSE_s^{pr} FN_s^{pr}, FP_s^{pr}, MC^{pr} \mid s = \{TE, TR\}, pr \geq 0\right\} \quad (2.49)$$

Where $RMSE_s^{pr}$ is Root Mean Square Error for dataset s with priority pr; FN_s^{pr} is the number of False Negatives; FP_s^{pr} is the number of False Positives and MC^{pr} states the Model Complexity. TE and TR represent Test and Training sets respectively. The higher value for pr states the higher priority for the corresponding objective. Eq. (2.50) shows the formula for calculating Model Complexity.

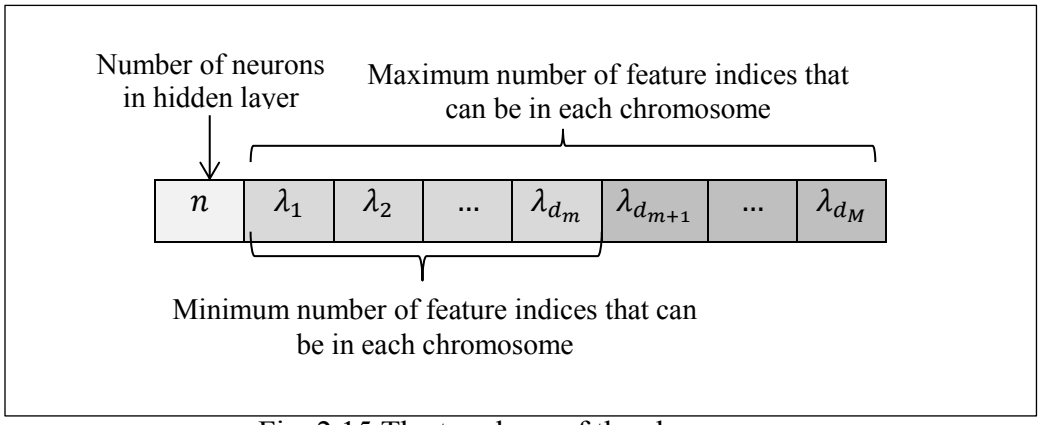


Fig. 2.15 The topology of the chromosome

$$MC = Number\ of\ input\ features \times Number\ of\ neurons\ in\ hidden\ layer \quad (2.50)$$

For evaluating the individuals in one generation, each chromosome is trained with the provided training dataset (i.e., using the features whose indices are depicted in chromosome). The Levenberg-Marquardt (LM) algorithm exploiting the linear-nonlinear relationship of the parameters is selected for training because of its higher accuracy and convergence rate. Since the result of gradient based methods for solving optimization problems, like LM, may depend on their initial parameters' values, for each individual in current generation, the training and test procedures are repeated α times.

Within these α times, the best result is picked up for determining the parameters of the individual (i.e., which are the centres, spreads and weights in RBFNs). There are $d + 2$ different ways for identifying which training trial is the best one (i.e., d is the number of objectives). The first strategy is to select the training trial which has minimized all objectives better than the others. In other

words, if we consider a d dimensional objective space, the one whose Euclidean distance from the origin is least, will be considered as the best. The green arrow in Fig. 2.16 visualize this situation for $d = 2$. In the second strategy, the average of objective values for all training trails is calculated and then the trial whose value is the closest to the average value will be selected as the best one (i.e., red arrow in Fig. 2.16).

The other d strategies are to select the training trial which minimized the i^{th} objective (i.e., $i = 1,2,\cdots,d$) quite better than the other trials. As an example, the yellow and blue arrows in Fig. 2.16 are the best training trials which minimized objective1 and objective 2 respectively.

Having trained each individual, we are now able to assign a fitness value based on the defined objectives, their corresponding priorities and restrictions, so that the population for the next generation can be produced. Moreover, as shown in Fig. 2.17, the non-dominated set is updated based on the individuals in current generation. It is expected that after a sufficient number of generations the population has evolved to achieve a non-dominated set which is not going to be altered anymore; in this stage, extracting the preferable individuals from the non-dominated surface can give us the best possible neural network models.

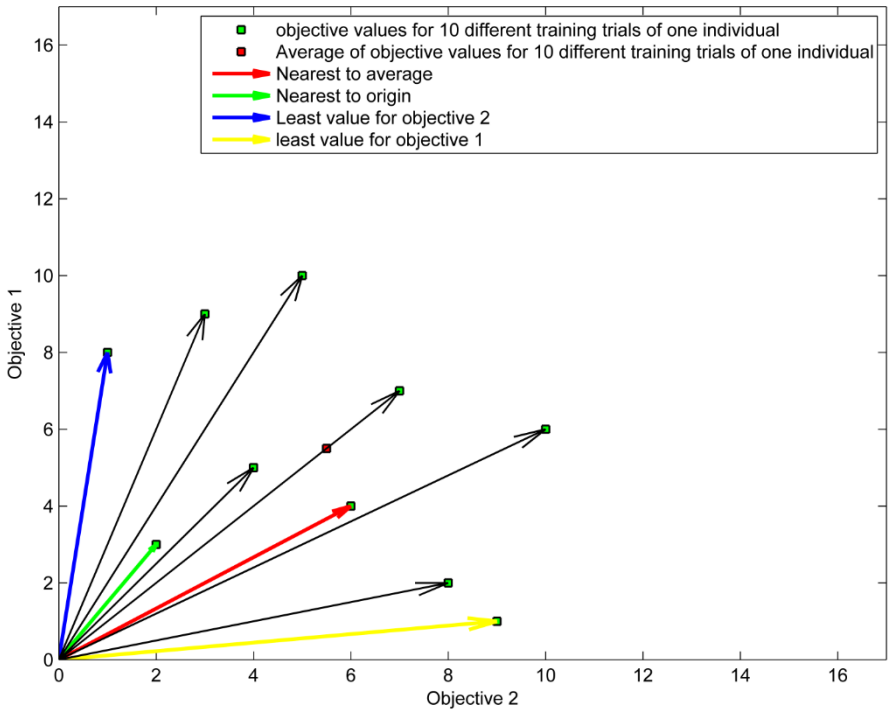


Fig. 2.16 Four different strategies for identifying the best training trial within α times training.

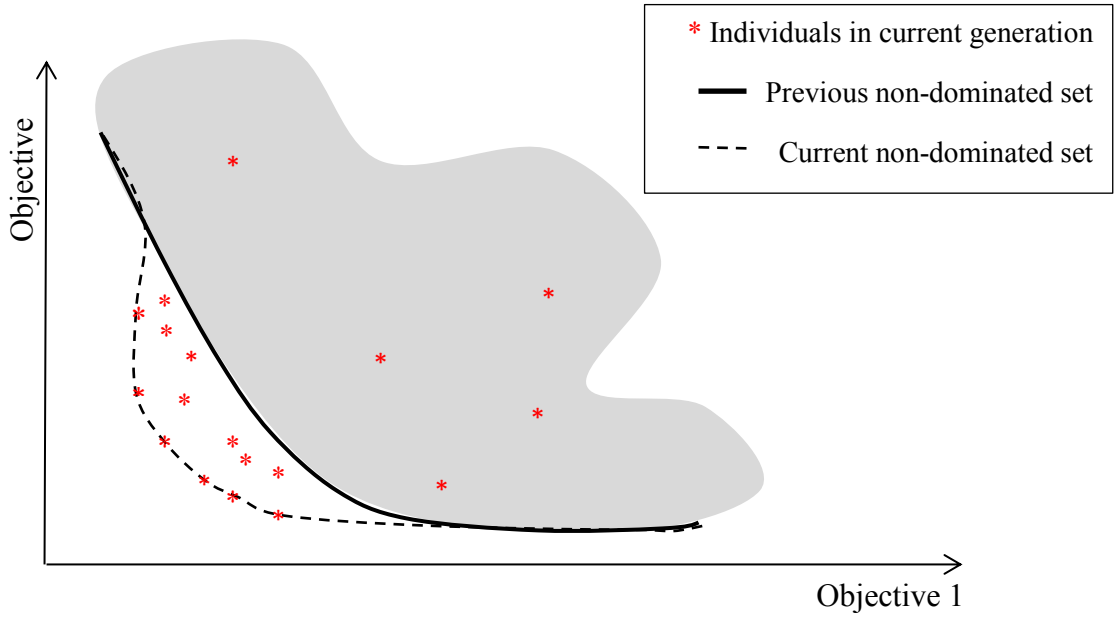


Fig. 2.17 The update of non-dominated set on arrival of new points

2.4 Active Learning

As discussed in the previous section, in order to find the best possible structure of RBF neural network and its corresponding parameters, MOGA uses the Genetic Algorithm approach. As a result, it has to execute for a sufficient number of generations to reach a point where non-dominated set has a negligible amount of alteration (this number typically can vary between 100 and 1000). Moreover, the population size of each generation should not be determined too small so the system has a better opportunity to find optimal solutions. As we can see, the system has to train a considerable amount of RBFNN structures to be able to construct the final non-dominated set (i.e., recall that the training process is done α times for each chromosome). As a result, in practice, constraints should be imposed on the size of the datasets that are provided to MOGA, otherwise the process would be very time consuming, or even impossible to implement. In this situation, if there is a large number of data samples which can be potentially used as the training dataset, an optimal way of getting the best possible solutions is to choose the most informative data samples for inclusion in the training set. This can be done through a mechanism in which the learner actively chooses more informative data samples to train as learning proceeds; this is called active learning [38]. Moreover, as stated in [39], if data samples are chosen using an active learning strategy, a

higher level of the generalization capability can be acquired. Fig. 2.18 illustrates a comparison between active and passive learning processes.

Active learning has been studied from two stand points depending on the optimality. One is the global optimality, where a set of all sample points is optimal. The other is the greedy optimality, where the next sample point to add is optimal in each step [39].

Generally, the global optimal methods give better generalization capability than the greedy optimal methods. However, the global optimal results have been obtained only for restricted cases. In contrast, the greedy optimal methods have been derived under general conditions [40].

Two main approaches to active learning can be defined, namely selective learning and incremental learning. Selective learning selects a completely new training subset from the candidate training set at each subset selection interval, based on some measure of data sample information. Each original candidate data sample is eligible for selection at each subset selection interval, regardless of whether the data sample has been selected at a previous subset selection interval. Incremental learning follows a similar approach, but with the exception that selected data samples are removed from the candidate training set, and added to the actual training set for the duration of training. The training set therefore grows during training, while the candidate training set shrinks [38].

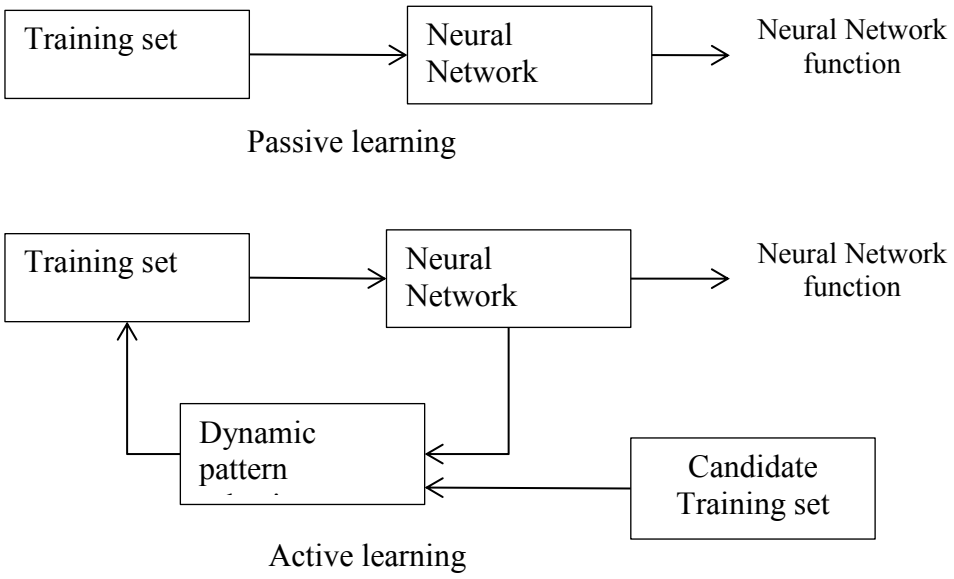


Fig. 2.18 Active learning vs. Passive learning [38]

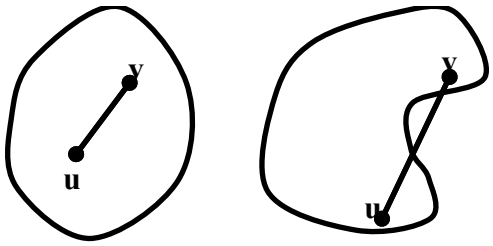
2.5 Aproxhull- a data selection approach

Neural networks and Support Vector Machines, as well as other data driven machine learning approaches, are well established methods for classification and regression tasks. Since the models generated by these approaches are data driven, selecting suitable data from large datasets for the design phase is a crucial task, as the accuracy of these models is affected by the data in the training dataset. Data must be selected in such way that it covers the whole input ranges in which the model is to be employed. Authors in [6] proposed a randomized approximation convex hull algorithm, Approxhull, that can be applied as a method for data selection for high dimensions in an acceptable execution time, and with low memory requirements. A brief overview of this method is presented in this section since it will be used in chapter 6 for selecting the most proper data to be included in the training set.

2.5.1 Convex hull definition

From a computational geometry's point of view, an object in Euclidean space is convex if for every pair of points within the object, every point on the straight line segment that joins them is also within the object. A set $\boldsymbol{S}$ is convex if, for every pair, $\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{S}$, and all $\boldsymbol{t} \in [0,1]$, the point $(1-\boldsymbol{t})\boldsymbol{u} + \boldsymbol{t}\boldsymbol{v}$ is in $\boldsymbol{S}$ (please see Fig. 2.19). Moreover, if $\boldsymbol{S}$ is a convex set, for any $\boldsymbol{u_1}, \boldsymbol{u_2}, \dots, \boldsymbol{u_r} \in \boldsymbol{S}$, and any nonnegative numbers $\{\lambda_1, \lambda_2, \dots, \lambda_r\}$: $\sum_{i=1}^{r} \lambda_i = 1$, the vector $\sum_{i=1}^{r} \lambda_i u_i$ is called a convex combination of $u_1, u_2, \dots, u_r$. According to the definitions above, the convex envelope of set $\boldsymbol{X}$ can be defined in terms of convex sets or convex combinations as follows [6, 41]:

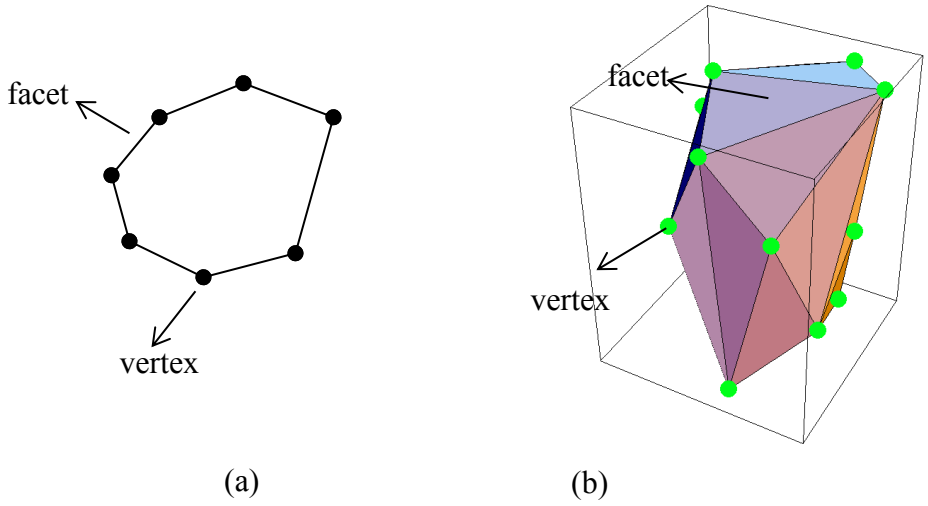
- the minimal convex set containing $\boldsymbol{X}$, or
- the intersection of all convex sets containing $\boldsymbol{X}$, or
- the set of all convex combinations of points in $\boldsymbol{X}$.



(a) (b)

Fig. 2.19 (a) represents a convex set; (b) represents a non-convex set.

Having defined the convex set of set $\boldsymbol{X}$, we will consider all data samples residing on the hull of the convex set as convex hull of set $\boldsymbol{X}$. Each data sample within this hull is called a convex point or a convex vertex. The connection between vertices is done by facets. The dimension of the facet is equal to the dimension of the dataset. For example, in a two dimensional space, convex points are connected to each other within lines (i.e., two dimensional facets) but in a three dimensional space, convex points are connected through planes (i.e., three dimensional facets) Fig. 2.20 shows vertices and facets of convex hull in two and three dimensions.



(a) (b)

Fig. 2.20 Vertices and facets of convex hull in (a) two dimensional space and (b) three dimensional space.

2.5.2 Aproxhull algorithm

Aproxhull algorithm consists of five main steps [6]: In the first step, each dimension is scaled to the range [-1, 1]. Identifying the maximum and minimum samples with respect to each dimension is done in second step. These samples are considered as vertices of the initial convex hull. A population of $\boldsymbol{k}$ facets based on current vertices of convex hull is generated in step 3. In step 4, the furthest points to each facet in the current population are identified as new vertices of convex hull, if they have not been detected before. Finally, in step 5, current convex hull is updated by adding newly found vertices into current set of vertices.

Steps 3 to 5 are executed iteratively until one of the following two termination criteria is met:

- There are no newly found vertices in Step 4

- Let $\boldsymbol{dc}$ be the maximum of approximated distances of furthest points to the current convex hull in each iteration. If there are new vertices as a consequence of Step 4 and the difference between the maximum and minimum of $\boldsymbol{dc}$ over $\boldsymbol{w}$ last iterations is less than a threshold (assume 0.1), and there is fluctuation in value of $\boldsymbol{dc}$ in this $\boldsymbol{w}$-sliding window, the algorithm ends.

Since computing the distance from a point to the current convex hull is complex and time consuming in high dimensions, the approximated distance of a newly found vertex to the current convex hull is computed based on $\boldsymbol{2 \times d}$ vertices which are nearest neighbors to the newly found vertex in the current convex hull, where $\boldsymbol{d}$ denotes the dimension.

2.6 Classification of imbalanced data sets

The class imbalance problem is intrinsic to some application domains. This issue occurs when the number of data samples representing the class of interest is much lower than the ones of the other classes. For example, in medical diagnosis which is the case we are studying in this thesis, disease cases are fairly rare comparing to the normal population. Another example is fraud detection in a collection of transactions where there are many more legitimate than fraudulent cases.

Training classifiers with datasets which suffer of imbalanced class distributions is a challenging task since the classifiers will be biased to the class with higher number of data samples (i.e., the majority class). As a result, the classifier can detect data samples from the majority class quite well but the misclassification rate in the minority class will be very high. Based on [42], the three reasons for poor performance of the existing classification algorithms on imbalanced data sets are:

1. They are accuracy driven i.e., their goal is to minimize the overall error to which the minority class contributes very little.
2. They assume that there is an equal distribution of data for all the classes.
3. They also assume that the errors coming from different classes have the same cost.

In this case, it is important to select the suitable training dataset for learning from imbalanced data. The state of the art solutions for imbalanced learning are mostly based on boosting and sampling methods such as Oversampling, Undersampling, Synthetic sampling, Cluster-based sampling and their combinations [43].

In [44] the approaches are divided into three categories:

1. Algorithmic level

The algorithmic level approaches force the classifier to converge to a decision threshold biased to an accurate classification of the minority class such as by adjusting the weights for each class. For instance, a weighted Euclidean distance function can be used to classify the samples using k-Nearest Neighbours (k-NN). Similarly, a Support Vector Machine with a kernel function biased to the minority class can improve the minority class prediction. Based on [42], algorithmic level solutions are classified as follows:

1. Adjusting the costs of the various classes so as to counter the class imbalance,
2. Adjusting the probabilistic estimate at the tree leaf (when working with decision trees),
3. Adjusting the decision threshold,
4. And recognition-based (i.e., learning from one class) rather than discrimination-based (two class) learning.

2. Data level

Data level approach does not modify the existing classifiers and is applied as a pre-processing technique prior to the training of a classifier. The data set can be re-sampled by oversampling the minority class and/or under sampling the majority class. Even though being independent of the classifier seems like an advantage, it is usually hard to determine the optimal re-sampling ratio automatically. Additionally, it might be problematic to oversample the minority classes while keeping the same distribution, especially in real world applications where overlaps between minority and majority classes are highly likely. Similarly, while under sampling the majority class, it is usually difficult to keep the new distribution of the majority class similar to the original distribution.

Some over sampling methods are as follows:

1. Random over-sampling, that is, a non-heuristic method that balances the class distribution through the random replication of positive examples. Nevertheless, since this method replicates existing examples in the minority class, overfitting is more likely to occur [42, 45].

2. Synthetic Minority Over-sampling Technique (SMOTE) in which the minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbours. Depending upon the amount of over-sampling required, neighbours from the k-nearest neighbours are randomly chosen [42, 45].
3. Synthetic Minority Over-sampling Technique Nominal Continuous (SMOTE-NC) which handles mixed datasets of continuous and nominal features [42].
4. Synthetic Minority Over-sampling Technique Nominal (SMOTE-N) [42].
5. borderline-SMOTE1 and borderline-SMOTE2, in which only the minority examples near the borderline are oversampled. These approaches achieve better TP rate and F-value than SMOTE and random over-sampling methods [42, 46].

Some under sampling methods are as follows:

1. Random under-sampling approach is one simple method of under-sampling that removes the examples in the majority class randomly. Consequently, Random under-sampling approach gives a simple method to get a balanced data set. But some important majority class samples may be removed [45].
2. NearMiss-1 selects the samples in majority class whose average distances to three closest examples in minority class are the smallest [45, 47].
3. NearMiss-2 selects those majority class examples whose average distances to three farthest minority class samples are the smallest [45, 47].
4. NearMiss-3 selects a given number of the closest majority examples for each minority example to guarantee that every minority example is surrounded by some majority examples [45, 47].
5. Most distant selects the majority class samples whose average distances to the three closest minority class examples are the farthest [45, 47].

3. Costs sensitive methods

As stated in [44], costs sensitive methods assign different costs to training examples of the majority and the minority classes. However, it is difficult to set the cost properly (it can be done in many ways) and may depend on the characteristics of the datasets. The standard public classification data

sets do not contain the costs and over-training is highly possible when searching to find the most appropriate cost.

2.7 Neural network ensemble

Neural network ensemble is a learning paradigm where many neural networks are jointly used to solve a problem. In general, a neural network ensemble is constructed in two steps, i.e., training a number of component neural networks and then combining the component predictions [48]. For training component neural networks, the Multi-Objective Genetic Algorithm approach, explained earlier in this chapter, is used to generate a non-dominated set of Radial Basis Functions Neural Networks. For combining the predictions of component neural networks, the most prevailing approaches are plurality voting or majority voting for classification tasks, and simple averaging or weighted averaging for regression tasks [48]. Since in this thesis we are designing RBFNN models to classify normal and abnormal pixels in brain CT images, the majority voting among the non-dominated set of RBFNN models obtained from MOGA is used to enhance the accuracy. For example, suppose that there are *n* RBFNN models in the non-dominated set of a conducted scenario in MOGA. If ⌊n/2⌋+1 of non-dominated models agree that the pixel is abnormal, the pixel will be considered as abnormal; otherwise it will be considered as normal.

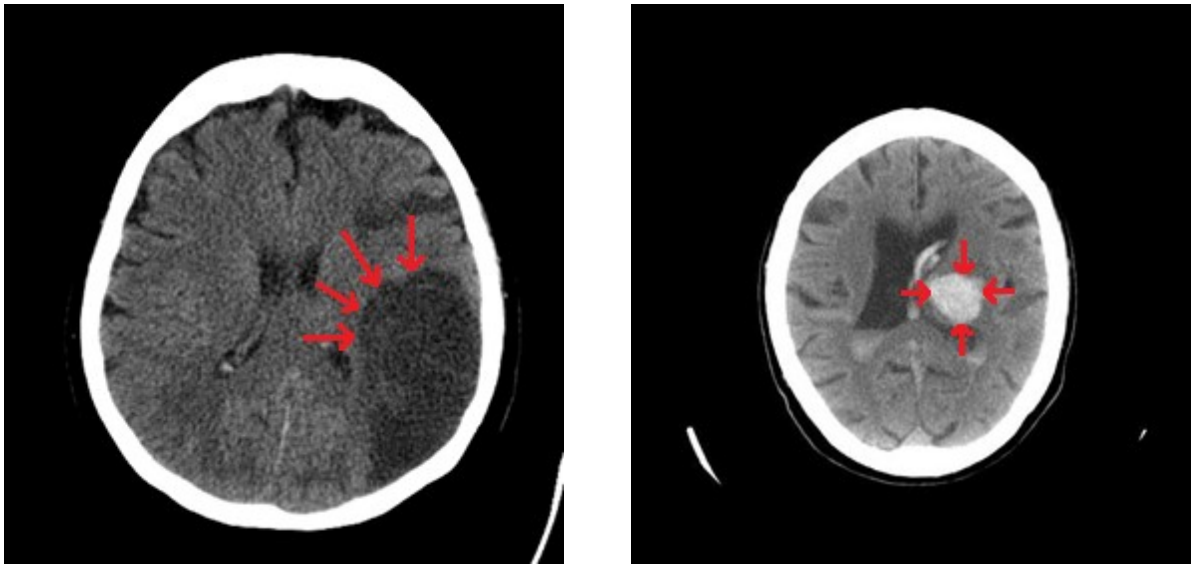
3. Medical imaging background and State of the Art

The aim of this chapter is to give a glance of the background information concerning Cerebral Vascular Accident, medical imaging techniques and the state of the art for automatic segmentation of lesions from brain tissues in already taken images.

The chapter is organized as follows: Section 3.2 gives a brief description about different types of CVA and how they appear in brain CT images. Different types of brain imaging techniques are discussed in section 3.3. Section 3.4 overviews existing medical image formats and discusses digital image representation of brain CT. Different types of artefacts that should be removed from brain CT images as a preliminary step for automatic diagnosis of CVA, as well as the corresponding most frequently used algorithms are presented in section 3.5. The problem of tilted head position in CT images and existing algorithms for realigning the images is described in section 3.6. Section 3.7 gives a brief overview of existing CAD methods for lesion detection from brain images. A review on existing textural feature extraction methods is presented in section 3.8.

3.1 Cerebral Vascular Accident

The Cerebral Vascular Accident, also called stroke, is caused by the interruption of blood supply to the brain, mainly due to a blood vessel blockage (i.e., extreme ischemia) or by an haemorrhagic event. As it can be seen in Fig. 3.1, the area of the brain that has been affected by an ischemic stroke is less dense (darker) than the normal areas in CT images. In contrast, a haemorrhagic stroke makes the affected area denser (lighter) than the normal part. The cut off of oxygen and nutrients supplies cause brain tissue irreversible damages if not detected during the first 2-3 hours. In Portugal the CVA is the leading cause of death [49], and several studies point out a prognosis of more than 80 CVA occurrences per day for the next 10 years. Stroke accounted for approximately 1 of every 19 deaths in the United States in 2009 to [50].



(a) (b)

Fig. 3.1 Brain CT slice with (a) Ischemic stroke [51]. (b) Haemorrhagic stroke [52].

3.2 Brain imaging techniques

Brain imaging techniques can be divided into two groups:

1. Structural imaging techniques, which reveal the anatomic information of the brain. Computed Tomography and Magnetic Resonance Imaging (MRI) can be placed in this category.
2. Functional imaging techniques, which visualizes changes in the blood flow or metabolism of the brain. Functional Magnetic Resonance Imaging (fMRI), Diffuse Optical Tomography (DOT) and Positron Emission Tomography (PET) are examples of functional imaging techniques.

Computed Tomography scanners use an x-ray emitter shooting x-ray beams while circulating around the region of interest (e.g., the brain). There is a digital x-ray detector in the opposite side of the emitter to capture the information from the already emitted x-ray beams. After a full rotation of the x-ray emitter, a software tool performs numerical integral calculations on the x-ray series obtained from different angles, and produces a 2-dimentional image which is one slice of the CT images. In order to produce the next slices the patient is moved forward incrementally and the procedure is repeated [53].

Magnetic Resonance Imaging scanners benefit from the fact that the human body is mostly composed of water molecules. MRI scanners contain very powerful magnets which make the proton particles inside the hydrogen atoms of the water molecules lined up. A radio wave is then sent to a certain area of the body (e.g., the brain), which makes the protons of that area out of alignment. As the radio waves are turned off, the protons send out radio signals while trying to realign themselves. These signals can specify the exact location of the corresponding protons. Moreover, since the speed of realignment of protons is different depending on the tissue they belong to, distinct signals are produced. As a result, the signals from the protons make a detailed image of the region of interest [54].

Functional Magnetic Resonance Imaging is capable of showing brain activity on MRI images. fMRI benefits from the fact that the protons in the oxygenated blood produce stronger signals than the ones which are located in the blood that has already released its oxygen. Since fMRI is very sensitive to oxygen usage in blood flow, it can detect the abnormal low blood flow in the brain indiciating an ischemic stroke.

Diffuse Optical Tomography emits near-infrared light through a laser. Detectors that are composed of optical fibre bundles are positioned a few centimetres away from the light source. These detectors monitor the path alteration of the emitted light, either through absorption or scattering. Regarding the brain, absorption reveals information about chemical concentration in the corresponding area and scattering shows the physiological characteristic such as the swelling of the neuron upon activation. [55]. This modality can be used for detecting ischemic and haemorrhagic strokes [56].

Positron Emission Tomography uses nuclear medicine to produce images. For taking images from a specific tissue, a radioactive medicine is attached to natural chemical substance that is normally used by that tissue (e.g., glucose for the brain). The combination of the natural chemical substance and the nuclear medicine is called a radiotracer which is injected in the human body. The radiotracer goes to those parts of the body that consume the natural chemical part for their metabolism. As radiotracer is broken down in the tissue, positrons (i.e., an antiparticle of electron with opposite charge) are emitted. Each positron collides with an electron in the tissue and produces a pair of gamma photons. The PET camera has specific crystals which can detect and absorb

gamma photons and produce light that is subsequently converted into an electrical signal [57]. [58, 59] discuss the use of PET for detecting haemorrhagic and ischemic strokes.

Table 3.1 briefly lists some properties of imaging modalities that are used for brain screening.

Table 3.1 Properties of brain imaging modalities

Modality	Properties
CT	1. Uses ionizing radiation 2. Detects the attenuation of emitted x-ray signals 3. Available in most emergency units 4. Faster than MRI and as a result less sensitive to patient movement 5. Cost effective
MRI/ fMRI	1. Uses powerful magnetic field and radio waves 2. Detects radio wave emitted from the protons inside the hydrogen atoms of tissue 3. Limited availability in emergency units 4. Cannot be used for people with metal implants, cardiac pacemakers or any ferromagnetic material that can be affected by the strong magnetic field. 5. Not suitable for patient with severe bleeding since blood clots will become a tissue which is difficult to distinguish from the normal tissue [7]. 6. Costly 7. Provides more detailed information rather than CT while imaging a soft tissue.
PET	1. Use radiotracer (the amount of radiation is the same as in most CT scans) 2. Detects the gamma photons emitted after collision of the positrons and electrons 3. Costly 4. Time consuming
DOT	1. Uses near infrared light 2. Detects path alteration of the emitted light

Modality	Properties
	3. Portable 4. Has no magnetic field 5. Is non-ionizing 6. Has low resolution and localization accuracy compared with CT and MRI 7. Capable of providing information of fast-changing processes

Since the aim of this thesis is to propose an intelligent support system that is capable of assisting experts in emergency units, only the CT imaging modality fulfils the majority of the emergency units imaging equipment so CT images are chosen as the input of the proposed system.

3.3 Digital image representation of brain CT

3.3.1 Medical image formats

Medical image files are typically composed of two parts: the metadata and the digital image. The metadata contains information about the image and is typically stored at the beginning of file as the header or in a separate file. As described in [60], medical image formats can be classified into two groups. The first group aims on standardizing images generated by imaging modalities (e.g., DICOM) while the second group intends to facilitate post-processing analysis (e.g., Analyze, Nifti and Minc). Table 3.2 describe the properties of four medical image formats.

Table 3.2 Properties of four medical image formats [60]

Format	*Header*	*extension*
Analyze	Fixed-length: 348 bytes binary format	.img & .hdr
Nifti	Fixed-length: 352 bytes binary format (348 bytes in the case of data stored as .img and .hdr). Nifti has a mechanism to extend the header.	.nii
Minc	Extensible binary format	.mnc

Format	*Header*	*extension*
Dicom	Variable length binary format	.dcm

The Analyze file format was developed by Biomedical Imaging Resource at Mayo Clinic for the Analyze software package, at the end of 1980s. Analyze stores header and image content into two separate files with the extensions of .img and .hdr respectively. Analyze format does not store any information about image orientation to prevent the left-right ambiguity in brain study [60, 61].

In the beginning of the century, a consortium of researchers from the National Institute of Mental Health and the National Institute of Neurological Disorders and Strokes developed a new format, named Nifti, in order to solve the problem of sharing data between different centres and software packages. Nifti solved the weaknesses of Analyze while maintaining its advantages (e.g., in this format, the problem of left-right ambiguity has been solved and one can detect the orientation of the image) [60, 61].

The Minc file format was designed by the Montreal Neurological Institute. The aim was to provide a modality-independent and a flexible way of storing medical images. The first version of the Minc format (Minc1) had three main drawbacks including: 1- The lack of support for files larger than 2 gigabytes; 2- The need to allow a single medical image file to contain data at several levels of resolution and 3- The need for internal, transparent data compression. To fix these shortcomings the second version of Minc file format (Minc2) was developed [60].

The Dicom standard was introduced by the American College of Radiology and the National Electric Manufacturers Association in 1985. Using Dicom standard enables different imaging modalities, workstations, printers, scanners and network hardware to communicate with each other through the Picture Archiving and Communication System (PACS). In the Dicom standard, the image header contains not only information about the image matrix, but also the most complete description about the whole procedure of image acquisition including scanning parameters. The header also contains patient information such as name, age, weight, height, gender, etc. In other words, the Dicom standard makes each image to be completely self-descriptive [7, 60], and is the widest standard encountered in Portugal, therefore it will be the one to be used on this thesis

3.3.2 Brain CT representation

Brain CT images are represented by a 512 x 512 matrix within a Dicom file. Each element of the matrix represents the absorbed amount of x-radiation in terms of Hounsfield Units (HU). Table 3.3 shows Hounsfield Units of brain tissues in CT images. In order to describe the location of each pixel, a Cartesian coordinate system whose origin is located in the top left corner of the image is considered. In this system, the x and y axis are extended from the origin to the right and downwards respectively.

Table 3.3 Hounsfield Units of brain tissues in CT images [62]

Tissue	Relative attenuation values (in Hounsfield Units)	Appearance on CT
Metal	1000	White
Bone/calcium	100-1000	
Blood		
Acute	80-85	
Subacute	25-50	
Chronic	0-25	
Gray matter	35-40	
White matter	25-30	
Water	0	
Fat	-100	
Air	-1000	Black

3.4 Artifacts

For detecting CVA abnormalities from head CT slices we have to focus on the intracranial part of the images. The Intracranial part is the one that is inside the skull. Other parts including the scalp, the skull and the U-shaped head holder are considered as artifacts and should be removed. Moreover, those slices which have been taken from the lower part of the head have too much noise from other organs like the eyes and the nose and contain a very small portion of intracranial area. This kind of slices is not also very suitable for CVA detection (Please see Fig. 3.2, slices 1-6).

Given a CT slice, the authors in [63] first produce a mask using a proper threshold T. This means that all pixels with intensity lower than T are set to zero and the intensity of all other pixels is set to one. In order to omit the noise around the head (i.e., due to the scalp) and to make the boundary more accurate, a mathematic morphological field transformation is done. For getting the left-right and top-bottom boundaries of the skull and omitting the noise resulted from the U-shaped head holder, a vertical/horizontal projection curve is used. The mask is then multiplied pointwise with the original grayscale image to produce the head image without noise. This method does not eliminate the skull bones.

In [64], in order to divide a given CT slice into intracranial, skull and extra-cranial regions, the gray level histogram of the image is obtained. Three distinct peaks representing air, brain and skull bone are then separated by thresholding. The skull is recognized by finding the largest region after applying the region-growing algorithm on all pixels with bone density. The method then applies a region growing algorithm, for the second time, on pixels with brain density at the centre of the skull image for identifying the intracranial region.

The method in [65] used a modified global thresholding algorithm to preserve the calcification inside the brain. The algorithm, first transforms the CT image into a binary image with pixel values (0, 1) using a global threshold. The region of calcification that has less than a certain number of pixels, say 500 square pixels are then removed. The binary image is then multiplied pointwise with the original grayscale image to remove the skull. [66] also uses global thresholding for skull removal process.

In [67], authors apply Algorithm 3.1 for artifact removal. This method is used in this thesis for removing artifacts from brain CT images. By applying Algorithm 3.1 on raw CT images of Fig. 3.2, 11 out of 20 CT slices were selected as the suitable ones for further processing (i.e., CT slices 9-19). The result can be seen in Fig. 3.4.

Algorithm 3.1 Artifact removal algorithm in brain CT images [67]

1. Skull detection:
   1.1. Remove pixels whose intensities are less than 250.
   1.2. Use the Connected Component algorithm [68] to choose the largest component as the candidate skull (Fig. 3.3-b).

1.3. Remove the small holes within the candidate skull by inverting the matrix of candidate skull and applying the Connected Component algorithm for the second time. Those connected components whose area are less than 200 pixels are considered as holes and will be filled by 1 (Fig. 3.3-c).

2. Removing slices with either unclosed skulls or skull containing too many separate regions:
   2.1. Having completed step 1.3, we have already all connected components at hand. As a result, we can count the number of big holes (i.e., holes whose area is larger than 200 pixels). If the number of big holes is equal to 2, it will be considered as closed skull; otherwise the slice will be removed from the desired set.
3. Intracranial area detection:
   3.1. All CT images that successfully passed step 2.1, contain only two black regions which are separated by the skull. In order to detect which black area is related to intracranial part, the centre of mass of the skull is calculated. The region which contains the centre of mass will be considered as intracranial area (Fig. 3.3-d).

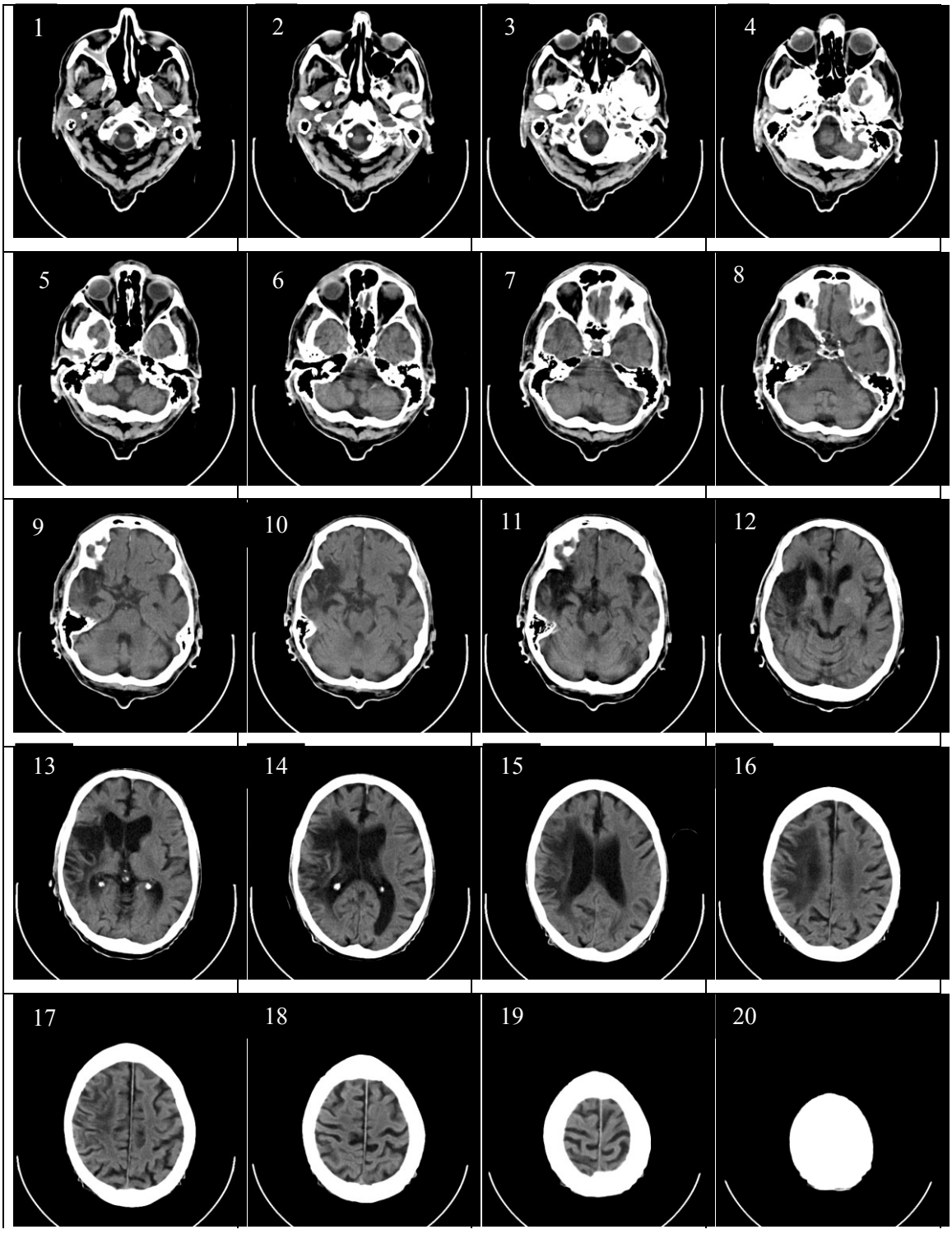


Fig. 3.2 Raw CT slices from one patient's head CT

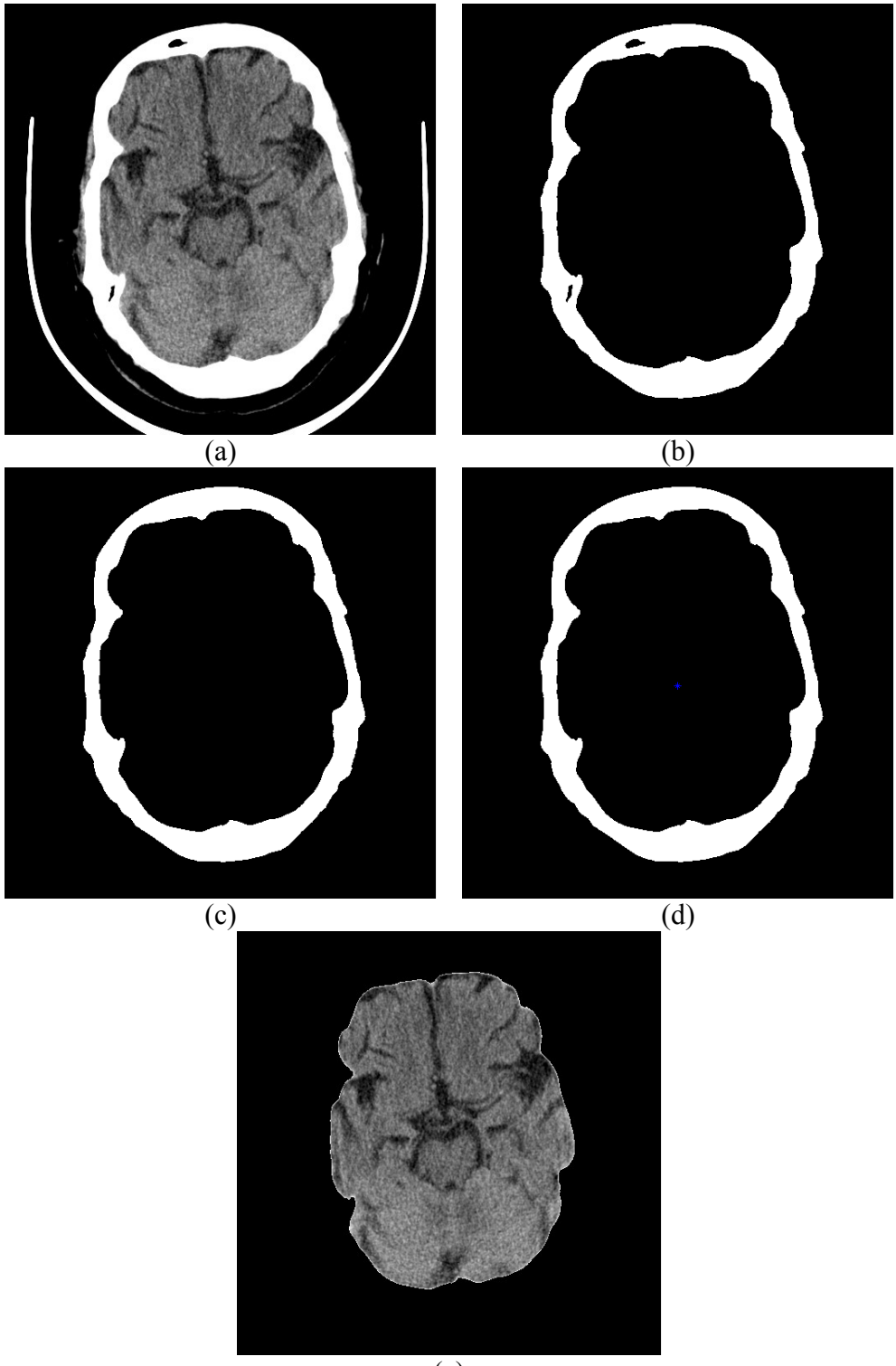


Fig. 3.3 Artifact removal process proposed in [67]. (a) The original image. (b) The largest connected component is selected as the skull after applying the threshold. (c) Small holes are filled. (d) The centre of mass of the skull is found (blue point). (e) The cranial part is filtered based on the centre of mass location.

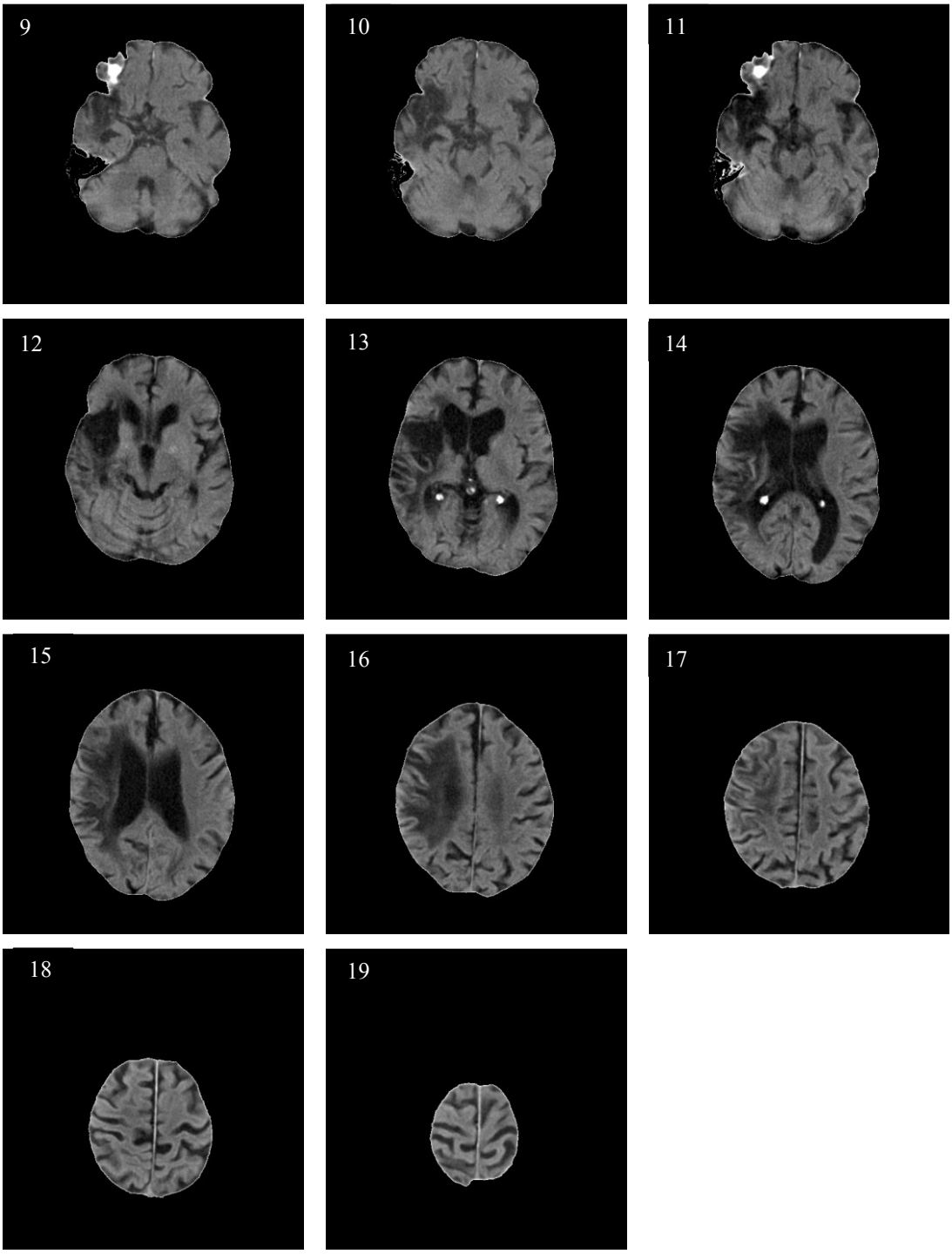


Fig. 3.4 Applying Algorithm 3.1 on raw CT images of Fig. 3.2

3.5 The problem of tilted images

Tilted head position in CT images can be either as a part of clinical process (e.g., for reducing beam-hardening artifacts in patients with aneurysm clips [69]) or as a result of patient movement during imaging process. In order to extract symmetry features we need to detect the actual midline of the brain and rotate the tilted images to make the actual midsagittal line perpendicular to the x-axis (i.e., the Cartesian coordinate system within which an image is presented, is described in subsection 3.5). The term "actual" refers to the position of midline when the brain is normal with no pathology.

The authors in [70] tried to detect the actual midline of the brain through the position of ventricles. The main idea is that if the system can detect the right and left ventricles within a CT slice, the actual midline is believed to be at the middle of the mass centres of the two ventricles. In order to find the CT slices that contain the ventricles, they consider a fixed window of size 140*200 in the upper centre area of the brain. K-means clustering is then used to distinguish ventricle pixels from all other pixels within the window. The algorithm then selects the first three slices with the highest percentage of ventricle pixels within the window. In order to be able to locate the mass centres of ventricles within the three candidates, the algorithm finds the contour of ventricles using the level set method. The idea behind level set (also known implicit active contours, or implicit deformable models) for image segmentation is quite simple. The user specifies an initial guess for the contour, which is then moved by image-driven forces to the boundaries of the desired objects. In such models, two types of forces are considered - the internal forces, defined within the curve, are designed to keep the model smooth during the deformation process, while the external forces, which are computed from the underlying image data, are defined to move the model toward an object boundary or other desired features within the image [71].

In [72], the authors use the location of ventricles also in order to estimate the actual midline within a CT slice. This method first identifies the edge points between the bilateral ventricles and then uses the average of the left side mean and the right side mean of the x-coordinates to define the x-coordinate of the midline. In order to find the edge points of the ventricles, they use a point-based shape matching approach, named shape context method which matches the ventricles' edge points of the CT slice at hand to the ventricle templates from MRI images.

The method that is used in this thesis is the one that is presented in [67, 73] and is summarized in Algorithm 3.2. In order to align tilted CT slices, a rotation is done around the mass centre of the skull. Fig. 3.5 shows the result of applying Algorithm 3.2 on raw CT images presented in Fig. 3.2.

Algorithm 3.2 Ideal midline detection of the brain CT images [67, 73]

1. Use Algorithm 3.1 to remove artifacts from brain CT images.
2. Since the concave shape of intra cranial region will affect the accuracy of search for finding the ideal midline, CT slices with high amount of concavity are found and excluded.
   2.1. For each CT slice
      2.1.1. Extract the contour of intracranial region.
      2.1.2. $Concavity = 0$
      2.1.3. For $\emptyset = 0$ to 180
         2.1.3.1. Rotate the contour by $\emptyset$ degree.
         2.1.3.2. $Concavity_{\emptyset} = 0$
         2.1.3.3. For $i = 1$ to $number\ of\ rows$
            2.1.3.3.1. Scan the pixels of the contour in row i and define the Far Left (FL_i) and Far Right (FR_i) junctions.
            2.1.3.3.2. Let C_i be the number of pixels in row i which resides between FL_i and FR_i and is **not** located inside the intracranial region.
            2.1.3.3.3. $Concavity_{\emptyset} = Concavity_{\emptyset} + C_i$
         End for
         2.1.3.4. $Concavity = Concavity + Concavity_{\emptyset}$
      End for
   End for
   2.2. Sort the CT slices based on their corresponding $Concavity$ values and select the first λ slices with the least amount of concavity.
3. In order to find the line that maximizes the symmetry of the resulting halves, a rotation angle search around the mass centre of the skull is performed:
   3.1. For each CT slice remained from step 2:
      3.1.1. Let θ be the maximum angle that a given CT image can be tilted.
      3.1.2. Let S_j be the symmetry cost at angle j
      3.1.3. For $j = -\theta$ to θ
         3.1.3.1. Calculate $S_j = \sum_{i=1}^{n} |l_i - r_i|$; where n is the number of rows in the current CT slice, l_i and r_i are the distances between the current approximate midline and the left and right side of the skull edge in row i, respectively.
      End for
      3.1.4. Select the rotation angle j whose symmetry cost S_j is minimum.

End for

3.2. The final rotation degree for all CT slices is determined as the median value of rotation angles obtained for each CT slice.

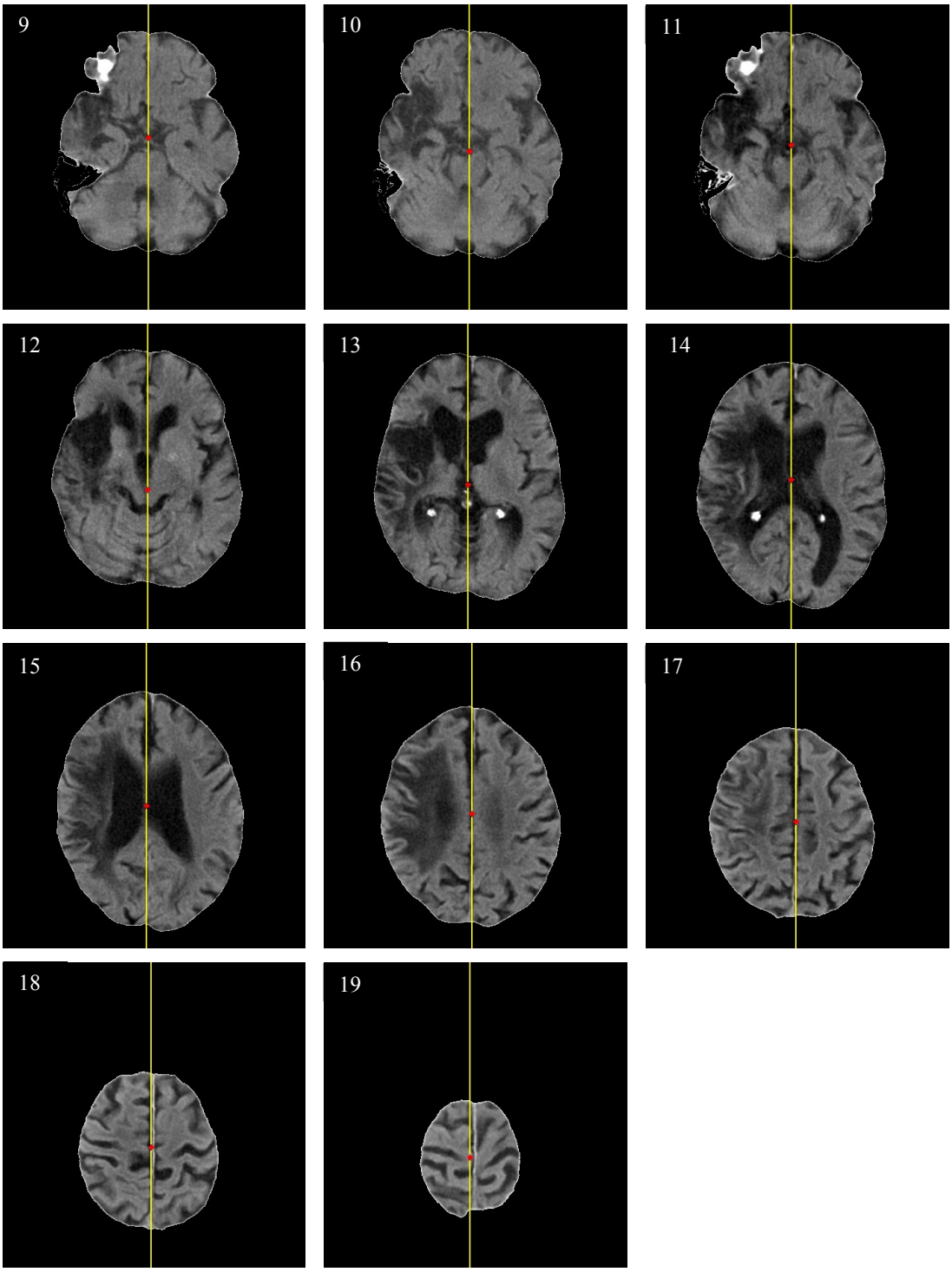


Fig. 3.5 Applying Algorithm 3.2 on raw CT images of Fig. 3.2. The yellow line is the ideal midsagittal line after rotating CT slices. The red point is the mass centre of the skull.

3.6 A review on existing computer aided detection methods for CVAs

As stated in [74], typically lesion segmentation strategies are divided into two subgroups: supervised and unsupervised strategies. Supervised approaches are those based on using some kind of a priori information or knowledge to perform the lesion segmentation.

The group of supervised strategies can be further subdivided into two sub-groups:

- In the first subgroup, all approaches use atlas information and therefore require the application of a registration process to the analyzed image to perform the segmentation.

As an example, the authors in [75] used a combination of two techniques for brain lesion detection from CT perfusion maps: finding asymmetries among the two hemispheres and then comparing the captured images to a brain atlas anatomy. For generating the asymmetry map, first the symmetry axis is approximated as the straight line that minimizes the least square error between all centers of masses' coordinates and then the intensity values of the corresponding pixels on the left and right side of the image are compared; those with a significant difference are considered as potential lesions. In order to perform a detailed description of lesions a second step is required, where position image registration of brain template is made. The goal of the registration algorithm is to maximize the similarity function between the template image and the newly acquired image.

The work done in [76] can also be considered in this subgroup. This study presents an automated template-guided algorithm for the segmentation of ventricular Cerebrospinal Fluid (CSF) from ischemic stroke CT images. In the proposed method, the authors use two ventricular templates, one extracted from a normal brain (VT1) and the other built from several pathological scans (VT2). VT1 is used for registration and VT2 to define the region of interest. In the registration process, they use the Fast Talairach Transformation [77], which takes care of the "tilting" angle. Automatic thresholding is applied on a slice-by-slice basis to cater for variability of CSF intensity values across the slices in the same scan. The distributions of the CSF, White Matter (WM) and Gray Matter (GM) are analyzed and only voxels in the CSF range and WM range are used in the calculation of the histogram employed by the Otsu's automatic thresholding algorithm [78]. Finally, artifacts are removed with the help of VT2.

- In the second subgroup, within which the work presented in thesis lies, all approaches perform an initial training step on features extracted from manually segmented images annotated by Neuroradiologists. Different classifiers, such as Artificial Neural Networks (ANNs), k-Nearest Neighbors, AdaBoost, Bayesian classifiers or decision trees, alone or combined, have been used to perform the segmentation.

The method applied in [79] is an example of this subgroup which first uses morphology operations and wavelets based filter for denoising. Asymmetric parts of the brain and their neighbors are then extracted as the region of interest for specifying relevant features such as: texture, contrast, homogeneity, etc. Finally, k-means clustering and Support Vector Machines are used for classification and provide the contour of the brain lesion (a tumor in their case).

The work done in [80] uses a wavelet based statistical method for classification of brain tissues into normal, benign and malignant tumours. The authors first obtain the second level discrete wavelet transform of each CT slice. The Gray Level Co-occurrence Matrix (GLCM) is then calculated over the low frequency part of the transformed image. Finally, features are calculated from the GLCM matrix. They use genetic algorithms and principle component analysis for feature selection and SVM for classification.

In [81] a computer tomography brain image analysis system is designed with four phases: enhancement, segmentation, feature extraction and classification. The enhancement phase reduces the noise using an Edge-based Selective Median Filter (ESMF); the segmentation phase extracts the suspicious region applying a modified version of a genetic algorithm; the feature extraction phase extracts the textural features from the segmented regions and the classification phase classifies the image. To diagnose and classify the image, the authors used a Radial Basis Functions classifier.

With regard to the unsupervised strategies, where no prior knowledge is used, two different sub-groups can also be identified:

- A sub-group of methods that segment the brain tissue to allow lesion segmentation. These approaches usually detect lesions as outliers on each tissue rather than adding a new class to the classification problem. The works presented in [82] and [83] follow this strategy.

- The other sub-group that uses only lesion properties for segmentation. These methods directly segment the lesions according to their properties, without providing tissue segmentation.

As an example, in order to segment the hemorrhage in brain CT images, the authors in [84] proposed a modified version of distance regularized level set evolution technique, which was originally proposed in [85]. The proposed method involves four stages: 1. Preprocessing which includes filtering and skull removal. 2. Segmentation using distance regularized level set evolution method. 3. Post-processing (to remove false positives and reduce false negatives) and 4. Validation, which includes comparison with the ground truth and the calculation of statistics.

In [86], an automatic method for tumor region detection and segmentation in brain Magnetic Resonance Images is suggested. The proposed method includes three main phases: preprocessing, automatic seed point selection and tumor segmentation with an improved fuzzy algorithm. In order to automatically select the seed point, the following three characteristic of tumor tissues are considered: 1. the intensity of tumors' pixels differs from the intensity of normal tissues (either higher or lower). 2. Within a tumor region, the difference between the intensity of each pixel and the mean value of tumor area is minimal. 3. Because intensity differences in homogeneous parts of the tumor tissue are low, the possibility of existing edge points in this section is very small. Therefore, counting the number of edge points around each pixel with radius R is used as another feature in the seed point selection.

The works described in [87, 88] also belong to this category.

Another approach to MR images' lesion segmentation is reported in [89]. In this paper segmentation methods are classified into four groups: (1) data-driven, (2) statistical, (3) intelligent, and (4) deformable models. In addition, some methods have the potential to be categorized to multiple categories. For these methods, the category that has more relevance to the core algorithm implementing the method is usually considered.

If a method concerns thresholding, region growing, and other spatial approaches, it is categorized into the data-driven group. Data-driven methods generate the lowest accuracies. In this category, the thresholding methods do not consider the overlaps among the intensity ranges of different tissues and do not therefore benefit from the spatial information; also, the selection of appropriate thresholds can be complicated. Comparison of different thresholding methods show the advantage

of using adaptive thresholding with respect to other common thresholding methods, though it needs more processing time. The region growing and the edge detection methods work based on the gradient of the intensities and thus are very sensitive to noise. Successful region growing, however, requires precise anatomical information to locate single or multiple seed pixels for each region.

In case a method is based on the estimation of probability density functions, it is classified into the statistical group. Statistical methods can be grouped into two main categories: non-parametric probability map model-based techniques and parametric techniques. These two kinds of techniques are combined to generate combinational techniques.

Statistical methods such as the Expectation Maximization (EM) algorithm [90] and non-parametric methods such as Parzen window [91] and kNN [92] are commonly used for image classification. In brain MRI segmentation applications, a disadvantage of the EM algorithm is the assumption of normal distributions for the intensity variations of the brain tissues, which is almost inaccurate especially when brain tissues present lesions. kNN suffers from the excessive calculation time which severely affects the training stage. Individual parametric statistical classifiers such as EM and Adaptive Mixtures Method (AMM) [93] or individual non-parametric kNN and Parzen window methods estimate non-Gaussian probability density functions with poor results. However, combining EM and kNN by assuming non-Gaussian density functions for the CSF and the multiple sclerosis lesions and Gaussian density functions for other tissue classes such as GM and WM can remarkably raise the segmentation performance.

If the method involves fuzzy logic and/or neural networks, it is categorized into the intelligent group. Publications using intelligent methods for the segmentation of brain images were also reviewed. ANN [94], Fuzzy C-Means (FCM) [88], fuzzy connectedness [95], fuzzy inference systems (FIS) [96] are commonly used methods in this category. ANN presents a good accuracy but it needs a good estimate of the number of layers and the number of nodes in each layer. Also, excessive training time is another handicap in this type of classifiers. FCM was shown to be superior on normal brain images, but worse on abnormal brain images. A shortcoming of FCM is its over-sensitivity to noise, which is also a flaw of many other intensity-based segmentation methods. However, combining the fuzzy segmentation methods with some statistical knowledge such as statistical atlases or probability maps can significantly enhance the FCM performance in

the MS detection. Fuzzy connectedness and FIS are the two other intelligent methods which present high accuracy, while FIS needs a good clinical knowledge to tune the right rules.

In case the method concerns volume estimation and also shrinking or increasing of the estimated volume, it is classified into the deformable group. Deformable techniques usually benefit from matching the MR images with an atlas, to locate the lesions. The philosophy of these methods is choosing a seed voxel of lesions manually. Thus, this selection should be based on an anatomical and biological knowledge of lesion growth. To automate these methods, a combination of deformable contours and FCM techniques was implemented, but still the performance is lower than that of the statistical methods.

Since the majority of the methods, including the one presented in this thesis, need textural features for the purpose of lesion segmentation, a review on existing textural feature extraction methods is presented in section 3.7.

3.7 A review on textural feature extraction methods

As described in [97] image texture gives us information about the spatial arrangement of colour or intensities in an image or in a selected region of an image. For natural textures which are composed of patterns with irregular sub-elements, statistical texture analysis is a practical approach. Statistical texture analysis sees an image texture as a quantitative measure of the arrangement of intensities in a region and analyses the spatial distribution of gray values. The reason behind this is the fact that the spatial distribution of gray values is one of the defining qualities of texture [98].

Statistical analysis is done by computing local features at each point in the image, and deriving a set of statistics from the distributions of the local features.

Depending on the number of pixels defining the local feature, statistical methods can be classified into first order (one pixel), second-order (two pixels) and higher-order (three or more pixels) statistics. Given an image matrix $I$of size $M \times N$, eq. (3.1)-(3.18) list some first-order grayscale features. The basic difference between first-order and higher order statistics is that first-order statistics estimate properties of individual pixel values, ignoring the spatial interaction between image pixels, whereas second- and higher order statistics estimate properties of two or more pixel values occurring at specific locations relative to each other [98]. For example, consider Fig. 3.6.a

and 3.6.b which are totally different images each with a 50% black and 50% white pixels that results in having the same gray level histogram. Therefore, we cannot distinguish between them using just first order statistical analysis. By computing the Gray Level Co-occurrence Matrix of the image, defined later, one can capture numerical features which consider spatial relations of similar gray tones.

(a) (b)

Fig. 3.6 (a) and (b) are two different images with same first order statistics features [97]

$$I(x,y) \tag{3.1}$$

Where $I(x,y)$ is the intensity value of the pixel located at (x,y).

$$x/512 \tag{3.2}$$

The feature stated in eq. (3.2) determines how much the location of the pixel under the study tends to the left or right. Since brain CT images in DICOM format have a dimension of 512×512 pixels the value of $x/512$ will be within range $[0,1]$ in which 0 and 1 values represent the most left and the right locations respectively.

$$\text{Mean} = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} I(x,y) \tag{3.3}$$

$$\text{Variance} = \sqrt[2]{\frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} (I(x,y) - Mean)^2} \tag{3.4}$$

$$\text{Skewness} = \frac{1}{Variance^{3}} \sum_{x=1}^{M} \sum_{y=1}^{N} (I(x,y) - Mean)^3 \tag{3.5}$$

$$\text{Kurtosis} = \frac{1}{Variance^{4}} \sum_{x=1}^{M} \sum_{y=1}^{N} (I(x,y) - Mean)^4 \tag{3.6}$$

$$\text{Energy} = \sum_{l=1}^{L}\left(\frac{H_l}{M\times N}\right)^2 \tag{3.7}$$

Where H_l is the l^{th}bin of the L gray level histogram H of image matrix I.

$$\text{Entropy} = -\sum_{l=1}^{L}\frac{H_l}{M\times N}\log_2\left\{\frac{H_l}{M\times N}\right\} \tag{3.8}$$

$$F_n = \frac{1}{a^2}\sum_{i=5(n-1)}^{5(n-1)+4} H_i \qquad n = 1,2,\dots,10 \tag{3.9}$$

Where H_i is the i^{th}bin of $L = 50$ gray level histogram H (i.e., considering the conditions stated in (3.9), $i \in [0\ 49]$) and a is the width of sliding window. This formula produces 10 features $F_1, F_2, F_3, \dots, F_{10}$ from the histogram. Each one of the ten features is the possibility of occurrence of 5 contiguous gray levels [99]. As previously said equations (3.3)-(3.8) represent first-order statistics, equation (3.9), although based on a first-order statistic is not classified as so, neither it fits on higher order statistics classifications.

In eq. (3.10)-(3.18) w refers to a window of size 31×31 pixels since based on [99, 100], features in smaller window sizes could not recognize CVA very well (i.e., The resolution of each CT slice is 512 × 512 pixels and the intensity value of each pixel is an integer in the range [0 255], where 0 is completely black and 255 is completely white). w is centered at the pixel (x, y) marked by a clinical expert as normal or CVA. Given w centred at point (x, y), L_h is a row vector with the pixel intensities of the 31 pixels taken from the horizontal line centered at (x, y) and L_v is a column vector with the pixel intensities of the 31 pixels taken from the vertical line centered at (x, y) [100].

$$\underset{m,n\,\in\, w}{\text{mean}}\, I(m,n) \tag{3.10}$$

$$\underset{m,n\,\in\, w}{\min}\, I(m,n) \tag{3.11}$$

$$\underset{m,n\,\in\, w}{\max}\, I(m,n) \tag{3.12}$$

$$\underset{m,n\,\in\, w}{\text{median}}\, I(m,n) \tag{3.13}$$

$$\text{std}_w = \left(\frac{1}{width(w)\times height(w)-1}\times\sum_{m=x-\frac{(height(w)-1)}{2}}^{x+\frac{(height(w)-1)}{2}}\sum_{n=y-\frac{(width(w)-1)}{2}}^{y+\frac{(width(w)-1)}{2}}(I(m,n)-\underset{m,n\,\in\, w}{\text{mean}}\, I(m,n))^2\right)^{1/2} \tag{3.14}$$

$$\underset{m,n \in w}{\text{average}}\, I(m,n) - \underset{1 \le m \le M,\ 1 \le n \le N}{\text{average}}\, I(m,n) \tag{3.15}$$

$$I(x,y) - \underset{1 \le m \le M,\ 1 \le n \le N}{\text{average}}\, I(m,n) \tag{3.16}$$

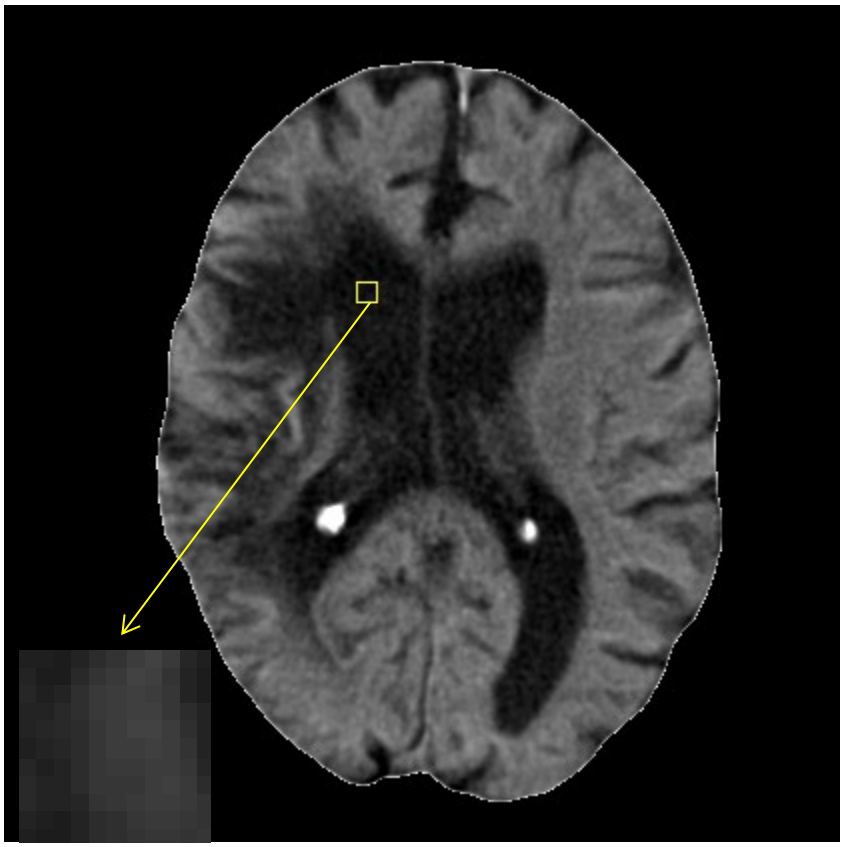
$$\text{plh} = \sum_{n=y-\frac{(width(w)-1)}{2}}^{y+\frac{(width(w)-1)}{2}} |\mathrm{L_h}(\mathrm{x}, \mathrm{n}+1) - \mathrm{L_h}(\mathrm{x}, \mathrm{n})| \tag{3.17}$$

$$\text{plv} = \sum_{m=x-\frac{(height(w)-1)}{2}}^{x+\frac{(height(w)-1)}{2}} |\mathrm{L_v}(\mathrm{m}+1, \mathrm{y}) - \mathrm{L_v}(\mathrm{m}, \mathrm{y})| \tag{3.18}$$

A co-occurrence matrix is a two-dimensional matrix C in which both rows and columns represent a set of possible gray tones in the range $[v_1\ v_2]$, (i.e., v_2 usually varies from 8 to 256 [101]. For example, authors in [7, 65] considered $v_1 = 1\ and\ v_2 = 8$). Since the intensity value of each pixel in a grayscale image is a discrete value in range [0 255], a scaling function is needed to scale pixel intensities into range $[v_1\ v_2]$ which is shown in eq. (3.19).

$$i_{scaled} = round\left(\frac{v_2 - v_1}{255-0} \times i + \left(v_1 - \frac{v_2 - v_1}{255-0} \times 0\right)\right) = round\left(\frac{v_2 - v_1}{255} \times i + v_1\right) \tag{3.19}$$

Where i and i_{scaled} are the intensity of a pixel before and after scaling, respectively. Fig. 3.7 shows the intensity values of part of a brain image and the corresponding scaled values in range [1 8].



(a)

31	31	28	35	44	51	60	66	57	38	31
35	28	28	41	51	57	66	63	51	35	28
38	31	35	47	57	60	63	60	54	35	28
35	35	41	47	57	60	63	63	54	44	35
35	38	41	44	54	60	66	66	60	47	35
31	35	38	44	54	60	60	63	60	51	35
38	31	35	44	54	57	57	60	60	54	38
41	35	35	44	54	54	54	60	63	57	44
41	35	35	38	51	54	57	60	60	57	51
35	31	28	35	44	47	54	57	57	54	51
31	28	28	35	44	47	51	54	54	51	54

(b)

2	2	2	2	2	2	3	3	3	2	2
2	2	2	2	2	3	3	3	2	2	2
2	2	2	2	3	3	3	3	2	2	2
2	2	2	2	3	3	3	3	2	2	2
2	2	2	2	2	3	3	3	3	2	2
2	2	2	2	2	3	3	3	3	2	2
2	2	2	2	2	3	3	3	3	2	2
2	2	2	2	2	2	2	3	3	3	2
2	2	2	2	2	2	3	3	3	3	2
2	2	2	2	2	2	2	3	3	2	2
2	2	2	2	2	2	2	2	2	2	2

(c)

Fig 3.7 (a) Magnified part of brain image (yellow square); (b) represents corresponding intensity values in range [0 255]; (c) intensity values are scaled into range [1 8].

Considering a GLCM matrix C , the value of $C(m\,,n)$ indicates how many times the gray tone m co-occurs with gray tone n in some designated spatial relationship [97]. The spatial relationship is usually defined by a distance d and a direction θ. Fig. 3.8 shows different possible values for θ in an image matrix.

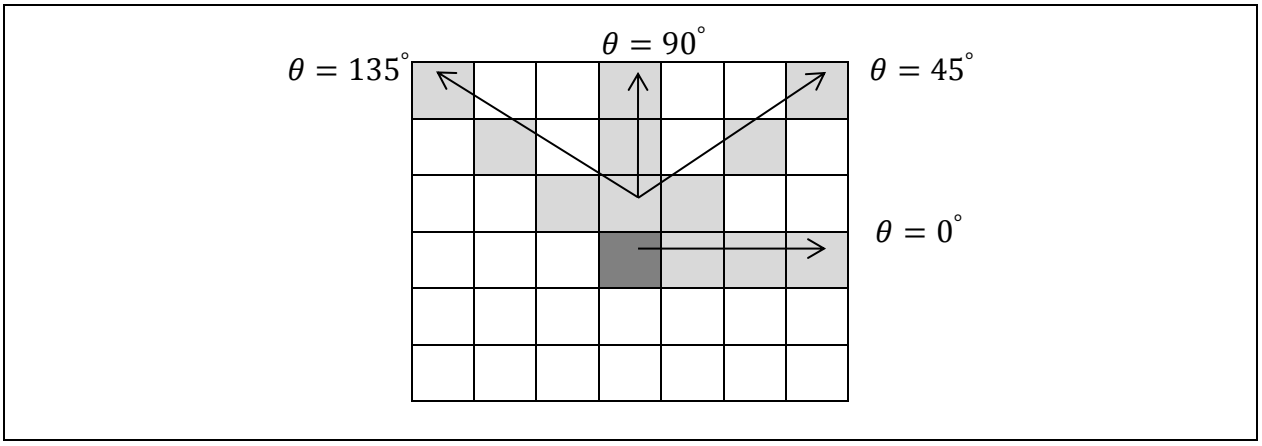


Fig. 3.8 Visualizing different values of θ for constructing GLCM

One problem with deriving texture measures from co-occurrence matrices is how to choose d and θ. A solution suggested in [97] is to use a X^2 statistical test to select the value(s) of d and θ that have the most structure; that is, to maximize eq. (3.20).

$$X^2(d,\theta) = \sum_i \sum_j \frac{N_{d,\theta}^2(i,j)}{N_{d,\theta}(i) N_{d,\theta}(j)} - 1 \tag{3.20}$$

Where N is a normalized version of co-occurrence matrix C. However authors in [80, 102-108] used $d = 1$ (for the purpose of fine texture analysis) and $\theta = 0{,}45{,}90$ and 135. Fig. 3.9 shows the calculated GLCM matrix for the image matrix shown in Fig. 3.7-c in different directions.

Having calculated GLCM matrix C, one can calculate features that are described in eq. (3.21)-(3.46). These features are used and discussed in [7, 65, 81, 99, 107, 109-113].

$$\text{Energy or Angular second moment} = \sum_{i,j} C(i,j)^2 \tag{3.21}$$

The energy of a texture describes the uniformity of the texture. In a homogeneous image, the co-occurrence matrix has fewer entries of large magnitude and the energy of the image is high when the image is homogeneous [65]. Energy is 1 for a constant image [112].

Gray tones	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	65	10	0	0	0	0	0
3	0	10	25	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

$\theta = 0^0, d = 1$

Gray tones	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	56	12	0	0	0	0	0
3	0	9	23	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

$\theta = 45^0, d = 1$

Gray tones	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	70	8	0	0	0	0	0
3	0	5	27	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

$\theta = 90^0, d = 1$

Gray tones	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	56	12	0	0	0	0	0
3	0	9	23	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

$\theta = 135^0, d = 1$

Fig. 3.9 GLCM calculation in different directions for image matrix shown in Fig. 3.7-c

$$\text{Entropy } = -\sum_{i,j} C(i,j) log C(i,j) \tag{3.22}$$

The entropy measures the randomness of the element. For a homogeneous image, the entropy should be low [65]. In other words, the entropy measures the disorder or complexity of an image. Complex textures tend to have high entropy. Entropy is strongly, but inversely correlated to energy [112].

$$\text{Inverse difference moment } = \sum_{i,j} \frac{1}{1+(i-j)^2} C(i,j) \tag{3.23}$$

The Inverse Difference Moment (IDM) is influenced by the homogeneity of the image. Because of the weighting factor $(1+(i-j)^2)^{-1}$, IDM will get small contributions from inhomogeneous areas (i,j). The result is a low IDM value for inhomogeneous images and higher value for homogeneous images [112]. IDM is large when the diagonal of C is large [65].

$$Inertia = \sum_{i,j}(i-j)^2 C(i,j) \tag{3.24}$$

The inertia is large when the non-diagonal values of C are large. The inertia and the inverse difference moment indicate the distribution of gray-scales in the image [65]. Its name in [107] is changed to Contrast but the formula is the same as Inertia. Contrast is a measure of amount of the local variation in the image. A higher contrast value indicates a high amount of local variation, so the higher the contrast the clearer the image is [107].

$$\text{Shade} = \sum_{i,j}\left(i+j-\mu_x-\mu_y\right)^3 C(i,j) \tag{3.25}$$

$$\text{Prominence} = \sum_{i,j}\left(i+j-\mu_x-\mu_y\right)^4 C(i,j) \tag{3.26}$$

Where μ_x and μ_y are calculated as stated in eq. (3.27) and (3.28), respectively.

$$\mu_x = \sum_{i,j} i.C(i,j) \tag{3.27}$$

$$\mu_y = \sum_{i,j} j.C(i,j) \tag{3.28}$$

Shade and prominence are measured by the skewness of C. The image is not symmetric when shade and prominence are high [65].

$$\text{Correlation} = \sum_{i,j} \frac{(i-\mu_x)+(j-\mu_y)}{\sqrt{\sigma_x \sigma_y}} C(i,j) \tag{3.29}$$

Where σ_x and σ_y are calculated as stated in eq. (3.30) and (3.31), respectively.

$$\sigma_x = \sum_{i,j}(i-\mu_x)^2.C(i,j) \tag{3.30}$$

$$\sigma_y = \sum_{i,j}\left(j-\mu_y\right)^2.C(i,j) \tag{3.31}$$

Correlation is a measure of the gray level linear dependency between the pixels at the specified positions relative to each other. Correlation will be high if an image contains a considerable amount of linear structures [112].

$$\text{Variance or Sum of Squares } = \sum_{i,j}(i-\mu)^2 C(i,j) \tag{3.32}$$

Where μ is the mean value of matrix C. The variance is a measure of the dispersion of the gray level differences at a certain distance, d. This feature puts relatively high weights on the elements that differ from the average value of $C(i,j)$ [112].

$$\text{Difference entropy} = -\sum_{i=0}^{G-1} C_{x-y}(i) \log\left(C_{x-y}(i)\right) \tag{3.33}$$

$$C_{x-y}(k) = \sum_{i=1}^{G}\sum_{j=1}^{G} C(i,j) \quad i-j=k\,, k = 0,1,\dots,G-1$$

Where G is the number of gray levels that exists in GLCM matrix; $C_x(i)$ is the ith entry in the marginal-probability matrix obtained by summing the rows of $C(i,j)$ and $C_y(i)$ is obtained by summing the columns of $C(i,j)$. Difference entropy is a measure of histogram content and logical value between two images. If two images are identical the difference entropy will be high otherwise it is low [112].

$$\text{Homogeneity} = \sum_{\text{i,j}} \frac{\text{C(i,j)}}{1+|\text{i}-\text{j}|} \tag{3.34}$$

Homogeneity returns a value that measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal. Homogeneity is 1 for a diagonal GLCM. A homogeneous image will result in a co-occurrence matrix with a combination of high and low $C(i,j)$s. A heterogeneous image will result in an even spread of $C(i,j)$s [112].

$$\text{Dissimilarity} = \sum_{i,j}|i-j|.\,C(i,j) \tag{3.35}$$

Dissimilarity is a measure of evenness between two groups [112].

$$\text{Sum average} = \sum_{i=2}^{2G} i C_{x+y}(i) \tag{3.36}$$

$$C_{x+y}(k) = \sum_{i=1}^{G}\sum_{j=1}^{G} C(i,j) \quad i+j=k\,, k = 2,3,\dots,2G$$

$$\text{Sum Entropy} = -\sum_{i=2}^{2G} C_{x+y}(i)\log(C_{x+y}(i)) \tag{3.37}$$

$$C_{x+y}(k) = \sum_{i=1}^{G}\sum_{j=1}^{G} C(i,j) \quad i+j=k\,, k = 2,3,\dots,2G$$

$$\text{Sum variance} = \sum_{i=2}^{2G}(i - \text{Sum Entropy})^2 C_{x+y}(i) \tag{3.38}$$

$$C_{x+y}(k) = \sum_{i=1}^{G}\sum_{j=1}^{G} C(i,j) \quad i+j=k\,, k = 2,3,\dots,2G$$

$$\text{Difference variance} = variance\ of\ C_{x-y} \tag{3.39}$$

$$C_{x-y}(k) = \sum_{i=1}^{G}\sum_{j=1}^{G} C(i,j) \quad i-j=k\,, k = 0,1,\dots,G-1$$

Difference in variance is the sum of difference between intensity of the central pixel and its neighborhood [112].

$$\text{Information measure of correlation1} = \frac{=-\sum_{i,j} C(i,j)\log(C(i,j)) - HXY1}{\max(HX\,, HY)} \tag{3.40}$$

Where HX and HY are Entropies of $C_x\ and\ C_y$ and $HXY1 = -\sum_{i,j} C(i,j) log\{C_x(i)C_y(j)\}$.

$$\text{Information measure of correlation2} = (1 - exp[-2.0(HXY2 - HXY)])^{1/2} \tag{3.41}$$

Where $HXY = Entropy\ \ , HXY2 = -\sum_i \sum_j C_x(i)C_y(j) log\{C_x(i)C_y(j)\}$.

$$\text{Maximal Correlation Coefficient} = (Second\ largest\ eigenvalue\ of\ Q)^{1/2} \quad (3.42)$$

Where $Q(i,j) = \sum_k \frac{C(i,k)C(j,k)}{C_x(i)C_y(k)}$.

Equations (3.40)-(3.42) are based on the mutual information concept. Mutual information measures the decrease of uncertainty about variable X caused by the knowledge of variable Y, and vice versa. In our problem, having two pixels that already hold the spatial relationship r, mutual information measures the decrease of uncertainty about the intensity value of the first pixel when we have the intensity value of the second one. Recall that in GLCM matrix C the value of $C(m,n)$ indicates how many times the gray tone m co-occurs with gray tone n in some designated spatial relationship r (i.e., usually defined by a distance d and a direction θ).

It is mentioned in [109] that the measures of correlation stated in (3.40)-(3.42) have some desirable properties which are not brought out in correlation measure presented in (3.29). Based on [114], (3.40)-(3.42) are able to measure dependency when there exists a nonlinear structure between the random variables, while the correlation coefficient only measures linear dependency between random variables.

$$\text{Inverse difference moment normalized} = \sum_{i,j=1}^{G} \frac{C(i,j)}{1+(i-j)^2/G^2} \quad (3.43)$$

$$\text{Inverse difference normalized} = \sum_{i,j=1}^{G} \frac{C(i,j)}{1+|i-j|/G} \quad (3.44)$$

In equations (3.43) and (3.44), the weighting factors $(1+(i-j)^2/G^2)^{-1}$ and $(1+|i-j|/G)^{-1}$ will be reduced when the difference between i and j increases. This usually occurs in inhomogeneous areas where there is a tangible difference between the intensity values of two pixels holding designated spatial relationship r. The result is a low Inverse difference moment normalized and Inverse difference normalized values for inhomogeneous images and higher values for homogeneous images.

$$GLCM\ autocorrelation = \sum_{i,j} i \times j \times C(i,j) \quad (3.45)$$

The GLCM autocorrelation provides a measure of gray-tone linear-dependencies [115] between each pixel and its immediate neighbors.

$$Maximum\ probability = \max_{i,j}(C(i,j)) \tag{3.46}$$

The maximum probability extracts the probability value of the most frequent difference between gray levels of adjacent pixel pairs within image. It is expected to be high if the occurrence of the most predominant pixel pairs is high. As stated in [116], the maximum probability plays a role similar to uniformity; the high values of this feature are usually associated with homogenous regions and the lower values with heterogeneous regions.

Another group of features are symmetry features. An analysis of symmetry features conducted in [4] concludes that the use of this type of features improves the accuracy of classifiers designed for automatic detection of CVA. Given the ideal mid-sagittal line, the proposed symmetry features aim on comparing one side of the brain to the other side and discover if there are any suspicious differences (i.e., the existence of Cerebral Vascular Accident can affect the symmetrical property of human brain, observable on Computer Tomography images).

To extract symmetry features, a window $w_1$of size $s \times s$ centred at the pixel (x, y) marked by a clinical expert as normal or abnormal and its contralateral part with respect to the midline, window w_2 centred at the pixel (x', y'), are considered (Please see Fig. 3.10). Having identified w_1 and w_2, we can then specify how similar these two regions are by calculating Pearson Correlation Coefficient (PCC) as stated in eq. (3.47). L_1 norm and squared L_2 norm are also two dissimilarity measures that can be calculated through eq. (3.48) and eq. (3.49) respectively. Comparing the intensity value of the pixel that is marked by the expert and its corresponding pixel in the contralateral part can give us another symmetry feature that is stated in eq. (3.50).

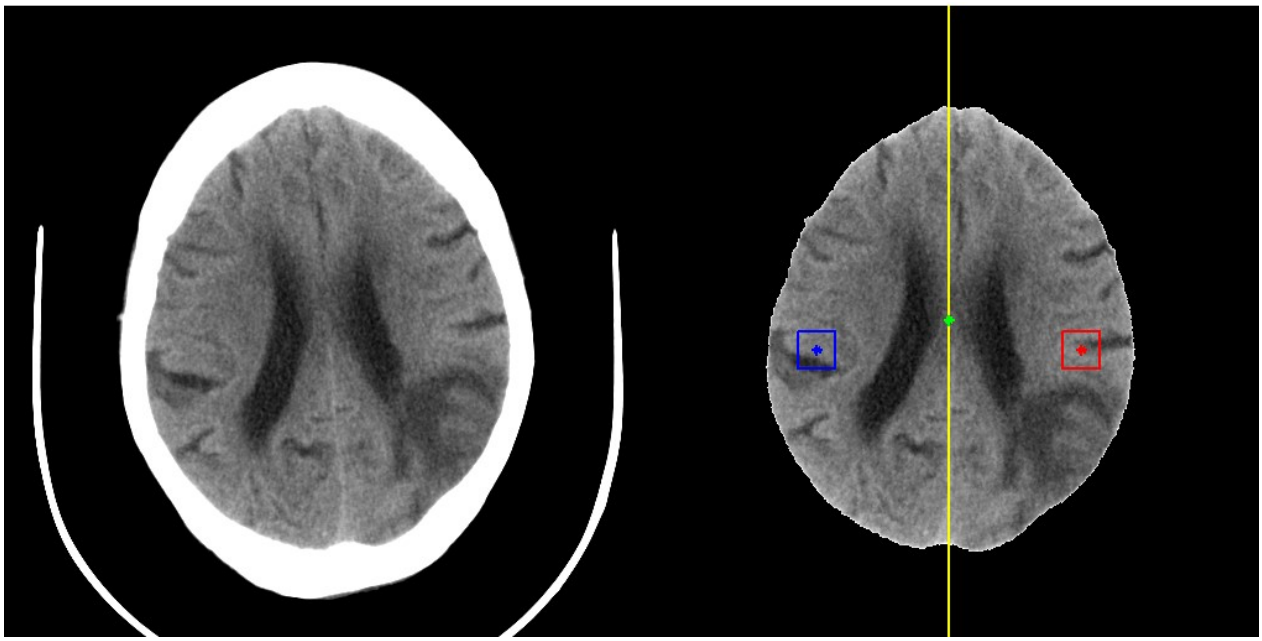
$$PCC = \frac{1}{n}\sum_{i=1}^{s}\sum_{j=1}^{s}\left(\frac{I_{w_1}^{i,j}-\mu_{w_1}}{\sigma_{w_1}}\right)\left(\frac{I_{w_2}^{i,j}-\mu_{w_2}}{\sigma_{w_2}}\right) \tag{3.47}$$

$$L_1 = \sum_{i=1}^{s}\sum_{j=1}^{s}|I_{w_1}^{i,j} - I_{w_2}^{i,j}| \tag{3.48}$$

$$L_2^2 = \sum_{i=1}^{s}\sum_{j=1}^{s}\left(I_{w_1}^{i,j} - I_{w_2}^{i,j}\right)^2 \tag{3.49}$$

$$diff = I_{w_1}^{x,y} - I_{w_2}^{x',y'} \tag{3.50}$$

Where $I^{(i,j)}$ is the intensity value of pixel located at (i,j) within the corresponding window; μ_{w_1}, μ_{w_2}, σ_{w_1} and σ_{w_2} are the mean and standard deviation of intensity values within window $w_1$and its contralateral part, window w_2 , respectively.



(a) (b)

Fig. 3.10: (a) Original brain CT image; (b) After skull removal and realignment, ideal midline is drawn in yellow color. The green point shows the mass centre (centroid) of skull. A window of size 31x31 is considered around pixel located at (365,279) and shown in red color; its contralateral part with respect to the midline is shown in blue color.

4. Data acquisition and registering tool

To train, test and validate the neural network models for classifying pathologic areas within brain CT images, we need to have a gold standard; In our case the gold standard was the clinical information provided by the Neuroradiologists. In order to collect this information in an accurate but also user-friendly digitally stored, a web-based tool was developed [3]. Using this tool, a database of CT images was created for Neuroradiologists to analyze and mark the images either as normal or abnormal. For the abnormal ones, the Neuroradiologists were guided to identify the type and degree of alteration, together with the identification of the region they considered abnormal on each CT's slice image with abnormal aspect. This chapter describes the software tool developed for data acquision and registration, being organized as follows: Section 4.2 describes different functionalities of the tool provided for different types of users and section 4.3 explains implementation details.

4.1 Software functionalities

As it can be seen from the use case diagram, shown in Fig. 4.1, two kinds of users can work with this software: the Administrator and Neuroradialogists. The software provides specific functionalities for each role which are explained in the following subsections.

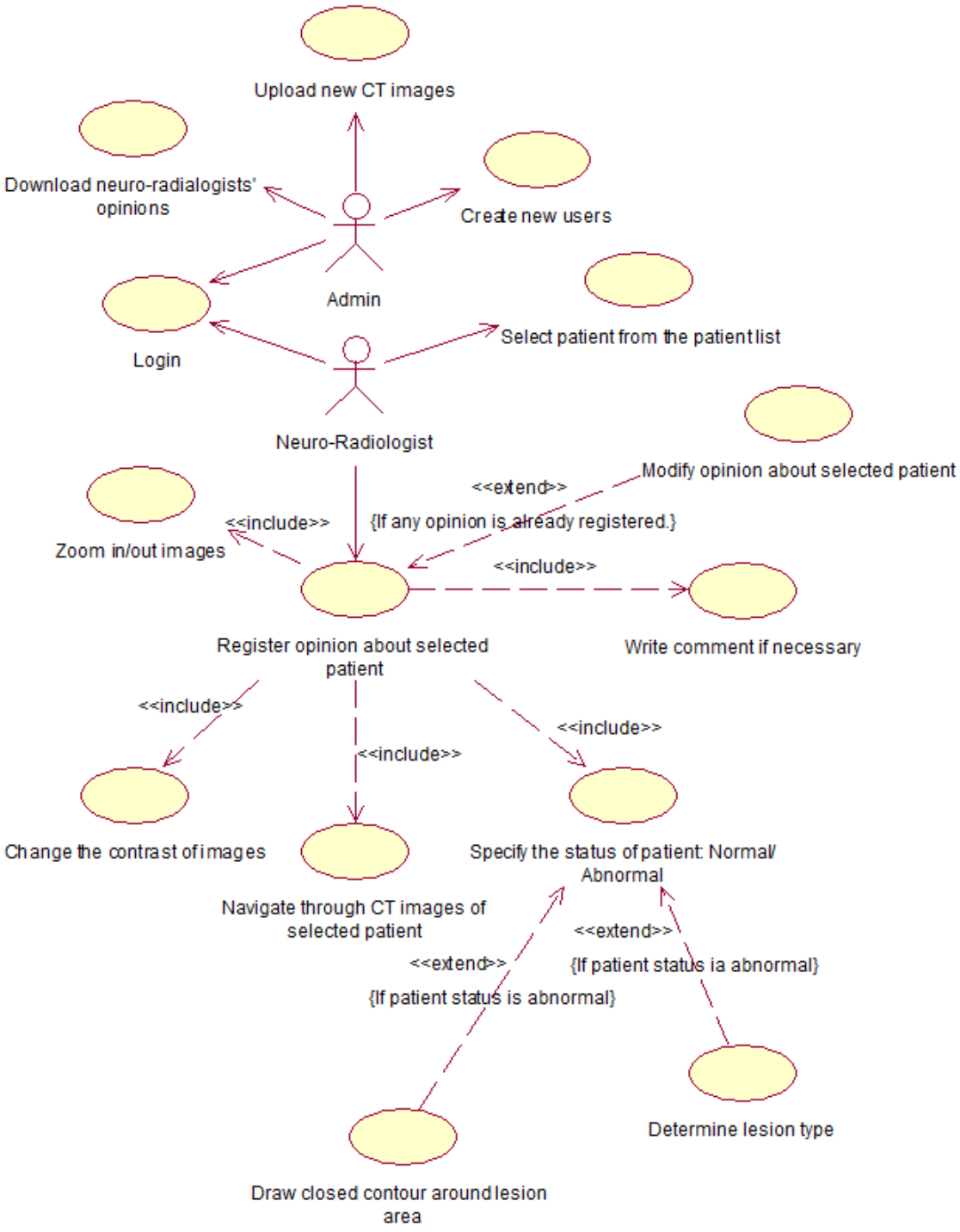


Fig. 4.1 General diagram of the Web-based tool for registering and identification of pathological areas in CT images

4.1.1 Administrator facilities

The Administrator has the possibility to upload brain CT images into the system, define new users and download Neuroradiologists' opinion in a format that can be used for the feature extraction process. Fig. 4.2 shows the software activity diagram of the user Administrator.

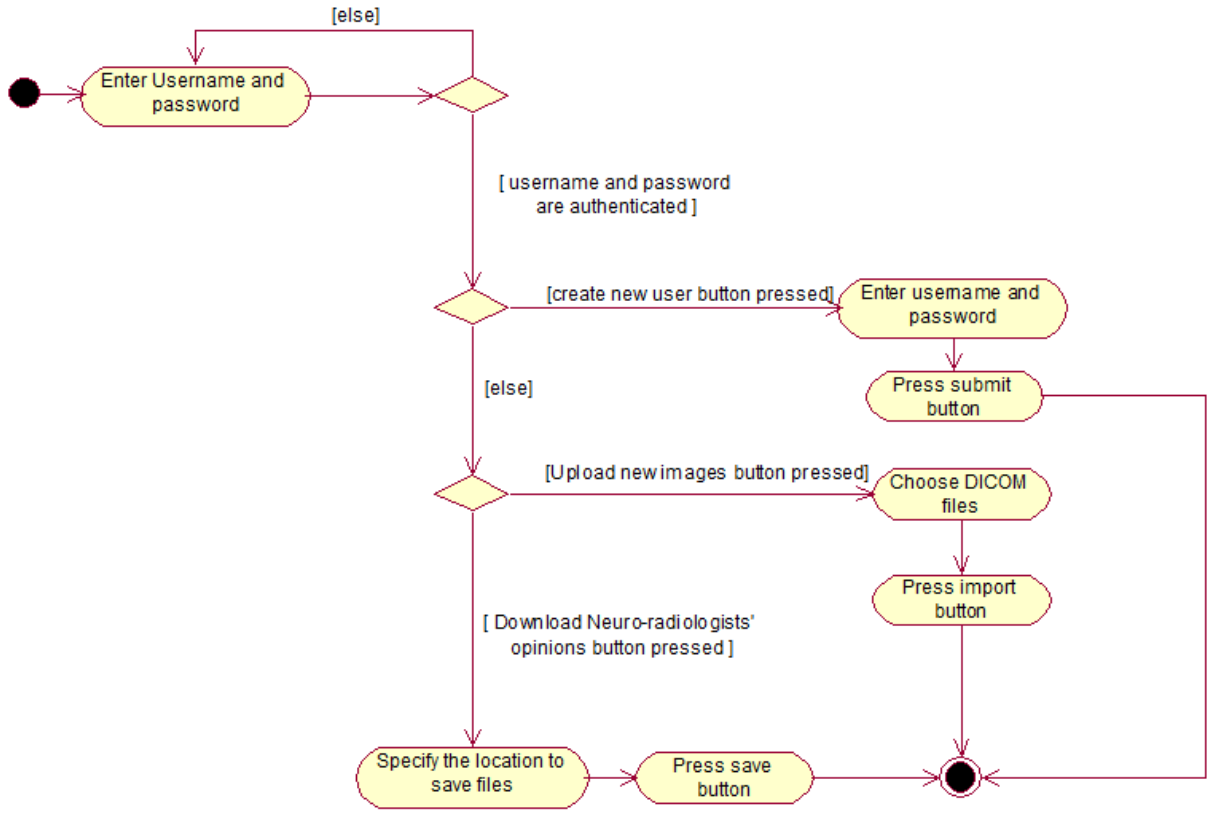


Fig. 4.2 Diagram of possible activities of the user Administrator

4.1.1.1 Uploading CT images

The Administrator can upload a set of DICOM files into the system which will be afterwards visible to the currently defined users. In order to make the series of CT images visible to the Neuroradiologists, with the proper quality, through our proposed tool, the binary data of each CT image from the DICOM files is extracted and the appropriate window level and window width applied. Window width describes the range of Hounsfield Units (HU) displayed in a CT image where HU measures the absorbed amount of x-radiation in CT scans. This means that, having a CT image, the tissues whose HU are above the range of window width are seen as white and the ones

whose HU are below the range are seen as black. Window level is the Hounsfield number in the center of the window width. In many DICOM viewer softwares like the DicomWorks [117], the window level and window width values for brain tissues are set to 40 and 80 respectively. The same setting for converting the DICOM files into PNG is used and PNG files are saved into the database. To group exams of the same patient, as well as displaying some useful information such as: patient age, patient gender, study description, etc. to the Neuroradiologists through the interface, useful metadata of each CT image are extracted from the corresponding DICOM file. Fig. 4.3 shows the interface for uploading new CT images.

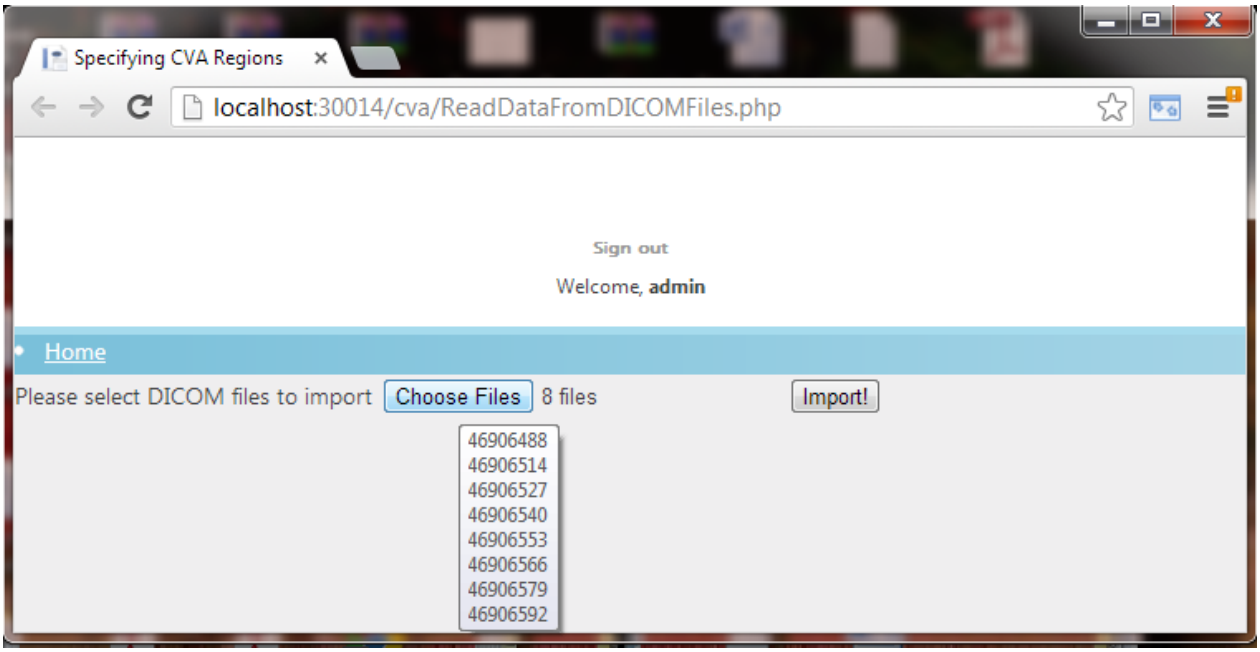
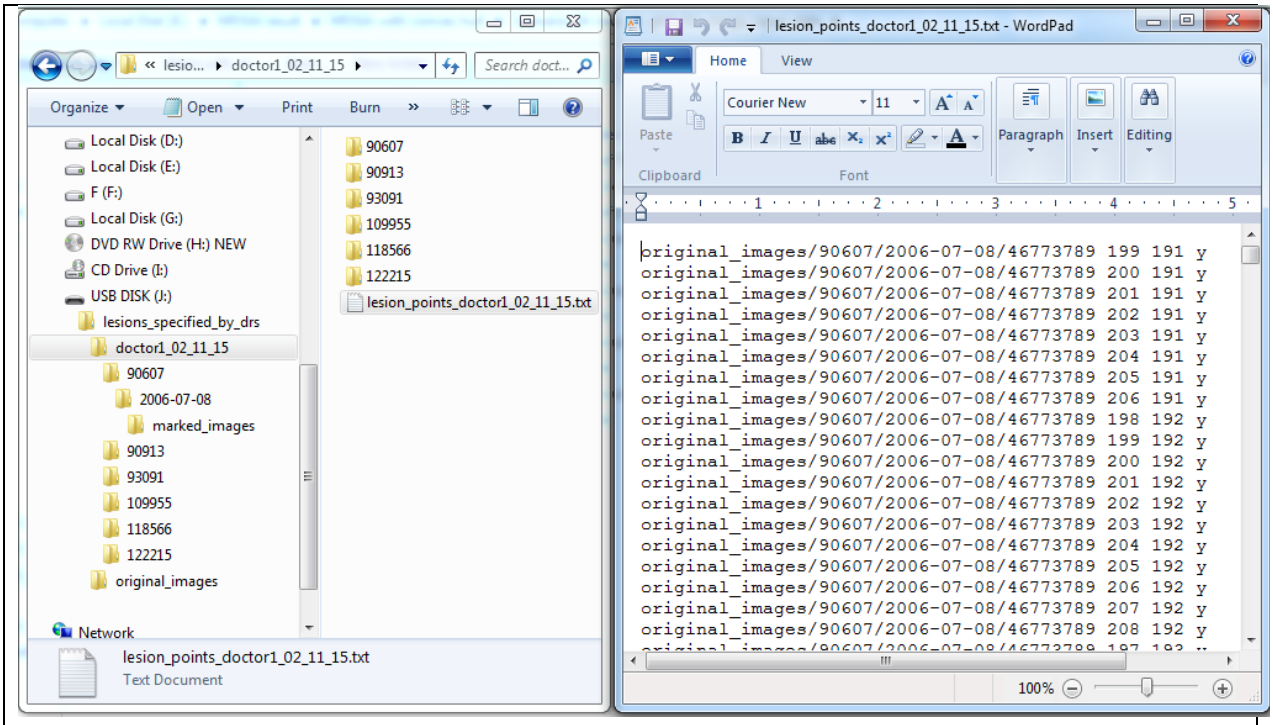


Fig. 4.3 Administrator interface for uploading new CT images.

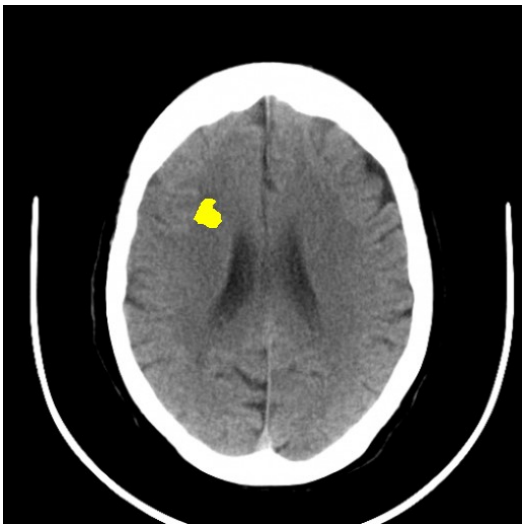
4.1.1.2 Downloading Clinical reports

By pressing this option, the Administrator will receive a zip file containing multiple folders, each one corresponding to a Neuradiologist already defined in the system. Within each folder, say “A”, one can find all CT images on which Neuroradiologist “A” has already marked some lesions; these CT images are organized in different folders based on the patient identity and the exam date. Beside these images, there is also a text file which specifies the “x” and “y” coordinates of all pixels that are marked as lesion by Neuroradiologist “A”, as well as their corresponding lesion type and the

path from which the image can be retrieved. Fig. 4.4 shows the structure of the zip file containing the Neuradiologists' opinions as well as the structure of the text file.



(a)



(b)

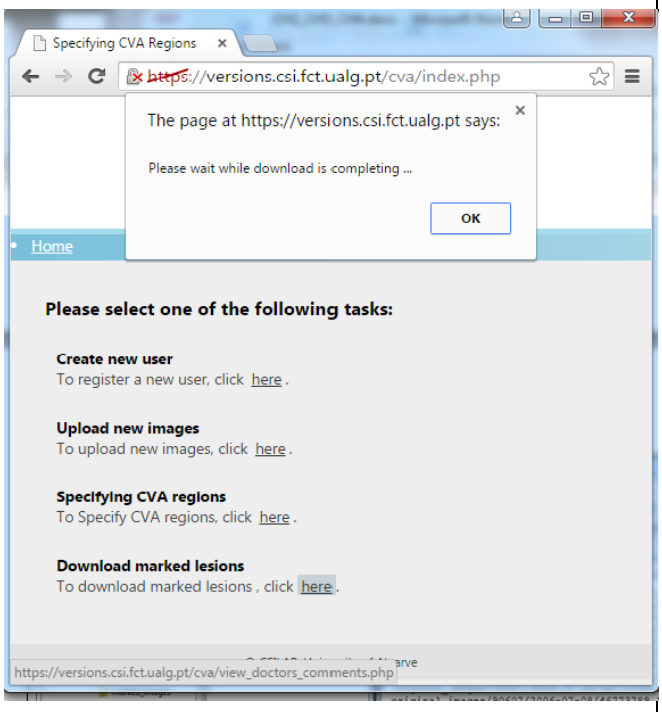
Fig. 4.4 (a) left side shows the structure of zip file containing Neuradiologists' opinions; right side shows the content of text file which determines the coordinate and type of lesion pixels including the ones that are already marked in (b).

As it can be seen, the first column in the text file is the path to the image, the second and third column specify the “x” and “y” coordinates of the marked pixel and the last column indicates the color by which the pixel is marked, as different types of lesions are coded by colors (i.e., “Y” stands for yellow).

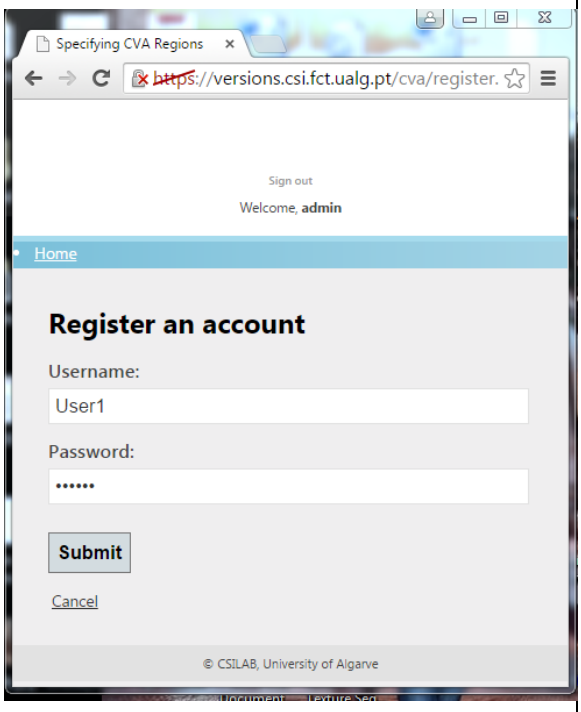
Fig. 4.5-(a) shows the user interface of the administrator. The last link provides the administrator to download the Neuroradiologists’ opinion.

4.1.1.3 Defining new users

The Administrator can create new users through defining usernames and passwords. Each Neuroradiologist will have access to the system by providing the given username and password. Fig. 4.5-(b) shows the user interface for creating new users.



(a)



(b)

Fig. 4.5 (a) Administrator interface where the last link is for downloading Neuroradiologists’ opinions; (b) Administrator interface for creating new users.

4.1.2 Users' facilities

After logging into the system, a list of CT exams is provided for the already registered Neuroradiologist to insert his clinical evaluation. Each CT exam has been associated with the patient identity (id) and the exam date. Looking into this list, the doctor can quickly find out which exams are already evaluated. The Normal/Abnormal status of already visited exams is also indicated (Please see Fig. 4.6). By clicking on the patient id of each CT exam, the Neuroradiologist will be redirected to a new page, shown in Fig. 4.7, where he/she can inspect all corresponding CT slices and register his/her evaluation.

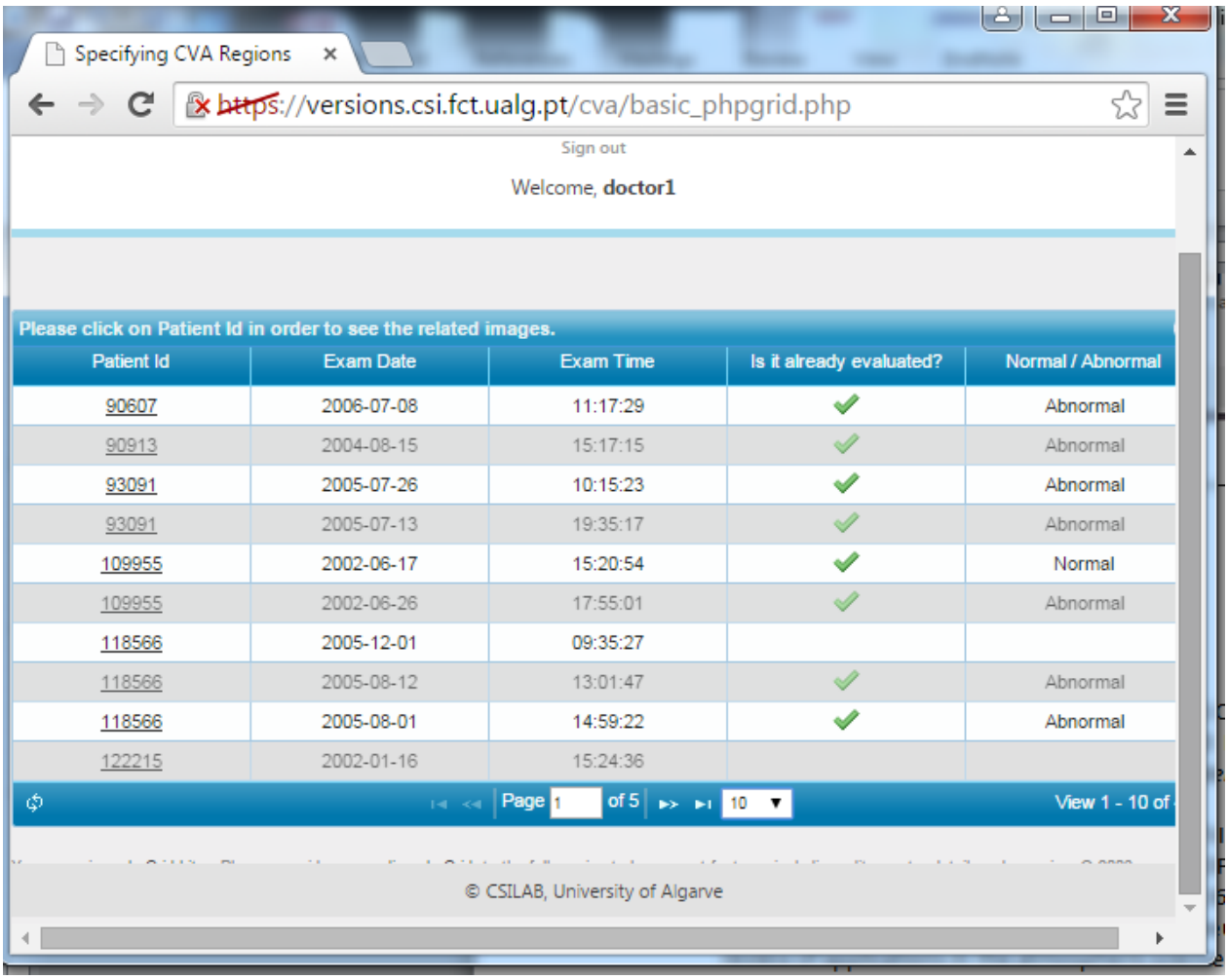


Fig. 4.6 User interface for displaying list of CT exams to Neuroradiologist

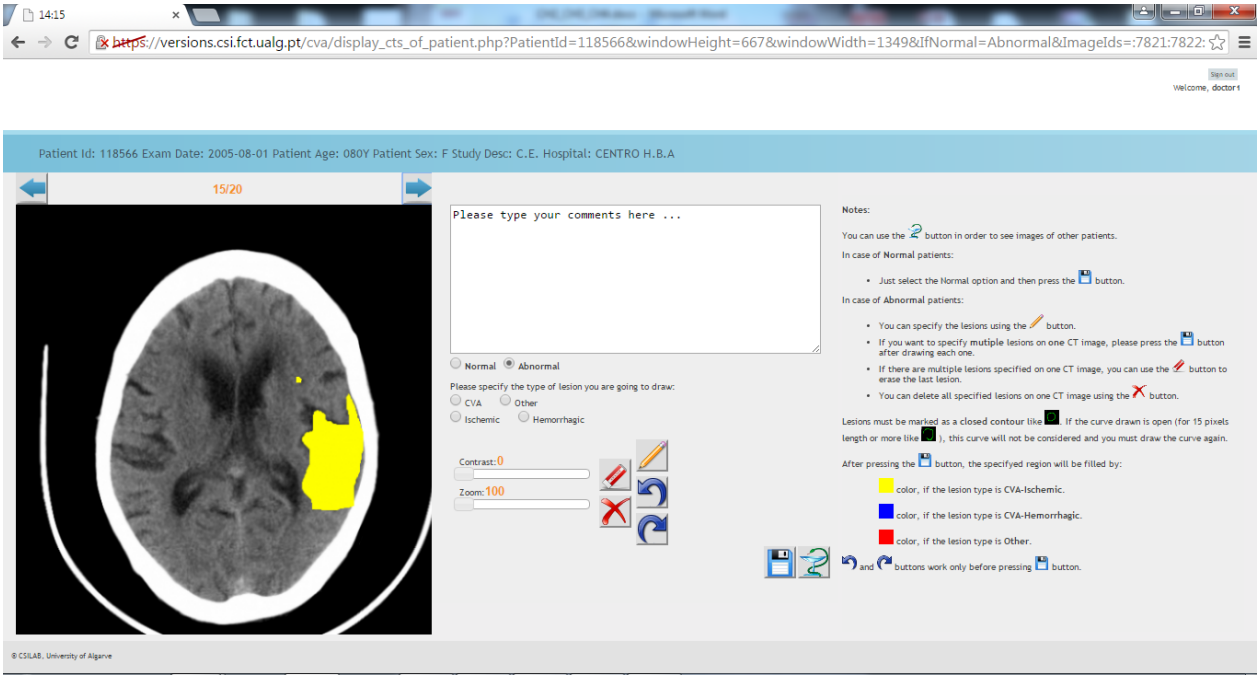


Fig. 4.7 User Interface for registering pathological areas

Through this page the Neuroradiologist can specify whether the patient is normal or abnormal. In the CT scan which is marked as abnormal, the expert can specify three different types of brain lesions, by drawing a contour around the lesion area. If the drawn contour is open and the discontinuity is less than a predefined threshold, say α, the tool will automatically close the contour; otherwise, the doctor will be asked to specify the lesion area more precisely. Afterwards, the interior of the closed contour will be filled by a different color corresponding to the lesion type and saved as the lesion area. Different colors are depicted in Table 4.1. The contrast and zoom level of each CT image can be changed, which helps the Neuroradiologists to inspect the images more precisely. The user also has the option of removing either the last specified lesion or all drawn lesions. In order to allow the Neuroradiologist to specify multiple lesions in one image, a layer strategy was applied. This means that each lesion will be saved as a transparent layer which contains only the current closed contour. For displaying the complete diagnosed areas, all transparent layers are superimposed to the original image. Figure 4.8 shows the different process stages that are applied into one sample image. The activity diagram for Neuroradiologist can be seen in Fig. 4.9.

Table 4.1 Different types of brain lesions are marked by different colours in each CT image

Lesion Type	Dedicated Color
CVA- Ischemic	Yellow
CVA-Hemorrhagic	Blue
Other types of lesions	Red

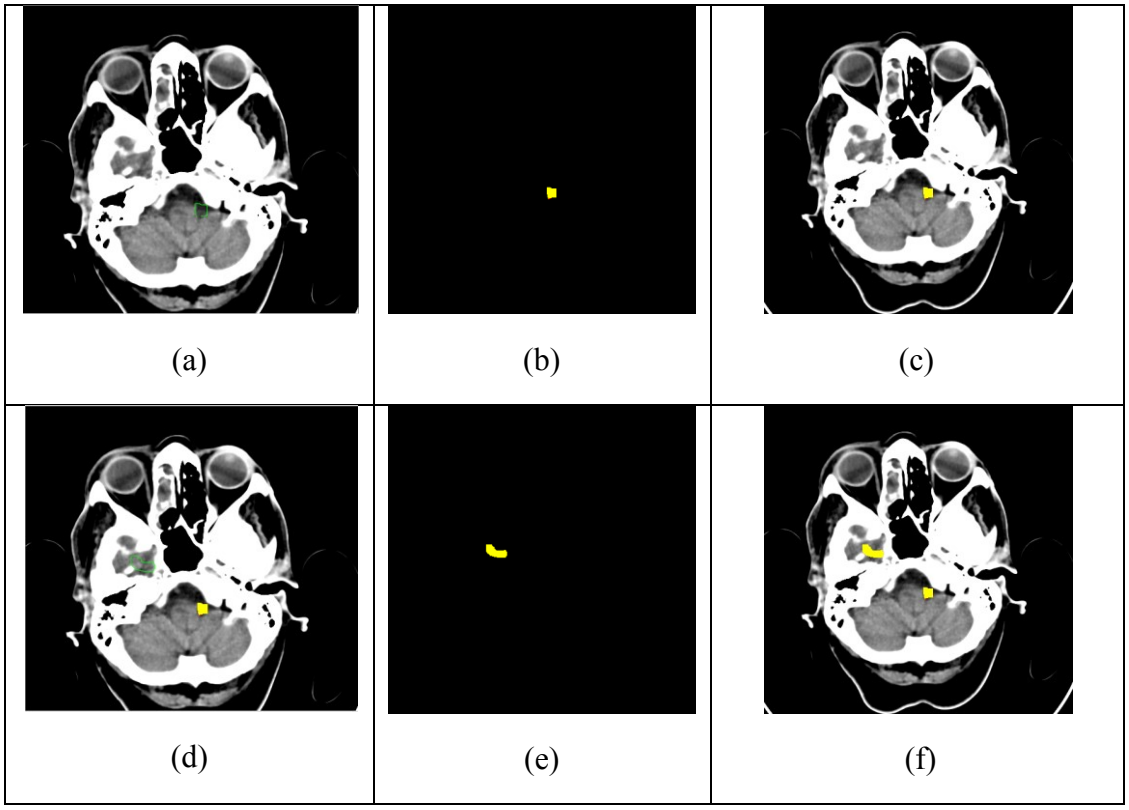


Fig. 4.8(a), (d) Green pixels shows the drawn contours; the first one is open and the second one is closed. (b), (e) the transparent layers of the processed contours. (c), (f) the transparent layers are super imposed on the original image.

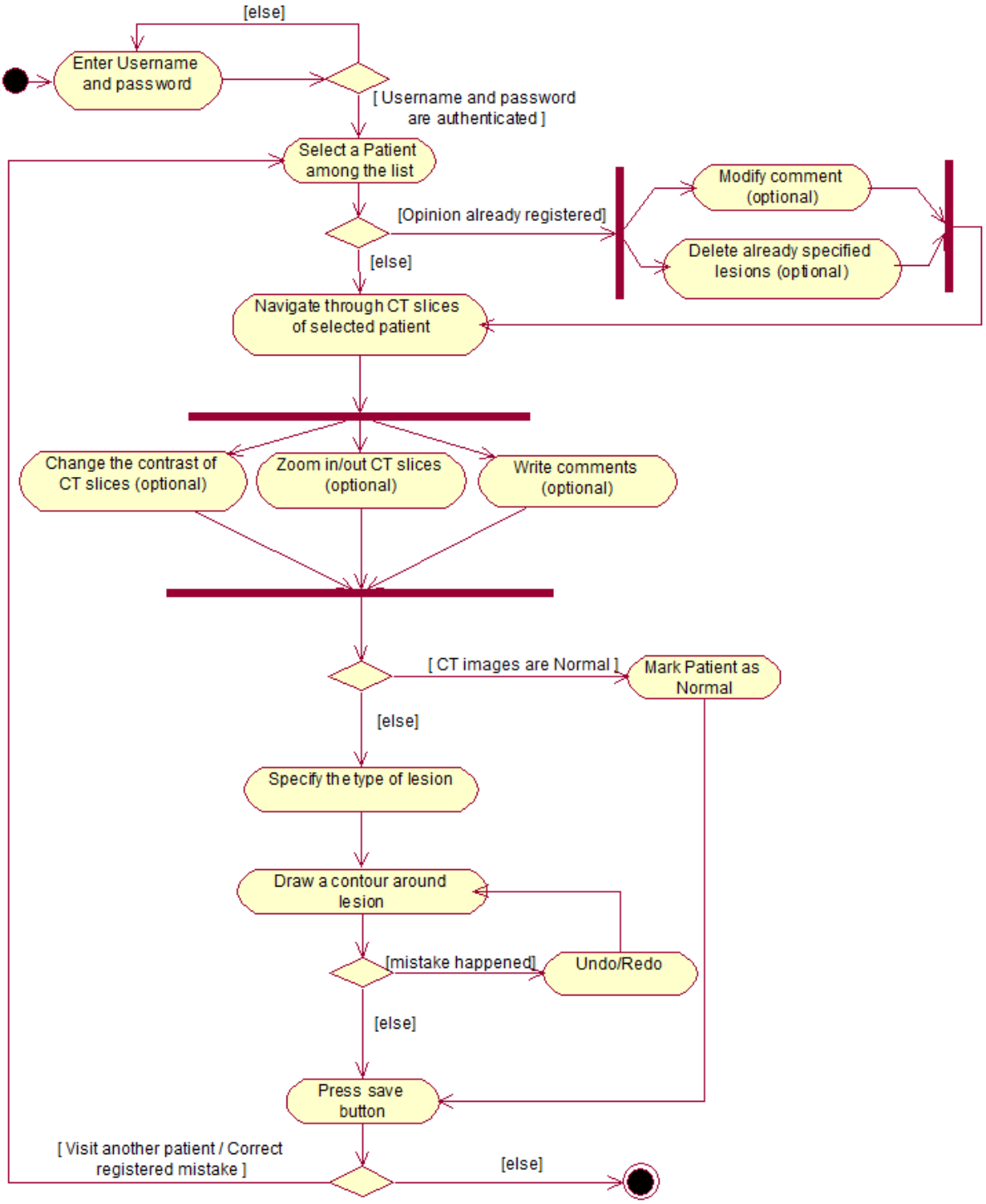


Fig. 4.9 Diagram of possible activities of other users, namely Neuroradiologists

4.2 Implementation details

The software has been implemented using PHP and MySQL. The database diagram of the software can be seen in Fig. 4.10. Table “origpngimage” contains all necessary data that is extracted from DICOM images including the exam date and time, patient age and sex, CT slice thickness and the space between the slices, as well as the binary image. Whenever a doctor marks a lesion area on a CT image, a transparent image which contains only the marked area is saved in the “contourpngimage” table. As mentioned before, this strategy was applied to allow the doctor to specify multiple lesions on one image. Moreover, the doctor is able to delete any previously marked lesion, keeping the other ones intact. All transparent layers are then superimposed to the original image and the final result is saved into “modifiedpngimage” table. Table “ifpatientisvisitedbydr” holds information about if a patient has/has not been already visited by each doctor. It also specifies each doctor’s opinion about each patient (i.e., whether it is Normal or Abnormal). Credentials of users are kept in “users” table. The role of each user is identified in “users_in_roles”. The available roles Administrator and Neuroradiologist are defined in the “roles” table.

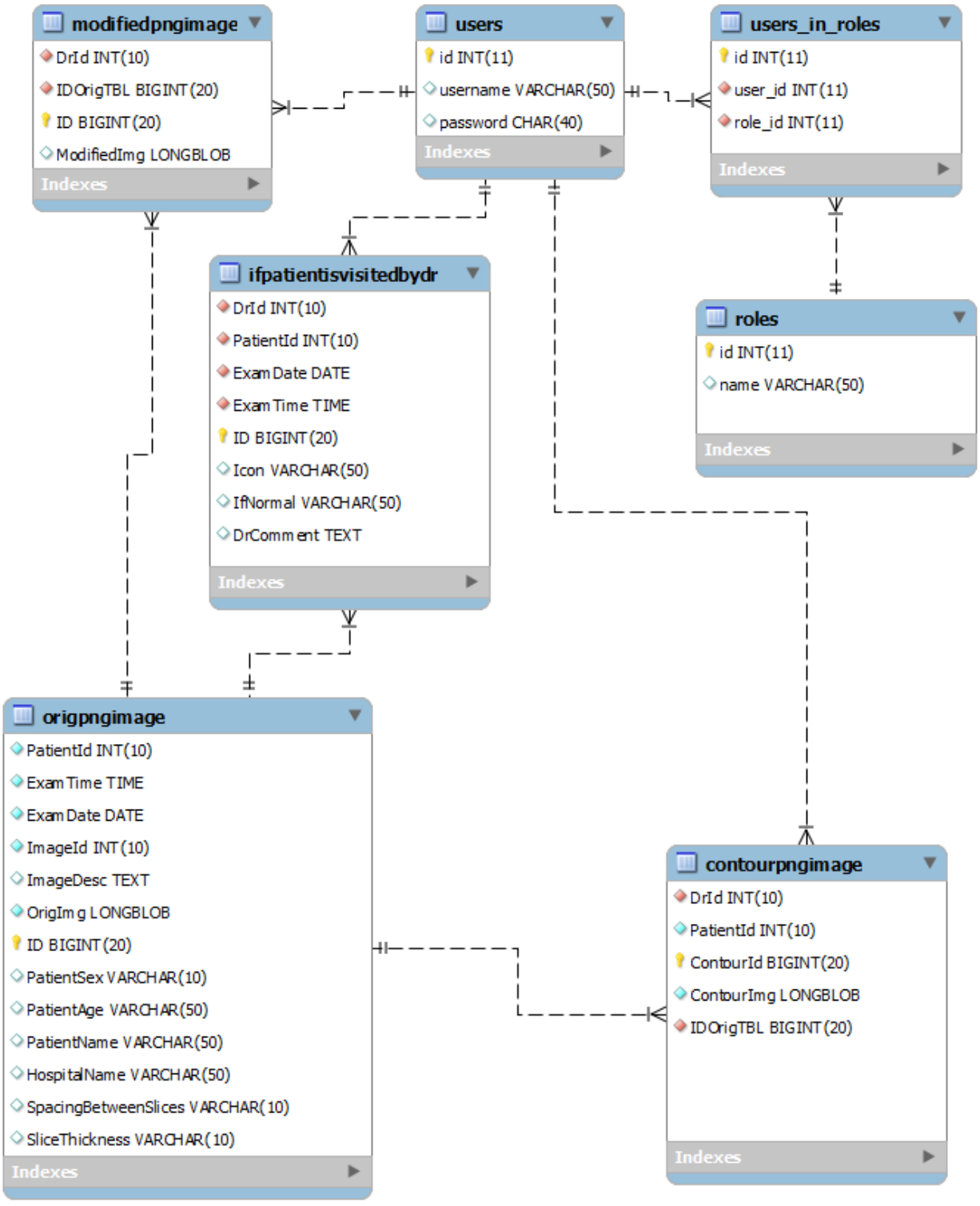


Fig. 4.10 Global diagram of the database of the web-based tool

In order to obtain an exact localization of pixels inside the lesion area, two modules were incorporated in the tool core, named Skeletonization and Moore Tracing modules. These allow detecting if the contour drawn by the Neuroradiologist has missing points to allow a closed contour or not. Since, on one hand, Moore Neighbor Tracing algorithm [118] detects a line whose width is more than one pixel as a closed contour and, on the other hand, depending on how the doctor is using the pencil tool, the width of the drawn contour can be larger than one pixel, the K3M algorithm [119] was used to force the contour to be one pixel width.

K3M consists of an iterative part of six phases and a single additional phase at the end, which is responsible for producing a one pixel width skeleton. The overview of the algorithm is presented in Fig. 4.11. Fig. 4.12 presents the flowchart of Phase 0, aimed at marking borders (pixels that are candidates for deletion). The iterative phases 1 to 5 together with "one pixel width skeleton" phase are very similar to each other. Hence they are presented using a common flowchart with parameter i that changes from 1 to 6 (please see Fig. 4.13). The iterative phases 1 to 5 aim at deleting pixels with a growing number of sticking neighbors. Each phase i uses a lookup array A_i which is presented in Table 4.2.

Table 4.2 Components of neighbourhood lookup arrays in K3M algorithm

A_0 = {3, 6, 7, 12, 14, 15, 24, 28, 30, 31, 48, 56, 60, 62, 63, 96, 112, 120, 124, 126, 127, 129, 131, 135, 143, 159, 191, 192, 193, 195, 199, 207, 223, 224, 225, 227, 231, 239, 240, 241, 243, 247, 248, 249, 251, 252, 253, 254}
A_1 = {7, 14, 28, 56, 112, 131, 193, 224}
A_2 = {7, 14, 15, 28, 30, 56, 60, 112, 120, 131, 135, 193, 195, 224, 225, 240}
A_3= {7, 14, 15, 28, 30, 31, 56, 60, 62, 112, 120, 124, 131, 135, 143, 193, 195, 199, 224, 225, 227, 240, 241, 248}
A_4= {7, 14, 15, 28, 30, 31, 56, 60, 62, 63, 112, 120, 124, 126, 131, 135, 143, 159, 193, 195, 199, 207, 224, 225, 227, 231, 240, 241, 243, 248, 249, 252},
A_5= {7, 14, 15, 28, 30, 31, 56, 60, 62, 63, 112, 120, 124, 126, 131, 135, 143, 159, 191, 193, 195, 199, 207, 224, 225, 227, 231, 239, 240, 241, 243, 248, 249, 251, 252, 254},

A_6= {3, 6, 7, 12, 14, 15, 24, 28, 30, 31, 48, 56, 60, 62, 63, 96, 112, 120, 124, 126, 127, 129, 131, 135, 143, 159, 191, 192, 193, 195, 199, 207, 223, 224, 225, 227, 231, 239, 240, 241, 243, 247, 248, 249, 251, 252, 253, 254}

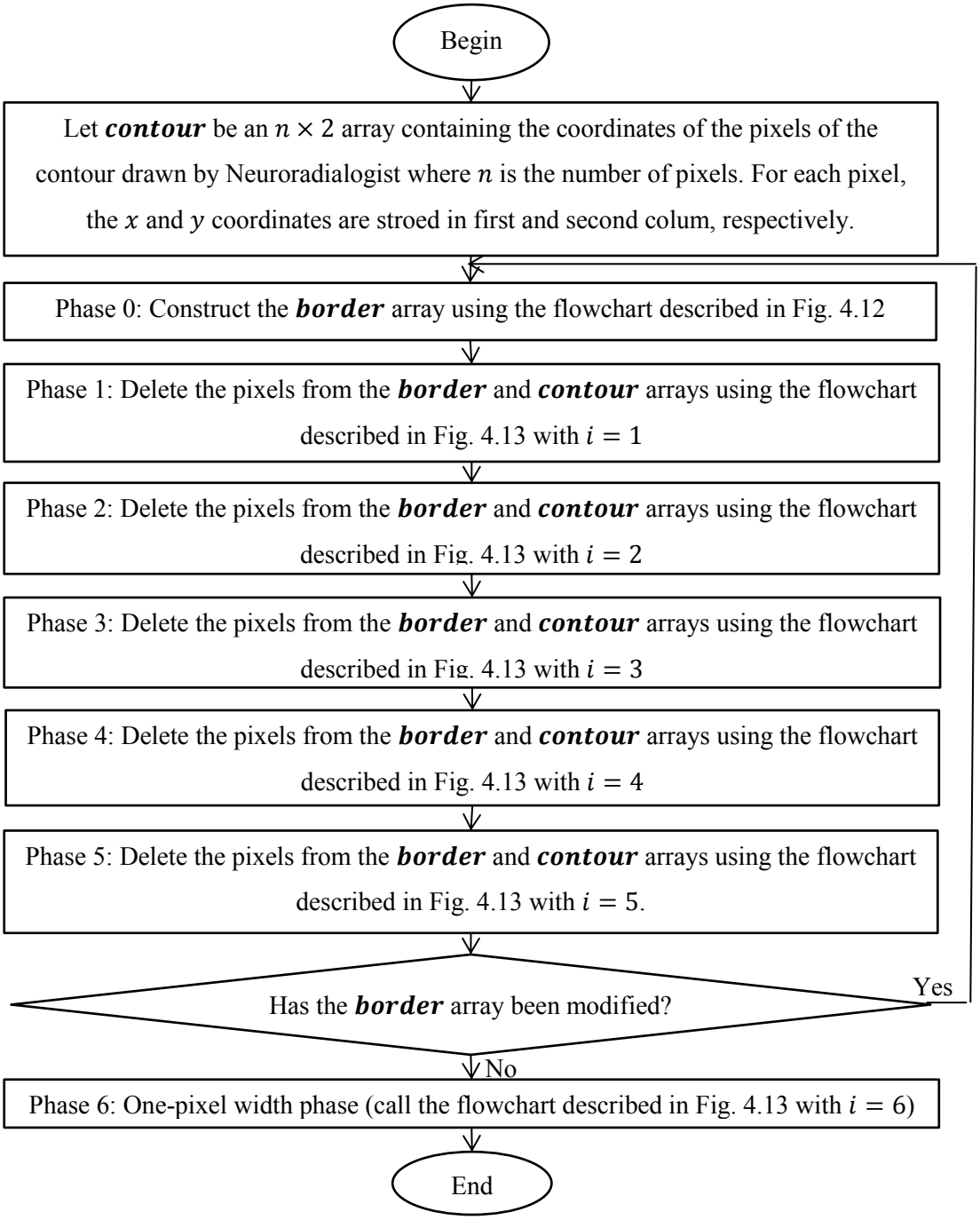


Fig. 4.11 K3M algorithm simplified flowchart

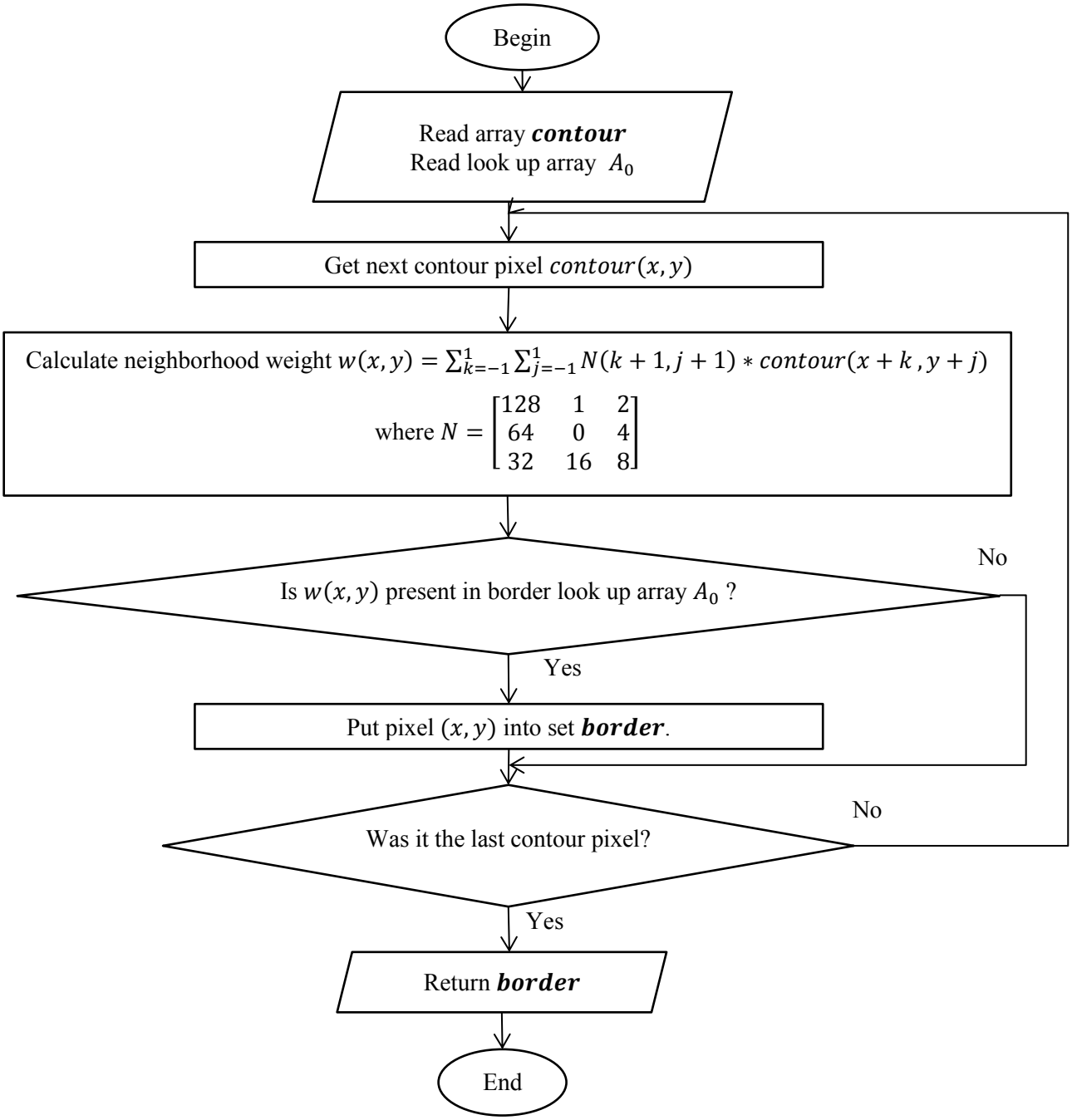


Fig. 4.12 Flowchart of phase 0 (marking border) of K3M algorithm.

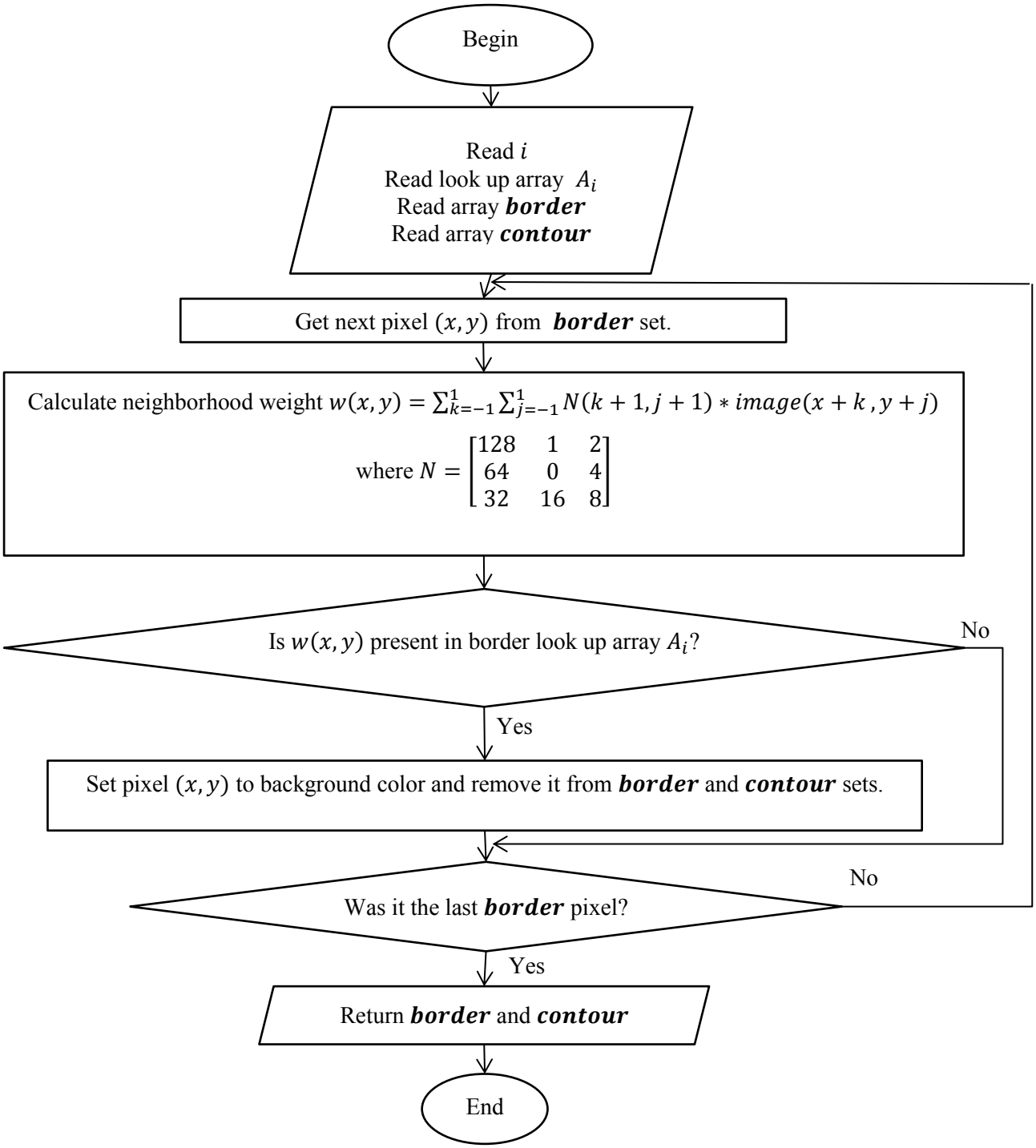
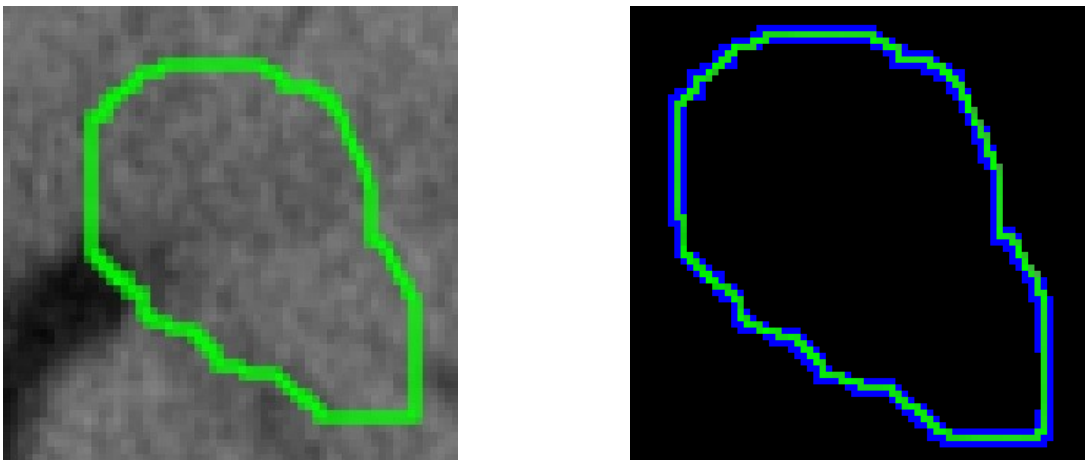


Fig. 4.13 Flowchart of Phase $i = \{1,2,3,4,5,6\}$ deleting pixels.

Fig. 4.14-b shows the result of applying K3M algorithm on the drawn contour presented in Fig. 4.14-a. After skeletonizing the drawn contour, it is passed to the Moore Neighbor Tracing module for further processing.



(a) (b)

Fig. 4.14 User hand Drawn contour on CT image in green color (a); Applying K3M algorithm to make the drawn contour one pixel width (b). Blue pixels will not be considered as part of the contour anymore.

Having made the drawn contour one pixel width, we now pass the contour to Algorithm 4.1 in order to detect the discontinuity of the drawn contour around lesion. Algorithm 4.1 uses the Moore-Neighbor tracing algorithm in order to see if the drawn contour is open and the discontinuity is less than a predefined threshold, say α. In this case, the tool will automatically close the contour; otherwise, the doctor will be asked to specify the lesion area more precisely. Moore-Neighbor tracing algorithm ignores holes in a given pattern and traces the complete outer contour of the pattern (which is a set of connected green pixels in our case).

Algorithm 4.1 Detecting the discontinuity of the drawn contour around lesion

Input: An image matrix , T, containing a connected component P of contour pixels.

1. Define $M(a) = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8\}$ to be the Moore neighborhood of pixel a. The Moore neighborhood of pixel a is the set of 8 pixels which share a vertex or edge with a.
2. Let p denote the current boundary pixel.
3. Let c denote the current pixel under consideration i.e. $c \in M(p)$.
4. Begin

5. Set B to be empty.
6. From bottom to top and left to right scan the elements of T until a contour pixel (a green pixel in our case), $s \in P$ is found.
7. Insert s in B.
8. Set the current boundary point p to s i.e. $p = s$
9. Backtrack i.e. move to the pixel from which s was entered.
10. Set c to be the next clockwise pixel in $M(p)$.
11. While not visiting the start pixel s for a second time in the same direction we originally entered it do
12. If c is a contour pixel (i.e., green pixel)
13. If c already exists in B
14. Go to If_Contour_Is_Closed
15. End if
16. insert c in B
17. set $p = c$
18. backtrack (move the current pixel c to the pixel from which p was entered)
19. Else
20. advance the current pixel c to the next clockwise pixel in $M(p)$
21. End if
22. End While
23. If_Contour_Is_Closed:
24. Let PixelCount be the number of pixels in B
25. Let flag=0 // indicates the contour is open
26. Let α be threshold to ignore the discontinuity of the contour
27. For $i = -\alpha$ to α
28. For $j = -\alpha$ to α
29. If $x_{\mathrm{B}[0]} + i = x_{\mathrm{B}[\mathrm{PixelCount}-1]}$ and $y_{\mathrm{B}[0]} + j = y_{\mathrm{B}[\mathrm{PixelCount}-1]}$
30. flag=1 // indicates the contour is closed
31. Go to Fill_The_Distance
32. End if

33. End for
34. End for
35. Fill_The_Distance:
36. if flag =1
37. Draw a line between $\mathrm{B}[0]$ and $\mathrm{B}[\mathrm{PixelCount}-1]$
38. Else
39. The discontinuity of the contour is more than α pixels. Please draw a closed contour.
40. End if
41. End

5. Software tool experiments, results and conclusions

As mentioned earlier, the aim of this thesis is to present a Radial Basis Functions Neural Network based diagnosis system for automatic identification of Cerebral Vascular Accident through analysis of Computer Tomographic images.

In order to identify the best possible RBF neural network structure and parameters, this work uses a multi-objective neural network models identification method (i.e., please refer to section 2.3 for further explanation). As mentioned the main reason to use a MOGA based neural network is to enable conflicting objectives to be simultaneously considered. For instance, as known neural networks' based classifiers usually present high accuracy for higher model orders; in this study we want to decrease the model complexity and enhance the accuracy of the classification at the same time. Another example is that we want both very small amount of errors in the training set but at the same time being able to select a model with good generalization.

Multi-Objective Genetic Algorithm first finds a non-dominated set of RBFNN models through a number of generations and then selects preferable models from the non-dominated or preferable set.

This chapter starts with explaining how our dataset is produced using the information obtained from the web-based tool described in chapter 4. To obtain the best possible RBFNN classifier, several scenarios were conducted in MOGA as explained in section 5.2. For comparison of the achieved performance we selected the support vector machine approach since it is almost a gold standard comparison on literature reviews; This comparison is described in Section 5.3 enlarging the evaluation of the proposed CAD system. Section 5.4 presents the comparison among our work and two other CAD systems. Section 5.5 reports the results obtained while visualizing abnormal regions in CT images using ensemble of preferable models obtained by MOGA in its best scenario. Section 5.6 discusses the discrimination power of the most frequent features in preferable models of the best scenario conducted in MOGA.

5.1 Producing the dataset

As previously mentioned in section 4.1.1.2, the administrator of the developed web-based tool can download a text file in which the coordinate of each marked pixel is specified. These pixels are

considered as abnormal data samples. Within a CT slice, all the intracranial pixels which are not marked as lesions will be considered as normal data samples. The downloaded text file from our collaborating Neuroradiologist contained 64,786 rows, which means 64,786 pixels were marked as abnormal pixels. In order to obtain the coordinates of all normal pixels, Algorithm 5.1 is used.

Algorithm 5.1 Obtaining the coordinate of normal pixels

Input: text file, say T, shown in Fig. 4.4 (a) in which the coordinate of abnormal pixels and the path from which the image can be retrieved is saved.

1. Let $Exams$ be a structure that is constructed from text file T. $Exams(i)$ contains the information of each CT exam (please see Fig. 5.1)
2. For $i = 1$ to $length(Exams)$
   2.1. Pass $Exams(i)$ to Algorithm 3.1 to remove the skull and other artifacts.
   2.2. For each image in $Exams(i)$
      2.2.1. Let X, Y be two vectors containing the location of abnormal pixels
      2.2.2. Let $P(a, b)$ be the intensity of the pixel located in (a, b)
      2.2.3. If $P(a, b) \neq 0$ and $a \notin X$ and $b \notin Y$
         2.2.3.1. Insert (a, b) and $path(image)$ as a row in a text file O.
            End if
      End for
   End for

Output: text file O

Patient Id						
Exam date						
Image 1						
File name						
Image matrix $\begin{bmatrix} \cdots \\ \vdots \quad \ddots \quad \vdots \\ \cdots \end{bmatrix}$						
X						
Y						
Class: Normal/Abnormal						
.						
.						
Image n						
File name						
Image matrix $\begin{bmatrix} \cdots \\ \vdots \quad \ddots \quad \vdots \\ \cdots \end{bmatrix}$						
X						
Y						
Class: Normal/Abnormal						

Fig. 5.1 $Exams(i)$ structure containing the information about one CT exam.

Applying algorithm 5.1, we obtained 1,802,816 normal pixels. As a result, we have a total of 1,867,602 normal and abnormal pixels to work with.

The next step would be extracting features and producing the dataset. Table 5.1 shows a list of first and second order features together with 10 symmetry features which were previously introduced in section 3.7 and used as our primary feature space. Recall that each CT image is represented as a matrix I with M rows and N columns where $I(m,n)$ stands for the intensity of pixel in row m and column n. The variance of pixel intensities within a window w is denoted by var_w. Given w centred at point (x,y), $\mathrm{L_h}$ is a row vector with the intensities of the 31 pixels taken from the horizontal line centered at (x,y) and $\mathrm{L_v}$ is a column vector with the intensities of the 31 pixels taken from the vertical line centred at (x,y). For calculating features $f15, f16$ and $f38$ to $f41$, $L = 8$ grey levels of histogram of pixel intensities within window w are calculated. Each bin of histogram is represented by H_l. $C(i,j)$ represents the elements of GLCM matrix C. In order to calculate the 8 grey level GLCM of w, the displacement parameters considered were $d = 1$ and $\theta = 0,45,90,135$. As a result 4 GLCM matrices are derived, each belonging to one specific θ and then the average is computed in order to obtain a direction invariant GLCM matrix. In the formulas

used in Table 5.1, the mean value of matrix C is represented by μ and the mean and standard deviation for the rows and columns of C are defined μ_x and σ_x respectively. Moreover, $C_x(i)$ is the i^{th} entry in the marginal-probability matrix obtained by summing the rows of $C(i,j)$ and $C_y(i)$ is obtained by summing the columns of $C(i,j)$. As shown in Table 5.1, symmetry features are calculated for three different window size (i.e., s={11,21,31}).

In order to have a good insight of the data set, appendix A provides an analysis on the discriminative power of each feature, by plotting its corresponding bi-histogram as well as box plot for normal and abnormal groups of pixels.

Table 5.1 Our primary feature space

	Description	**Eq.**
$f1$	$I(x,y)$	(3.1)
$f2$	$\min_{m,n \in w} I(m,n)$	(3.11)
$f3$	$\text{average}_{m,n \in w} I(m,n)$	(3.10)
$f4$	$\max_{m,n \in w} I(m,n)$	(3.12)
$f5$	$\text{median}_{m,n \in w} I(m,n)$	(3.13)
$f6$	$\text{std}_w=(\frac{1}{width(w)\times height(w)-1}\times\sum_{m=x-\frac{(height(w)-1)}{2}}^{x+\frac{(height(w)-1)}{2}}\sum_{n=y-\frac{(width(w)-1)}{2}}^{y+\frac{(width(w)-1)}{2}}(I(m,n)-f3)^2)^{1/2}$	(3.14)
$f7$	$\text{average}_{1\le m\le M,\ 1\le n\le N} I(m,n)$	(3.3)
$f8$	$\text{average}_{m,n \in w} I(m,n) - \text{average}_{1\le m\le M,\ 1\le n\le N} I(m,n)$	(3.15)
$f9$	$I(x,y) - \text{average}_{1\le m\le M,\ 1\le n\le N} I(m,n)$	(3.16)
$f10$	$\text{Plh} = \sum_{n=y-\frac{(width(w)-1)}{2}}^{y+\frac{(width(w)-1)}{2}} \lvert \text{L}_\text{h}(\text{x},\text{n}+1) - \text{L}_\text{h}(\text{x},\text{n}) \rvert$	(3.17)
$f11$	$\text{plv} = \sum_{m=x-\frac{(height(w)-1)}{2}}^{x+\frac{(height(w)-1)}{2}} \lvert \text{L}_\text{v}(\text{m}+1,\text{y}) - \text{L}_\text{v}(\text{m},\text{y}) \rvert$	(3.18)
$f12$	$\text{cxm} = x/512$	(3.2)
$f13$	$\text{Skewness}=\frac{1}{var_w{}^3}\sum_{m=x-\frac{(height(w)-1)}{2}}^{x+\frac{(height(w)-1)}{2}}\sum_{n=y-\frac{(width(w)-1)}{2}}^{y+\frac{(width(w)-1)}{2}}(I(m,n)-f3)^3$	(3.5)

$f14$	Kurtosis=$\frac{1}{var_w{}^4}\sum_{m=x-\frac{(height(w)-1)}{2}}^{x+\frac{(height(w)-1)}{2}}\sum_{n=y-\frac{(width(w)-1)}{2}}^{y+\frac{(width(w)-1)}{2}}(I(m,n)-f3)^4$	(3.6)
$f15$	Energy=$\sum_{l=1}^{L}(\frac{H_l}{width(w)\times height(w)})^2$	(3.7)
$f16$	Entropy=$-\sum_{l=1}^{L}\frac{H_l}{width(w)\times height(w)}\log_2\{\frac{H_l}{width(w)\times height(w)}\}$	(3.8)
$f17$	Autocorrelation=$\sum_{i,j}(ij)C(i,j)$	(3.45)
$f18$	Correlation=$\frac{\sum_{i,j}(ij)C(i,j)-\mu_x\mu_y}{\sigma_x\sigma_y}$	(3.29)
$f19$	Cluster Prominence=$\sum_{i,j}\left(i+j-\mu_x-\mu_y\right)^4C(i,j)$	(3.26)
$f20$	Cluster shade=$\sum_{i,j}\left(i+j-\mu_x-\mu_y\right)^3C(i,j)$	(3.25)
$f21$	Dissimilarity=$\sum_{i,j}\lvert i-j\rvert.C(i,j)$	(3.35)
$f22$	GLCM Energy =$\sum_{i,j}C(i,j)^2$	(3.21)
$f23$	GLCM Entropy=$-\sum_{i,j}C(i,j)\log(C(i,j))$	(3.22)
$f24$	Homogeneity= $\sum_{i,j}\frac{C(i,j)}{1+\lvert i-j\rvert}$	(3.34)
$f25$	Homogeneity = $\sum_{i,j}C(i,j)/(1+(i-j)^2)$	(3.23)
$f26$	Maximum probability=$\underset{i,j}{MAX}\,C(i,j)$	(3.46)
$f27$	Sum of squares =$\sum_{i,j}(i-\mu)^2C(i,j)$	(3.32)
$f28$	Sum average=$\sum_{i=2}^{2G}iC_{x+y}(i)$ where $C_{x+y}(k)=\sum_{i=1}^{G}\sum_{j=1}^{G}C(i,j)\mid i+j=k\,,k=2,3,\dots,2G$	(3.36)
$f29$	Sum variance= $\sum_{i=2}^{2G}(i-f30)^2C_{x+y}(i)$ where $C_{x+y}(k)=\sum_{i=1}^{G}\sum_{j=1}^{G}C(i,j)\mid i+j=k\,,k=2,3,\dots,2G$	(3.38)
$f30$	Sum entropy=$-\sum_{i=2}^{2G}C_{x+y}(i)\log(C_{x+y}(i))$ where $C_{x+y}(k)=\sum_{i=1}^{G}\sum_{j=1}^{G}C(i,j)\mid i+j=k\,,k=2,3,\dots,2G$	(3.37)
$f31$	Difference variance= $variance\ of\ C_{x-y}$ where $C_{x-y}(k)=\sum_{i=1}^{G}\sum_{j=1}^{G}C(i,j)\mid i-j=k\,,k=0,1,\dots,G-1$	(3.39)
$f32$	Difference entropy= $-\sum_{i=0}^{G-1}C_{x-y}(i)\log\left(C_{x-y}(i)\right)$ where $C_{x-y}(k)=\sum_{i=1}^{G}\sum_{j=1}^{G}C(i,j)\mid i-j=k\,,k=0,1,\dots,G-1$	(3.33)

$f33$	Information measure of correlation1= $\frac{f23-HXY1}{\max\{HX,HY\}}$ where HX and HY are Entropies of C_x and C_y and $HXY1=-\sum_{i,j}C(i,j)log\{C_x(i)C_y(j)\}$	(3.40)
$f34$	Information measure of correlation2= $(1-exp[-2.0(HXY2-f23)])^{1/2}$ where $HXY2=-\sum_{i,j}C_x(i)C_y(j)log\{C_x(i)C_y(j)\}$	(3.41)
$f35$	Inverse difference normalized=$\sum_{i,j=1}^{G}\frac{C(i,j)}{1+\lvert i-j\rvert/G}$	(3.44)
$f36$	Inverse difference moment normalized=$\sum_{i,j=1}^{G}\frac{C(i,j)}{1+(i-j)^2/G^2}$	(3.43)
$f37$	$Var_w=\frac{1}{width(w)\times height(w)}\times\sum_{m=x-\frac{(height(w)-1)}{2}}^{x+\frac{(height(w)-1)}{2}}\sum_{n=y-\frac{(width(w)-1)}{2}}^{y+\frac{(width(w)-1)}{2}}(I(m,n)-f3)^2$	(3.4)
$f38$	$(H_1+H_2)/(width(w))^2$	(3.9)
$f39$	$(H_3+H_4)/(width(w))^2$	(3.9)
$f40$	$(H_5+H_6)/(width(w))^2$	(3.9)
$f41$	$(H_7+H_8)/(width(w))^2$	(3.9)
$f42$	$PCC,\quad s=31$	(3.47)
$f43$	$diff$	(3.50)
$f44$	$L_1,\quad s=31$	(3.48)
$f45$	$L_2^2,\quad s=31$	(3.49)
$f46$	$PCC,\quad s=21$	(3.47)
$f47$	$L_1,\quad s=21$	(3.48)
$f48$	$L_2^2,\quad s=21$	(3.49)
$f49$	$PCC,\quad s=11$	(3.47)
$f50$	$L_1,\quad s=11$	(3.48)
$f51$	$L_2^2,\quad s=11$	(3.49)

5.2 Conducted scenarios in MOGA

In order to find the best possible structure of the RBF neural classifier and its corresponding parameters, we used the Multi-Objective Genetic Algorithm approach. In this approach, the system

has to train a considerable amount of RBFNN structures to be able to construct the final non-dominated set (i.e., recall that the training process is done α times for each individual). As a result, in practice, some constraints should be imposed on the size of the dataset that we are providing as the input to MOGA, otherwise the process would be very time consuming. As mentioned in section 5.1, we have 1,867,602 pixels whose status (i.e., normal or abnormal) is already determined by the Neuroradiologists hereafter called BIG_DS. Among these pixels 1,802,816 are normal (96.53% of data samples) and 64,786 are abnormal (3.47% of data samples). Hence, BIG_DS is an imbalanced dataset whose size is $1{,}867{,}602 \times 52$ (i.e., 51 features and 1 target column).

For all experiments, in order to reduce the model complexity of RBFNN models, the system was allowed to choose the number of neurons in the hidden layer and the number of features from ranges $[2{,}30]$ and $[1{,}30]$, respectively. The number of generations and number of individuals in each generation were both set to 100. Early stopping with a maximum number of 100 iterations was used as termination criterion for training of each individual. The number of training trial for each individual (α) was set to 10, and, nearest to origin strategy was used to select the best training trial. The proportion of random immigrants was 10%, the selective pressure was set to 2 and the crossover rate to 0.7, as previous research in [120] has proved that these values are well behaved ones for the mentioned parameters in MOGA.

For all experiments, MOGA objectives are set to $obj = \{FN_{TR}^{0}, FP_{TR}^{0}, FN_{TE}^{0}, FP_{TE}^{0}, MC^{0}\}$. For some experiments, some objectives were set as constraints, which will be pointed out while explaining the corresponding scenario. All the objectives have same priority pr=0.

For each experiment i, the composing subsets of its corresponding dataset $DS(i)$ are represented in a table. Composing subsets are usually represented in the format of $s_{superset}^{class}$ where $class = \{nrm, abnrm\}$ determines whether the data samples of subset s are normal (nrm) or abnormal (abnrm); $superset$ is a set containing all elements of s and $s = \{TR, TE, V, cvh, rndm\}$. TR, TE and V stand for Training, Test and Validation subsets of $DS(i)$. cvh represents data samples that are convex points of $superset$ and $rndm$ represents data samples which are randomly selected from $superset$.

In order to determine the best model of each experiment, we evaluated all obtained models using BIG_DS and then picked the model whose number of False Positive (FP) and False Negative (FN)

on BIG_DS is minimum. It is worthwhile noticing that before feeding BIG_DS to the obtained neural network models, it is normalized column-wise between [-1,1] using $MIN_{DS(i)}$ and $MAX_{DS(i)}$. $MIN_{DS(i)}$ and $MAX_{DS(i)}$ are two 1×52 vectors representing the minimum and maximum values, computed column by column over the $DS(i)$ dataset. $DS(i)$ is the input dataset to MOGA for scenario i. Having the output of RBFNN model, say $\hat{y}$, for a given pixel P at hand, we will consider P as abnormal if $\hat{y} > 0$; otherwise it will be considered as a normal pixel.

5.2.1 Maintaining the ratio within normal and abnormal pixels (Scenario 1)

The dataset of scenario 1, $DS(1)$, has 3000 data samples. This scenario aims to maintain the ratio between normal and abnormal pixels the same as the one appeared in BIG_DS (96.53% normal samples and 3.47% abnormal samples). As a result we are going to select 2896 (i.e., $\frac{3000 \times 96.53}{100} = 2895.9$) normal and 104 (i.e., $\frac{3000 \times 3.47}{100} = 104.1$) abnormal data samples from BIG_DS**.** To help MOGA to use data samples which have extreme values as well, we first selected those data samples whose value in at least one of the features is minimum or maximum (142 data samples; 75 normal and 67 abnormal). Afterwards, we added 2821 (i.e., $2896 - 75$) random normal and 37 (i.e., $104 - 67$) random abnormal data samples to produce $DS(1)$.

To split $DS(1)$ into train, test and validation sets, $DS(1)$ is firstly passed through the Approxhull algorithm [13] to extract its convex points. Convex points are then considered as a part of training set. Incorporating convex points in the training set will help covering the whole range of data where the classifier is going to be used. As it can be seen in Table 5.2., 883 data samples of $DS(1)$ are convex points (i.e., 804 normal and 79 abnormal). Afterwards, random normal and abnormal samples are added to training set to reach its size to 66% of $DS(1)$ in a way to maintain the ratio between normal and abnormal pixels the same as the one appeared in BIG_DS. The remaining data samples are then split into two subsets with equal size to form the test and validation sets.

195 models are in non-dominated set of scenario 1. Tables 5.3 and 5.4 show top 5% models in terms of number of FN and FP in BIG_DS respectively. Highlighted rows are best models. As it can be seen, within BIG_DS, FP rate is very low and the models are correctly classifying most normal pixels but they do not perform well in identifying abnormal pixels (i.e., the smallest rate of FN within BIG_DS is 43% and belongs to model 4311). As a result, one can say that maintaining

the ratio between normal and abnormal pixels the same as the one appeared in BIG_DS it is not a good approach.

Table 5.2 DS(1) specification

<table>
<tr><td rowspan="4">$DS(1) = \{TR_{DS(1)} \cup TE_{DS(1)} \cup V_{DS(1)}\}$
$size(DS(1)) = 3000$</td><td rowspan="2">$TR_{DS(1)} = \{TR^{nrm}_{DS(1)} \cup TR^{abnrm}_{DS(1)}\}$
$size(TR_{DS(1)}) = 2010$</td><td>$TR^{nrm}_{DS(1)} = \{cvh^{nrm}_{DS(1)} \cup rndm^{nrm}_{DS(1)}\}$
$size(TR^{nrm}_{DS(1)}) = 1931$
$size(cvh^{nrm}_{DS(1)}) = 804$
$size(rndm^{nrm}_{DS(1)}) = 1127$</td></tr>
<tr><td>$TR^{abnrm}_{DS(1)} = \{cvh^{abnrm}_{DS(1)} \cup rndm^{abnrm}_{DS(1)}\}$
$size(TR^{abnrm}_{DS(1)}) = 79$
$size(cvh^{abnrm}_{DS(1)}) = 79$
$size(rndm^{abnrm}_{DS(1)}) = 0$</td></tr>
<tr><td colspan="2">$TE_{DS(1)} = \{TE^{nrm}_{DS(1)} \cup TE^{abnrm}_{DS(1)}\}, \quad size(TE_{DS(1)}) = 495$
$size(TE^{nrm}_{DS(1)}) = 478, \quad size(TE^{abnrm}_{DS(1)}) = 17$</td></tr>
<tr><td colspan="2">$V_{DS(1)} = \{V^{nrm}_{DS(1)} \cup V^{abnrm}_{DS(1)}\}, \quad size(V_{DS(1)}) = 495$
$size(V^{nrm}_{DS(1)}) = 487, \quad size(V^{abnrm}_{DS(1)}) = 8$</td></tr>
</table>

Table 5.3 Top 5% models of scenario 1 in terms of number of FN in BIG_DS

Model No.	TR_{DS1}			TE_{DS1}			V_{DS1}			BIG_DS			MC
	FP (%)	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	
6789	0.00	15.19	0.60	0.00	41.18	1.41	0.62	50.00	1.41	0.93	44.30	2.44	161
9681	0.00	7.59	0.30	0.00	35.29	1.21	0.82	62.50	1.82	1.10	44.02	2.59	399
706	0.00	3.80	0.15	0.21	35.29	1.41	0.82	75.00	2.02	1.16	45.17	2.69	437
6991	0.00	5.06	0.20	0.21	29.41	1.21	0.62	62.50	1.62	1.44	46.38	3.00	390
4311	0.00	6.33	0.25	0.00	29.41	1.01	0.41	62.50	1.41	0.99	43.00	2.45	435

1776	0.05	10.13	0.45	0.21	35.29	1.41	0.41	62.50	1.41	0.64	45.07	2.18	221
9644	0.05	8.86	0.40	0.42	29.41	1.41	1.03	62.50	2.02	0.96	44.98	2.49	220
2266	0.00	2.53	0.10	0.00	23.53	0.81	0.21	50.00	1.01	0.97	44.65	2.48	600
2119	0.26	12.66	0.75	0.21	47.06	1.82	0.21	75.00	1.41	0.74	46.31	2.32	114

Table 5.4 Top 5% models of scenario 1 in terms of number of FP in BIG_DS

Model No.	TR_{DS1}			TE_{DS1}			V_{DS1}			BIG_DS			MC
	FP (%)	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	
5080	0.00	74.68	2.94	0.00	64.71	2.22	0.00	100.00	1.62	0.08	80.83	2.88	34
1824	0.10	65.82	2.69	0.00	82.35	2.83	0.00	87.50	1.41	0.13	72.35	2.64	10
5710	0.16	62.03	2.59	0.00	58.82	2.02	0.00	87.50	1.41	0.16	61.78	2.30	16
9129	0.00	73.42	2.89	0.00	94.12	3.23	0.21	87.50	1.62	0.13	88.49	3.20	14
3760	0.00	74.68	2.94	0.00	82.35	2.83	0.00	87.50	1.41	0.11	71.45	2.58	10
6619	0.00	93.67	3.68	0.00	88.24	3.03	0.21	100.00	1.82	0.16	95.72	3.48	6
5338	0.10	75.95	3.08	0.00	88.24	3.03	0.00	100.00	1.62	0.15	83.96	3.06	6
4602	0.05	73.42	2.94	0.00	94.12	3.23	0.21	87.50	1.62	0.15	87.97	3.19	12
5157	0.00	83.54	3.28	0.00	82.35	2.83	0.00	75.00	1.21	0.10	72.51	2.61	8

5.2.2 Balanced amount of normal and abnormal pixels (Scenario 2)

The dataset of scenario 2, $DS(2)$, has 3000 data samples. This scenario aims to maintain the balance between the amount of normal and abnormal pixels as no good models were obtained by maintaining the ratio of normal and abnormal pixels as the one found in $\_DS$. To construct $DS(2)$, we first split BIG_DS into two sub-dataset $BIG_DS^{nrm}$and $BIG_DS^{abnrm}$where BIG_DS^{nrm} and BIG_DS^{abnrm} contain normal and abnormal data samples respectively. Afterwards, for each sub-dataset, those data samples whose value in at least one of the features is minimum or maximum

were selected (144 data samples; 75 normal and 69 abnormal). Finally, we added 1428 random normal and 1428 random abnormal data samples to produce $DS(2)$.

To split $DS(2)$ into train, test and validation sets, $DS(2)$ is firstly applied to the Approxhull algorithm to extract its convex points, which are then considered as a part of training set. As it can be seen in Table 5.5, 835 data samples of $DS(2)$ are convex points (i.e., 500 normal and 335 abnormal). Afterwards, 530 random normal and 615 random abnormal samples are added to training set to reach its size to 66% of $DS(2)$, while maintaining the balance between the amount of normal and abnormal pixels. The remaining data samples are then split into two subsets with equal size to form the test and validation sets.

Table 5.5 DS(2) specification

<table>
<tr><td rowspan="4">$DS(2) = \{TR_{DS(2)} \cup TE_{DS(2)} \cup V_{DS(2)}\}$
$size(DS(2)) = 3000$</td><td rowspan="2">$TR_{DS(2)} = \{TR^{nrm}_{DS(2)} \cup TR^{abnrm}_{DS(2)}\}$
$size(TR_{DS(2)}) = 1980$</td><td>$TR^{nrm}_{DS(2)} = \{cvh^{nrm}_{DS(2)} \cup rndm^{nrm}_{DS(2)}\}$
$size(TR^{nrm}_{DS(2)}) = 1030$
$size(cvh^{nrm}_{DS(2)}) = 500$
$size(rndm^{nrm}_{DS(2)}) = 530$</td></tr>
<tr><td>$TR^{abnrm}_{DS(2)} = \{cvh^{abnrm}_{DS(2)} \cup rndm^{abnrm}_{DS(2)}\}$
$size(TR^{abnrm}_{DS(2)}) = 950$
$size(cvh^{abnrm}_{DS(2)}) = 335$
$size(rndm^{abnrm}_{DS(2)}) = 615$</td></tr>
<tr><td colspan="2">$TE_{DS(2)} = \{TE^{nrm}_{DS(2)} \cup TE^{abnrm}_{DS(2)}\}$, $size(TE_{DS(2)}) = 510$
$size(TE^{nrm}_{DS(2)}) = 228$, $size(TE^{abnrm}_{DS(2)}) = 282$</td></tr>
<tr><td colspan="2">$V_{DS(2)} = \{V^{nrm}_{DS(2)} \cup V^{abnrm}_{DS(2)}\}$, $size(V_{DS(2)}) = 510$
$size(V^{nrm}_{DS(2)}) = 245$, $size(V^{abnrm}_{DS(2)}) = 265$</td></tr>
</table>

674 non-dominated solutions were obtained in scenario 2. Table 5.6 shows 28 models whose number of FPs and FNs is less than 6% in BIG_DS. As it can be seen, comparing to scenario 1, the FN rate in BIG_DS has significantly decreased. Moreover, FN and FP rates within BIG_DS are approximately in the same range. Among models presented in Table 5.6, model 4874 has the least FN rate on BIG_DS, which is 3.92%. The minimum FP rate is 5.33% which belongs to model 2930 but the norm of linear weights for this model is infinity. The second least FP rate 5.36% which corresponds to model 7101.

Table 5.6 Models of scenario 2 whose number of FPs and FNs is less than or equal to 6% in BIG_DS

Model No.	TR_{DS2}			TE_{DS2}			V_{DS2}			BIG_DS			MC
	FP (%)	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	
8368	1.84	2.32	2.07	7.02	5.32	6.08	6.94	3.77	5.29	5.78	4.56	5.74	234
7057	1.36	1.68	1.52	5.26	4.61	4.90	6.12	5.28	5.69	5.97	5.10	5.94	348
7092	1.55	2.00	1.77	6.58	5.67	6.08	6.94	5.28	6.08	5.48	5.39	5.48	323
7101	2.14	3.26	2.68	6.58	4.26	5.29	6.12	4.53	5.29	**5.36**	5.19	5.35	252
2816	1.84	2.42	2.12	5.26	3.90	4.51	7.35	4.53	5.88	5.86	5.32	5.84	357
5438	1.46	2.84	2.12	5.26	5.32	5.29	7.35	3.02	5.10	5.83	5.08	5.80	306
5593	2.23	3.16	2.68	5.70	6.03	5.88	6.12	4.53	5.29	5.73	5.97	5.74	231
7739	0.49	0.84	0.66	5.26	4.61	4.90	6.12	5.28	5.69	5.87	4.90	5.84	580
4874	0.68	0.84	0.76	6.14	3.19	4.51	6.12	4.53	5.29	5.81	**3.92**	5.75	493
6008	1.07	2.53	1.77	4.82	3.90	4.31	8.16	5.28	6.67	5.80	4.41	5.75	480
6421	2.33	3.37	2.83	5.70	4.26	4.90	6.94	3.77	5.29	5.86	5.04	5.83	200
1688	2.23	3.89	3.03	7.02	4.96	5.88	9.39	4.15	6.67	5.75	5.14	5.73	230
98	1.75	3.05	2.37	3.95	4.61	4.31	6.12	5.28	5.69	5.71	5.37	5.70	325
823	1.65	3.37	2.47	4.82	4.61	4.71	5.31	5.28	5.29	5.62	5.54	5.62	350
2146	2.72	2.53	2.63	4.39	3.90	4.12	6.94	3.40	5.10	5.89	4.53	5.84	323
215	2.62	2.42	2.53	4.82	4.26	4.51	5.31	5.28	5.29	5.85	5.14	5.82	286

Model No.	TR_{DS2}			TE_{DS2}			V_{DS2}			BIG_DS			MC
	FP (%)	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	
7337	1.94	4.00	2.93	6.58	4.26	5.29	6.53	4.53	5.49	5.68	5.30	5.67	252
1316	1.36	2.95	2.12	4.39	4.96	4.71	6.53	4.91	5.69	6.00	5.00	5.96	357
1461	2.62	3.68	3.13	4.39	6.03	5.29	6.94	4.91	5.88	5.87	5.79	5.87	220
5020	0.97	2.11	1.52	5.26	4.26	4.71	7.35	4.53	5.88	5.64	4.64	5.61	375
5154	0.87	0.74	0.81	4.82	4.61	4.71	4.90	4.91	4.90	5.94	4.52	5.89	600
5237	0.39	1.05	0.71	5.70	4.26	4.90	6.53	4.53	5.49	5.89	4.23	5.83	609
5782	1.65	2.32	1.97	7.02	3.55	5.10	6.12	4.91	5.49	5.65	4.46	5.61	270
5900	3.01	3.79	3.38	4.39	5.32	4.90	5.71	5.28	5.49	5.81	5.29	5.79	240
5923	0.68	1.16	0.91	5.26	4.26	4.71	6.12	4.15	5.10	5.96	4.96	5.92	456
2930	1.26	2.21	1.72	3.51	3.90	3.73	5.71	5.66	5.69	**5.33**	4.82	5.31	374
332	2.23	2.95	2.58	4.39	4.26	4.31	7.35	6.04	6.67	5.80	5.50	5.79	338
3338	1.46	3.47	2.42	5.70	6.74	6.27	6.53	4.91	5.69	5.61	5.87	5.62	216

In order to obtain better models, we decided to conduct two groups of scenarios based on model 4874 which had 3.92% FN and 5.81% FP rates on BIG_DS. The first group of scenarios is discussed in section 5.2.2.1 and the second group is explained in section 5.2.2.2.

5.2.2.1 Active learning - increasing the size of the training set

The scenarios that are discussed in this subsection try to improve the results obtained by MOGA in Scenario 2 (i.e., introduced in section 5.2.2). All scenarios will alter $\boldsymbol{TR_{DS(2)}}$ by importing new data samples from $\boldsymbol{BIG\_DS}$. There is no restriction on the size of training set while importing new data samples.

5.2.2.1.1 Importing an imbalanced amount of normal and abnormal data samples to the training set (Scenario 3)

In order to construct the dataset of scenario 3, $DS(3)$, we first extracted those data samples from BIG_DS which were falsely classified (i.e., either as normal or abnormal) by model 4874 obtained from scenario 2 and put them in a set named FD_{4874}. 107,295 data samples were placed in FD_{4874}. Afterwards, we applied Approxhull on FD_{4874} from which 6057 convex points were obtained. The idea is to include a portion of these convex points in the training set in order to help MOGA learn from these data samples and hopefully obtain models which are able to correctly classify FD_{4874}. In this scenario we decided to randomly select 10% of the obtained convex points (i.e., 605 data samples) to training set of scenario 2, $TR_{DS(2)}$, and leave the test and validation sets unchanged (i.e., $TE_{DS(3)} = TE_{DS(2)}\ and\ V_{DS(3)} = V_{DS(2)}$). Table 5.7 shows the composing subsets of $DS(3)$. As it can be seen from Table 5.7, among 605 newly added data samples, 551 are normal and 54 are abnormal.

554 models are in non-dominated set of scenario 3. Table 5.8 shows 6 models whose number of FPs and FNs is less than 7% in BIG_DS. As it can be seen in Table 5.8, model 5738 with 3.91% FP and 6.51% FN rate within BIG_DS is the best model. Comparing this result with the best model of scenario 2, we can see that although the FP rate is reduced, the FN rate is increased and the result is not satisfactory. Since in this scenario there was no control over the ratio between newly imported normal and abnormal pixels, we decided to conduct scenario 4, presented in section 5.2.2.1.2, in which a balanced amount of normal and abnormal data samples are imported in the training set. Scenario 4 will be repeated for two times to see if the results will improve.

Table 5.7 DS(3) specification

		$TR^{nrm}_{DS(3)} = \{TR^{nrm}_{DS(2)} \cup cvh^{nrm}_{FD_{4874}}\}$ $size(TR^{nrm}_{DS(3)}) = 1581$ $size(TR^{nrm}_{DS(2)}) = 1030$ $size(cvh^{nrm}_{FD_{4874}}) = 551$

<table>
<tr><td rowspan="3">$DS(3) = \{TR_{DS(3)} \cup TE_{DS(3)} \cup V_{DS(3)}\}$
$size(DS(3)) = 3605$</td><td>$TR_{DS3} = \{TR^{nrm}_{DS(3)} \cup TR^{abnrm}_{DS(3)}\}$
$size(TR_{DS(3)}) = 2585$</td><td>$TR^{abnrm}_{DS(3)} = \{TR^{abnrm}_{DS(2)} \cup cvh^{abnrm}_{FD_{4874}}\}$
$size(TR^{abnrm}_{DS(3)}) = 1004$
$size(TR^{abnrm}_{DS(2)}) = 950$
$size(cvh^{abnrm}_{FD_{4874}}) = 54$</td></tr>
<tr><td colspan="2">$TE_{DS(3)} = \{TE^{nrm}_{DS(3)} \cup TE^{abnrm}_{DS(3)}\}, \quad size(TE_{DS(3)}) = 510$
$size(TE^{nrm}_{DS(3)}) = 228, \quad size(TE^{abnrm}_{DS(3)}) = 282$</td></tr>
<tr><td colspan="2">$V_{DS3} = \{V^{nrm}_{DS(3)} \cup V^{abnrm}_{DS(3)}\}, \quad size(V_{DS(3)}) = 510$
$size(V^{nrm}_{DS(3)}) = 245, \quad size(V^{abnrm}_{DS(3)}) = 265$</td></tr>
</table>

Table 5.8 Models of scenario 3 whose number of FPs and FNs is less than 7% in BIG_DS

Model No.	TR_{DS3}			TE_{DS3}			V_{DS3}			BIG_DS			MC
	FP (%)	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	
6061	1.83	2.69	2.17	4.82	4.61	4.71	4.08	8.30	6.27	3.98	6.83	4.08	672
2970	1.77	3.19	2.32	5.70	3.90	4.71	6.53	6.79	6.67	4.42	6.90	4.51	870
3424	1.39	2.19	1.70	5.70	5.32	5.49	3.27	4.53	3.92	4.46	6.67	4.53	696
4810	1.39	2.39	1.78	5.26	7.80	6.67	5.31	6.42	5.88	4.38	6.94	4.47	720
5738	2.21	5.08	3.33	4.82	5.32	5.10	5.31	6.04	5.69	**3.91**	**6.51**	4.00	462
7082	1.27	2.89	1.90	5.26	4.26	4.71	4.08	6.79	5.49	4.51	6.56	4.58	840

5.2.2.1.2 Importing a balanced amount of normal and abnormal data samples to training set (Scenario 4)

As previously mentioned in section 5.2.2.1.1, FD_{4874} had 6057 convex points (i.e., 5558 normal and 499 abnormal). In order to construct the dataset of scenario 4, $DS(4_a)$, we will randomly select 10% of these convex points (i.e., 606 data samples) to be imported to training set of scenario 2,

$TR_{DS(2)}$, but ,this time, 50% of these newly added data points would be normal and 50% would be abnormal (i.e., 303 normal and 303 abnormal). Test and validation sets are left unchanged (i.e., $TE_{DS(4_a)} = TE_{DS(2)}\ and\ V_{DS(4_a)} = V_{DS(2)}$). Table 5.9 shows the composing subsets of $DS(4_a)$.

Table 5.9 $DS(4_a)$ specification

<table>
<tr><td rowspan="4">$DS(4_a) = \{TR_{DS(4_a)} \cup TE_{DS(4_a)} \cup V_{DS(4_a)}\}$
$size(DS(4_a)) = 3606$</td><td rowspan="2">$TR_{DS(4_a)} = \{TR^{nrm}_{DS(4_a)} \cup TR^{abnrm}_{DS(4_a)}\}$
$size(TR_{DS(4_a)}) = 2586$</td><td>$TR^{nrm}_{DS(4_a)} = \{TR^{nrm}_{DS(2)} \cup cvh^{nrm}_{FD_{4874}}\}$
$size(TR^{nrm}_{DS(4_a)}) = 1333$
$size(TR^{nrm}_{DS(2)}) = 1030$
$size(cvh^{nrm}_{FD_{4874}}) = 303$</td></tr>
<tr><td>$TR^{abnrm}_{DS(3)} = \{TR^{abnrm}_{DS(2)} \cup cvh^{abnrm}_{FD_{4874}}\}$
$size(TR^{abnrm}_{DS(4_a)}) = 1253$
$size(TR^{abnrm}_{DS(2)}) = 950$
$size(cvh^{abnrm}_{FD_{4874}}) = 303$</td></tr>
<tr><td colspan="2">$TE_{DS(4_a)} = \{TE^{nrm}_{DS(4_a)} \cup TE^{abnrm}_{DS(4_a)}\}, \quad size(TE_{DS(4_a)}) = 510$
$size(TE^{nrm}_{DS(4_a)}) = 228$
$size(TE^{abnrm}_{DS(4_a)}) = 282$</td></tr>
<tr><td colspan="2">$V_{DS(4_a)} = \{V^{nrm}_{DS(4_a)} \cup V^{abnrm}_{DS(4_a)}\}, \quad size(V_{DS(4_a)}) = 510$
$size(V^{nrm}_{DS(4_a)}) = 245, \quad size(V^{abnrm}_{DS(4_a)}) = 265$</td></tr>
</table>

617 models are in non-dominated set of scenario 4. Table 5.10 shows 7 models whose number of FPs and FNs is less than 7% in BIG_DS. Comparing the statistics of models shown in Table 5.10 with model 4874 from scenario 2 (i.e., the best model that has been obtained till now), we can see that the FN rate of model 5405 within BIG_DS, which is 3.70%, is less than that of model 4874 from scenario 2 but its FP rate is 6.98% which is bigger than that of model 4874. This result

motivates us to run active learning for the second time to see whether FN and FP rates will continue to reduce.

Table 5.10 Models of scenario 4 whose number of FPs and FNs is less than 7% in BIG_DS

Model No.	TR_{DS1}			TE_{DS1}			V_{DS1}			BIG_DS			MC
	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	
4515	3.30	2.79	3.05	7.46	3.55	5.29	8.57	6.04	7.25	6.76	4.37	6.67	550
5405	1.50	1.04	1.28	10.53	2.48	6.08	4.90	6.04	5.49	6.98	**3.70**	6.86	754
1136	2.55	2.63	2.59	8.33	2.48	5.10	7.35	4.91	6.08	6.67	4.26	6.59	621
791	2.85	1.76	2.32	7.89	2.84	5.10	6.53	4.53	5.49	6.50	4.15	6.42	609
2779	4.20	3.83	4.02	6.58	4.96	5.69	9.39	4.91	7.06	6.85	6.37	6.84	494
8654	3.38	3.51	3.44	6.14	2.48	4.12	6.94	4.53	5.69	**6.44**	4.25	6.36	529
7184	2.55	2.00	2.28	6.14	4.26	5.10	8.98	4.91	6.86	6.59	4.49	6.52	700

To build the dataset for the second round of active learning process, $DS(4_b)$, we need to select one of the good models from Table 5.10. $DS(4_b)$ incorporates data samples which cannot be classified correctly by the selected model. As mentioned before model 5405 has a FN rate of 3.70% in BIG_DS but its FP rate that is the highest one (i.e., 6.98%). On the other hand, model 8654 has the minimum FP rate 6.44% in BIG_DS but its FN rate is 4.25% (i.e., the 3rd best FN rate). Within the models presented in Table 5.10, FN and FP rates of model 791 are both the second best rates (i.e., FP=6.50% and FN=4.15%). Hence, this model is selected to build the (4_b).

In order to construct $DS(4_b)$, we first extracted those data samples from BIG_DS which were falsely classified (i.e., either as normal or abnormal) by model 791 and put them in a set named FD_{791}. 119,886 data samples were placed in FD_{791}. Afterwards, Approxhull is applied on FD_{791} from which 5547 convex points (i.e., 5013 normal and 534 abnormal) were obtained. Finally, 10% of these convex points (i.e., 554 data samples) has been selected to be imported into $TR_{DS(4_a)}$ with the criterion that 50% of these newly added data points would be normal and 50% would be

abnormal (i.e., 277 normal and 277 abnormal). Test and validation sets are left unchanged (i.e., $TE_{DS(4_b)} = TE_{DS(4_a)}\ and\ V_{DS(4_b)} = V_{DS(4_a)}$). Table 5.11 shows the specification of $DS(4_b)$.

Table 5.11 $DS(4_b)$ specification

<table>
<tr><td rowspan="4">$DS(4_b) = \{TR_{DS(4_b)} \cup TE_{DS(4_b)} \cup V_{DS(4_b)}\}$
$size(DS(4_b))$ =4160</td><td rowspan="2">$TR_{DS(4_b)} = \{TR^{nrm}_{DS(4_b)} \cup TR^{abnrm}_{DS(4_b)}\}$
$size(TR_{DS(4_b)}) = 3140$</td><td>$TR^{nrm}_{DS(4_b)} = \{TR^{nrm}_{DS(4_a)} \cup cvh^{nrm}_{FD_{791}}\}$
$size(TR^{nrm}_{DS(4_b)}) = 1610$
$size(TR^{nrm}_{DS(4_a)}) = 1333$
$size(cvh^{nrm}_{FD_{791}}) = 277$</td></tr>
<tr><td>$TR^{abnrm}_{DS(4_b)} = \{TR^{abnrm}_{DS(4_a)} \cup cvh^{abnrm}_{FD_{791}}\}$
$size(TR^{abnrm}_{DS(4_b)}) = 1530$
$size(TR^{abnrm}_{DS(4_a)}) = 1253$
$size(cvh^{abnrm}_{FD_{791}}) = 277$</td></tr>
<tr><td colspan="2">$TE_{DS(4_b)} = \{TE^{nrm}_{DS(4_b)} \cup TE^{abnrm}_{DS(4_b)}\}$, $size(TE_{DS(4_b)}) = 510$
$size(TE^{nrm}_{DS(4_b)}) = 228$, $size(TE^{abnrm}_{DS(4_b)}) = 282$</td></tr>
<tr><td colspan="2">$V_{DS(4_b)} = \{V^{nrm}_{DS(4_b)} \cup V^{abnrm}_{DS(4_b)}\}$, $size(V_{DS(4_b)}) = 510$
$size(V^{nrm}_{DS(4_b)}) = 245$, $size(V^{abnrm}_{DS(4_b)}) = 265$</td></tr>
</table>

821 models are in non-dominated set of Scenario 4, second round. Table 5.12 shows 14 models whose number of FPs and FNs is less than 8% in BIG_DS. As we can see, comparing to first round, the FN rate in BIG_DS has decreased significantly (i.e., FN rate for model 4870 is 1.94%) but FP rate in BIG_DS has not improved at all. The minimum FP rate is 6.84% which belongs to model 2015 but the norm of its linear weights is infinity. The second least FP rate is 7.29% and corresponds to model 8867 which is not good enough.

Table 5.12 Models of scenario 4-second round whose number of FPs and FNs is less than 8% in BIG_DS

Model No.	TR_{DS1}			TE_{DS1}			V_{DS1}			BIG_DS			MC
	FP (%)	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	
6976	2,86	2,09	2,48	8,33	2,84	5,29	8,98	3,02	5,88	7,88	3,05	7,72	630
4870	3,66	1,11	2,42	7,02	1,06	3,73	10,61	1,89	6,08	7,80	**1,94**	7,60	780
2015	2,98	1,37	2,20	7,89	1,77	4,51	10,61	3,77	7,06	**6,84**	2,99	6,71	690
1628	3,66	1,70	2,71	10,53	0,71	5,10	6,53	2,26	4,31	7,72	2,52	7,54	660
2974	3,54	2,35	2,96	9,65	1,77	5,29	8,16	2,26	5,10	7,81	2,58	7,63	551
8867	3,04	1,24	2,17	7,46	2,13	4,51	10,61	1,89	6,08	**7,29**	2,33	7,12	720
1232	3,35	1,70	2,55	9,21	1,77	5,10	11,43	3,40	7,25	7,93	2,33	7,74	624
5750	4,35	1,70	3,06	10,09	1,77	5,49	11,43	2,64	6,86	7,71	3,19	7,55	520
8002	2,11	1,24	1,69	7,02	4,61	5,69	8,98	4,53	6,67	7,74	3,58	7,60	840
6169	3,98	2,88	3,44	12,28	2,48	6,86	11,02	3,02	6,86	7,93	2,76	7,75	450
8087	1,93	1,57	1,75	6,14	1,77	3,73	10,20	2,64	6,27	7,96	3,03	7,79	720
6449	2,55	1,50	2,04	7,02	2,13	4,31	10,61	2,26	6,27	7,99	2,69	7,80	870
7235	3,60	2,81	3,22	11,40	2,84	6,67	10,61	3,77	7,06	7,95	3,29	7,79	504
7064	2,86	2,75	2,80	7,02	2,84	4,71	10,20	2,26	6,08	7,64	3,40	7,49	630

5.2.2.2 Active learning - fixing the size of the training set (Scenario 5)

This subsection presents another experiment which also tries to improve the obtained results of scenario 2 by adding new data samples from BIG_DS to $TR_{DS(2)}$; however, contrary to the strategy

followed in section 5.2.2.1, this time, we are going to keep the size of training set the same as the size of $TR_{DS(2)}$. This experiment will be repeated two times in order to see if the applied strategy will decrease the FP and FN rates in BIG_DS. In order to construct the dataset for the first round of scenario 5, $DS(5_a)$, we will randomly select 10% of convex points of FD_{4874} (i.e., 606 data samples) to be imported to the training set of scenario 2, $TR_{DS(2)}$, 50% of these newly added data points being normal and 50% abnormal (i.e., 303 normal and 303 abnormal). Recall that FD_{4874} is a set containing data samples of BIG_DS which have not been correctly classified by model 4874 of scenario 2. Since we are going to keep the size of $DS(5_a)$ equal to the size of $TR_{DS(2)}$, we have to omit 606 data samples from $DS(5_a)$. These data samples should not belong either to $cvh_{DS(2)}$ or $cvh_{FD_{4874}}$. The test and validation sets are left unchanged (i.e., $TE_{DS(5_a)} = TE_{DS(2)}\ and\ V_{DS(5_a)} = V_{DS(2)}$). Table 5.13 shows the composing subsets of $DS(5_a)$.

Table 5.13 $DS(5_a)$ specification

<table>
<tr>
<td rowspan="2">$DS(5_a) = \{TR_{DS(5_a)} \cup TE_{DS(5_a)} \cup V_{DS(5_a)}\}$
$size(DS(5_a)) = 3000$</td>
<td rowspan="2">$TR_{DS(5_a)} = \{TR^{nrm}_{DS(5_a)} \cup TR^{abnrm}_{DS(5_a)}\}$
$size(TR_{DS(5_a)}) = 1980$</td>
<td>$R1 = TR^{nrm}_{DS(2)} - cvh^{nrm}_{DS(2)}$
$TR^{nrm}_{DS(5_a)} = \{cvh^{nrm}_{DS(2)} \cup cvh^{nrm}_{FD_{4874}} \cup rndm^{nrm}_{R1}\}$
$size(TR^{nrm}_{DS(5_a)}) = 1072$
$size(cvh^{nrm}_{DS(2)}) = 500$
$size(cvh^{nrm}_{FD_{4874}}) = 303$
$size(rndm^{nrm}_{R1}) = 269$</td>
</tr>
<tr>
<td>$R2 = TR^{abnrm}_{DS(2)} - cvh^{abnrm}_{DS(2)}$
$TR^{abnrm}_{DS(5_a)} = \{cvh^{abnrm}_{DS(2)} \cup cvh^{abnrm}_{FD_{4874}} \cup rndm^{abnrm}_{R2}\}$
$size(TR^{abnrm}_{DS(5_a)}) = 908$
$size(cvh^{abnrm}_{DS(2)}) = 335$
$size(cvh^{abnrm}_{FD_{4874}}) = 303$</td>
</tr>
</table>

		$size(rndm_{R2}^{abnrm}) = 270$
	$TE_{DS(5_a)} = \{TE_{DS(5_a)}^{nrm} \cup TE_{DS(5_a)}^{abnrm}\}, \quad size(TE_{DS(5_a)}) = 510$ $size(TE_{DS(5_a)}^{nrm}) = 228, \quad size(TE_{DS(5_a)}^{abnrm}) = 282$	
	$V_{DS(5_a)} = \{V_{DS(5_a)}^{nrm} \cup V_{DS(5_a)}^{abnrm}\}, \quad size(V_{DS(5_a)}) = 510$ $size(V_{DS(5_a)}^{nrm}) = 245, \quad size(V_{DS(5_a)}^{abnrm}) = 265$	

844 models are in non-dominated set of scenario 5 - first round. Table 5.14 shows 5 models whose number of FPs and FNs is less than 8% in BIG_DS. As it is shown, the minimum FP rate in BIG_DS is 7.68% which belongs to model 2972 and the minimum FN rate is 6.59% which corresponds to model 8336. Comparing the results with best model of Scenario 2, we can see that there is no reduction in FP and FN rates.

Although the results are not satisfactory, the second round of active learning process has been conducted to see whether FN and FP rates will reduce. To build the dataset for the second round of scenario 5, $DS(5_b)$, we need to select one of the good models from Table 5.14. $DS(5_b)$ uses data samples which cannot be classified correctly by selected model. Within the two candidates (i.e., models 2972 and 8336) model 2972 was selected due to its smaller FD rate in BIG_DS.

Table 5.14 Models of scenario 5-first round whose number of FPs and FNs is less than 8% in BIG_DS

Model No.	$TR_{DS(5_a)}$			$TE_{DS(5_a)}$			$V_{DS(5_a)}$			BIG_DS			MC
	FP (%)	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	
5718	1,49	1,54	1,52	8,77	4,96	6,67	11,02	7,55	9,22	7,93	6,44	7,88	750
2972	5,69	6,50	6,06	9,21	5,67	7,25	10,20	6,42	8,24	**7,68**	6,96	7,65	255

Model No.	$TR_{DS(5_a)}$			$TE_{DS(5_a)}$			$V_{DS(5_a)}$			BIG_DS			MC
	FP (%)	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	
6677	5,13	4,19	4,70	9,65	3,19	6,08	10,61	6,42	8,43	7,78	7,01	7,75	304
8336	5,04	4,30	4,70	8,77	4,26	6,27	9,39	7,55	8,43	7,88	**6,59**	7,84	405
6767	6,62	8,59	7,53	8,33	6,74	7,45	10,61	7,55	9,02	7,82	7,98	7,83	270

In order to construct $DS(5_b)$, we first extracted those data samples from BIG_DS which were falsely classified (i.e., either as normal or abnormal) by model 2972 and put them in a set named FD_{2972}. 142,960 data samples were placed in FD_{2972}. Afterwards, Approxhull is applied on FD_{2972} from which 4126 convex points (i.e., 3647 normal and 479 abnormal) were obtained. Finally, 10% of these convex points (i.e., 412 data samples) has been selected to be imported into $TR_{DS(5_a)}$ 50% normal and 50% abnormal (i.e., 206 normal and 206 abnormal). Since we are going to keep the size of $DS(5_b)$ equal to the size of $TR_{DS(2)}$, we have to omit 412 data samples from $DS(5_b)$. These data samples are not from $\{cvh_{DS(2)} \cup cvh_{FD_{4874}} \cup cvh_{FD_{2972}}\}$. Test and validation sets are left unchanged (i.e., $TE_{DS(5_b)} = TE_{DS(5_a)}\ and\ V_{DS(5_b)} = V_{DS(5_a)}$). Table 5.15 shows the composing subsets of $DS(5_b)$.

Table 5.15 $DS(5_b)$ specification

$DS(5_b) = \{TR_{DS(5_b)} \cup TE_{DS(5_b)} \cup V_{DS(5_b)}\}$	$TR_{DS(5_b)} = \{TR^{nrm}_{DS(5_b)} \cup TR^{abnrm}_{DS(5_b)}\}$ $size(TR_{DS(5_b)}) = 1980$	$R3 = TR^{nrm}_{DS(5_a)} - \{cvh^{nrm}_{DS(2)} \cup cvh^{nrm}_{FD_{4874}}\}$ $TR^{nrm}_{DS(5_b)} = \{cvh^{nrm}_{FD_{2972}} \cup cvh^{nrm}_{FD_{4874}} \cup cvh^{nrm}_{DS(2)} \cup rndm^{nrm}_{R3}\}$ $size(TR^{nrm}_{DS(5_b)}) = 1072$ $size(cvh^{nrm}_{FD_{2972}}) = 206$ $size(cvh^{nrm}_{FD_{4874}}) = 303$ $size(cvh^{nrm}_{DS(2)}) = 500$ $size(rndm^{nrm}_{R3}) = 63$

<table>
<tr><td rowspan="3">$size\big(DS(5_b)\big)$ =3000</td><td></td><td>$R4 = TR^{abnrm}_{DS(5_a)} - \{cvh^{abnrm}_{DS(2)} \cup cvh^{abnrm}_{FD_{4874}}\}$
$TR^{abnrm}_{DS(5_b)} = \{cvh^{abnrm}_{FD_{2972}} \cup cvh^{abnrm}_{FD_{4874}} \cup cvh^{abnrm}_{DS(2)} \cup rndm^{abnrm}_{R4}\}$
$size\big(TR^{abnrm}_{DS(5_b)}\big) = 908$
$size\big(cvh^{abnrm}_{FD_{2972}}\big) = 206$
$size\big(cvh^{abnrm}_{FD_{4874}}\big) = 303$
$size(cvh^{abnrm}_{DS(2)}) = 335$
$size(rndm^{abnrm}_{R4}) = 64$</td></tr>
<tr><td colspan="2">$TE_{DS(5_b)} = \{TE^{nrm}_{DS(5_b)} \cup TE^{abnrm}_{DS(5_b)}\}, \quad size(TE_{DS(5_b)}) = 510$
$size\big(TE^{nrm}_{DS(5_b)}\big) = 228, \quad size(TE^{abnrm}_{DS(5_b)}) = 282$</td></tr>
<tr><td colspan="2">$V_{DS(5_b)} = \{V^{nrm}_{DS(5_b)} \cup V^{abnrm}_{DS(5_b)}\}, \quad size(V_{DS(5_b)}) = 510$
$size\big(V^{nrm}_{DS(5_b)}\big) = 245, \quad size(V^{abnrm}_{DS(5_b)}) = 265$</td></tr>
</table>

851 models are in non-dominated set of scenario 5- second round. Table 5.16 shows 3 models whose number of FPs and FNs is less than 11% in BIG_DS. As we can see the results are not satisfactory at all.

Table 5.16 Models of scenario 5-second round whose number of FPs and FNs is less than 11% in BIG_DS

Model No.	$TR_{DS(5_b)}$			$TE_{DS(5_b)}$			$V_{DS(5_b)}$			***BIG_DS***			***MC***
	FP (%)	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	
3729	1,03	0,55	0,81	11,40	9,93	10,59	13,06	9,06	10,98	10,54	8,62	10,47	900
6249	3,92	2,64	3,33	11,40	10,28	10,78	13,06	12,45	12,75	10,92	9,53	10,87	435
5504	4,29	2,20	3,33	13,16	4,61	8,43	13,06	4,15	8,43	10,80	5,17	10,60	616

5.2.3 Incorporating a fraction of the convex points of BIG_DS to the training set (Scenario 6)

In this scenario we are going to incorporate a portion of the convex points of BIG_DS to the training set. To construct the dataset of scenario 6, $DS(6)$, Approxhull is applied on BIG_DS from which 13023 convex points (i.e., 1291 abnormal and 11732 normal) were obtained. We decided to incorporate all 1291 abnormal convex points plus 1291 randomly selected normal convex points in the training set of scenario 6, $TR_{DS(6)}$. After excluding $TR_{DS(6)}$ from BIG_DS, 500 data samples were randomly selected to be in $TE_{DS(6)}$ and 500 data samples were randomly selected to be in $V_{DS(6)}$. It should be noted that in both $V_{DS(6)}$ and $TE_{DS(6)}$, 50% of data samples are normal and 50% are abnormal. Table 5.17 shows the structure of $DS(6)$.

Table 5.17 $DS(6)$ specification

$DS(6) = \{TR_{DS(6)} \cup TE_{DS(6)} \cup V_{DS(6)}\}$ $size(DS(6))$ =3582	$R5 = cvh_{BIG\_DS}^{nrm}$ $TR_{DS(6)} = \{cvh_{BIG\_DS}^{abnrm} \cup rndm_{R5}^{nrm}\}, \quad size(TR_{DS(6)}) = 2582$ $size(cvh_{BIG_{DS}}^{abnrm}) = 1291, \quad size(rndm_{R5}^{nrm}) = 1291$
	$TE_{DS(6)} = \{TE_{DS(6)}^{nrm} \cup TE_{DS(6)}^{abnrm}\}, \quad size(TE_{DS(6)}) = 500$ $size(TE_{DS(6)}^{nrm}) = 250, \quad size(TE_{DS(6)}^{abnrm}) = 250$
	$V_{DS(6)} = \{V_{DS(6)}^{nrm} \cup V_{DS(6)}^{abnrm}\}, \quad size(V_{DS(6)}) = 500$ $size(V_{DS(6)}^{nrm}) = 250, \quad size(V_{DS(6)}^{abnrm}) = 250$

746 models are in non-dominated set of scenario 6. Tables 5.18 and 5.19 show the top 1% models in terms of FN and FP rates in BIG_DS, respectively. Highlighted rows represent best models. As it can be seen, within BIG_DS, the FN rate is very low and the models are correctly classifying most abnormal pixels but they do not perform well in identifying normal pixels (i.e., the smallest rate of FP within BIG_DS is 14.91% and belongs to model 8029). Obtaining this high FP rate is probably due to incorporating just convex points in $TR_{DS(6)}$. As a result, in the following

subsections, we are going to conduct some active learning processes within which some random non-convex normal and abnormal data points will be added to $TR_{DS(6)}$ as well.

Table 5.18 Top 1% models of scenario 6 in terms of FN rate in BIG_DS

Model No.	$TR_{DS(6)}$			$TE_{DS(6)}$			$V_{DS(6)}$			BIG_DS			MC
	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	
792	1.63	0.54	1.08	25.20	1.20	13.20	28.80	2.40	15.60	25.06	1.04	24.23	352
956	0.39	0.00	0.19	26.80	1.60	14.20	29.20	2.80	16.00	27.81	1.05	26.88	594
8288	12.39	3.80	8.09	42.00	0.40	21.20	40.00	0.80	20.40	39.02	0.94	37.70	42
5033	4.88	0.93	2.90	31.60	0.00	15.80	35.60	0.80	18.20	33.70	1.00	32.56	187
9443	0.70	0.00	0.35	21.20	0.00	10.60	23.20	0.80	12.00	20.94	0.93	20.25	750
5098	2.32	0.77	1.55	25.20	0.00	12.60	33.60	0.00	16.80	29.74	**0.70**	28.73	338
6056	0.70	0.00	0.35	26.40	0.40	13.40	30.00	1.60	15.80	28.29	0.91	27.34	560
8200	3.87	0.93	2.40	26.80	0.00	13.40	29.60	1.20	15.40	29.51	0.81	28.52	207

Table 5.19 Top 1% models of scenario 6 in terms of FP rate in BIG_DS

Model No.	$TR_{DS(6)}$			$TE_{DS(6)}$			$V_{DS(6)}$			BIG_DS			MC
	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	
1459	10.77	10.46	10.61	18.80	7.60	13.20	20.40	10.00	15.20	16.77	9.29	16.51	32
531	2.56	2.32	2.44	14.40	2.80	8.60	19.60	3.60	11.60	16.93	3.17	16.46	221
4593	3.56	2.40	2.98	14.40	5.60	10.00	18.40	4.80	11.60	16.09	3.94	15.67	153
5045	0.46	0.08	0.27	15.20	4.40	9.80	16.40	6.00	11.20	16.74	3.75	16.29	783
8029	4.88	2.71	3.80	14.00	5.20	9.60	16.40	6.00	11.20	**14.91**	4.83	14.56	190
6786	4.80	4.34	4.57	14.00	6.00	10.00	18.80	8.40	13.60	16.69	5.46	16.30	99

8702	13.32	14.41	13.87	13.60	12.40	13.00	20.00	13.20	16.60	15.70	12.06	15.58	15
9893	7.90	5.96	6.93	17.20	8.00	12.60	17.20	6.00	11.60	17.03	5.69	16.63	35

5.2.3.1 Active learning – Adding random non-convex points to the training set

In order to decrease the FP rate over BIG_DS, we start our work by adding 250 random normal and 250 random abnormal non-convex points into $TR_{DS(6)}$. This lets MOGA learn not only about the boundary data points but also some random inner points. These were obtained from $BIG\_DS - \{cvh_{BIG\_DS} \cup DS(6)\}$. The test and validation sets are left unchanged which means $V_{DS(6_a)} = V_{DS(6)}$ and $TE_{DS(6_a)} = TE_{DS(6)}$ where $DS(6_a)$ is the dataset for the first round of active learning on scenario 6. The structure of $DS(6_a)$ is shown in Table 5.20.

Table 5.20 $DS(6_a)$ specification

$DS(6_a) = \{TR_{DS(6_a)} \cup TE_{DS(6_a)} \cup V_{DS(6_a)}\}$ $size(DS(6_a))$ =4082	$R6 = BIG\_DS - \{cvh_{BIG\_DS} \cup DS(6)\}$ $TR_{DS(6_a)} = \{TR_{DS(6)} \cup rndm_{R6}^{nrm} \cup rndm_{R6}^{abnrm}\}$ $size(TR_{DS(6_a)}) = 3082$, $size(rndm_{R6}^{nrm}) = 250$ $size(rndm_{R6}^{abnrm}) = 250$, $size(TR_{DS(6)}) = 2582$
	$TE_{DS(6_a)} = TE_{DS(6)}$, $size(TE_{DS(6_a)}) = 500$
	$V_{DS(6_a)} = V_{DS(6)}$, $size(V_{DS(6_a)}) = 500$

719 models are in non-dominated set of this experiment. Table 5.21 shows 7 models whose FPs and FNs rates are less than 10% over BIG_DS. As we can see, there is a significant reduction on FP and FN rates over BIG_DS after adding a few number of random non-convex points to $TR_{DS(6)}$. For example, comparing model 4271, shown in Table 5.21 with model 8029, shown in Table 5.19, we can see that there is a 6.23% and 1.72% reduction in the FN and FP rates over BIG_DS respectively.

To further improve, we first focus on reducing the FP rate over BIG_DS by gradually replacing normal convex points with random normal non-convex points in $TR_{DS(6_a)}$ until the results show there would be no further reduction on FP rate over BIG_DS (sections 5.2.3.2 and 5.2.3.3). Afterwards, the focus will be on reducing FN rate by replacing some abnormal convex points with random abnormal non-convex points in $TR_{DS(6_a)}$ (sections 5.2.3.4).

Table 5.21 Models of scenario 6-first round of active learning whose number of FPs and FNs is less than 10% in BIG_DS

Model No.	$TR_{DS(6_a)}$			$TE_{DS(6_a)}$			$V_{DS(6_a)}$			BIG_DS			MC
	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	
1417	0,97	0,13	0,55	5,60	2,80	4,20	11,60	3,60	7,60	9,64	2,11	9,37	567
3518	3,05	2,27	2,66	8,00	4,80	6,40	9,60	5,20	7,40	9,80	4,74	9,63	240
4271	0,97	0,32	0,65	6,40	4,80	5,60	8,00	3,60	5,80	**8,68**	**3,11**	8,48	506
754	1,36	0,84	1,10	8,80	4,00	6,40	12,80	3,60	8,20	9,78	3,66	9,56	368
7588	0,97	0,19	0,58	9,60	2,40	6,00	10,00	4,80	7,40	9,92	3,06	9,69	560
7020	2,01	1,88	1,95	6,40	5,20	5,80	9,60	4,80	7,20	9,88	4,61	9,70	390
8816	2,66	2,21	2,43	6,00	5,60	5,80	8,40	4,40	6,40	9,88	4,36	9,69	256

5.2.3.2 Active learning – Substituting a fraction of normal convex points with normal non-convex points

As mentioned earlier, in this section, to further reduce FP rate over BIG_DS, 500 normal convex points within $TR_{DS(6_a)}$ have been replaced by 500 randomly selected normal non-convex points from $BIG\_DS - \{DS(6_a) \cup cvh_{BIG\_DS}\}$. Test and validation sets are left unchanged which means $V_{DS(6_b)} = V_{DS(6_a)}$ and $TE_{DS(6_b)} = TE_{DS(6_a)}$ where $DS(6_b)$ is the dataset of this experiment represented in Table 5.22.

Table 5.22 $DS(6_b)$ specification

$DS(6_b) = \{TR_{DS(6_b)} \cup TE_{DS(6_b)} \cup V_{DS(6_b)}\}$ $size(DS(6_b))$ =4082	$R7 = cvh^{nrm}_{TR_{DS(6_a)}}$ $R8 = BIG\_DS - \{DS(6_a) \cup cvh_{BIG\_DS}\}$ $TR_{DS(6_b)} = \{TR_{DS(6_a)} - rndm_{R7}\} \cup rndm^{nrm}_{R8}$ $size(TR_{DS(6_b)}) = 3082$, $size(TR_{DS(6_a)}) = 3082$ $size(rndm_{R7}) = 500$, $size(rndm^{nrm}_{R8}) = 500$
	$TE_{DS(6_b)} = TE_{DS(6_a)}$, $size(TE_{DS(6_b)}) = 500$
	$V_{DS(6_b)} = V_{DS(6_a)}$, $size(V_{DS(6_b)}) = 500$

623 models are in non-dominated set of this experiment. Table 5.23 shows 17 models whose FP and FN rates are less than 6% over BIG_DS. As it is shown in Table 5.23, we have succeeded to reduce the FP rate over BIG_DS by importing some randomly selected normal non-convex points. Comparing model 1951 in Table 5.23 with model 4271 in Table 5.21, one can see a reduction of 4.06% on FP rate over BIG_DS; however, the FN rate has increased by 2.21%

Table 5.23 Models of scenario 6-second round of active learning whose FP and FN rates are less than 6% in BIG_DS

Model No.	$TR_{DS(6_b)}$			$TE_{DS(6_b)}$			$V_{DS(6_b)}$			***BIG_DS***			***MC***
	FP (%)	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	
5401	1,43	1,04	1,23	5,20	5,20	5,20	8,00	4,80	6,40	5,96	4,86	5,93	570
4555	1,30	1,30	1,30	6,00	4,80	5,40	6,80	5,60	6,20	5,98	5,68	5,97	450
8599	0,91	1,17	1,04	6,40	5,20	5,80	6,40	7,20	6,80	5,94	5,72	5,94	540
2133	0,84	1,04	0,94	4,80	3,20	4,00	7,20	5,20	6,20	5,76	5,17	5,74	676

Model No.	$TR_{DS(6_b)}$			$TE_{DS(6_b)}$			$V_{DS(6_b)}$			BIG_DS			MC
	FP (%)	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	
4128	0,52	0,58	0,55	4,40	4,40	4,40	6,40	5,60	6,00	5,94	5,26	5,92	756
9732	1,30	1,30	1,30	4,40	4,40	4,40	6,80	6,40	6,60	5,93	5,55	5,92	513
5991	1,69	1,69	1,69	4,80	3,60	4,20	4,00	6,80	5,40	5,59	5,66	5,59	350
7358	0,71	0,32	0,52	6,80	6,40	6,60	6,40	5,60	6,00	5,83	4,99	5,80	720
4191	1,56	2,01	1,78	6,40	3,60	5,00	7,20	7,20	7,20	5,19	4,63	5,17	504
4820	1,23	1,75	1,49	5,60	5,60	5,60	6,40	4,80	5,60	5,29	5,76	5,30	493
264	1,56	1,62	1,59	5,20	5,60	5,40	7,20	9,20	8,20	5,98	5,42	5,96	425
120	1,82	2,21	2,01	4,00	6,00	5,00	5,60	6,80	6,20	5,83	5,84	5,83	510
4038	0,97	1,82	1,40	6,80	4,00	5,40	4,80	5,60	5,20	5,83	5,98	5,83	504
1354	0,39	0,65	0,52	6,40	4,80	5,60	7,60	6,00	6,80	5,95	4,69	5,90	667
29	0,91	0,52	0,71	5,20	4,80	5,00	6,40	6,00	6,20	5,84	5,43	5,83	644
1951	1,04	0,58	0,81	4,40	5,60	5,00	6,80	6,00	6,40	**4,62**	**5,32**	4,65	609
2891	2,60	2,60	2,60	5,20	5,60	5,40	5,60	6,80	6,20	5,98	5,72	5,97	325

5.2.3.3 Active learning – Substituting a fraction of normal convex points with normal non-convex points

To see whether replacing further normal convex points with normal random non-convex points in $TR_{DS(6_a)}$ will result in having smaller FP rate over BIG_DS, we conducted a second round of replacements. In this experiment 250 normal convex points within $TR_{DS(6_b)}$ have been replaced by 250 randomly selected normal non-convex points from $BIG\_DS - \{cvh_{BIG\_DS} \cup DS(6_b)\}$. Test

and validation sets are left unchanged which means $V_{DS(6_c)} = V_{DS(6_b)}\ and\ TE_{DS(6_c)} = TE_{DS(6_b)}$ where $DS(6_c)$ is the dataset of this experiment represented in Table 5.24.

Table 5.24 $DS(6_c)$ specification

$DS(6_c) = \{TR_{DS(6_c)} \cup TE_{DS(6_c)} \cup V_{DS(6_c)}\}$ $size(DS(6_c))$ =4082	$R9 = cvh^{nrm}_{TR_{DS(6_b)}}$ $R10 = BIG\_DS - \{DS(6_b) \cup cvh_{BIG\_DS}\}$ $TR_{DS(6_c)} = \{TR_{DS(6_b)} - rndm_{R9}\} \cup rndm^{nrm}_{R10}$ $size(TR_{DS(6_c)}) = 3082$, $size(TR_{DS(6_b)}) = 3082$ $size(rndm_{R9}) = 250$, $size(rndm^{nrm}_{R10}) = 250$
	$TE_{DS(6_c)} = TE_{DS(6_b)}$, $size(TE_{DS(6_c)}) = 500$
	$V_{DS(6_c)} = V_{DS(6_b)}$, $size(V_{DS(6_c)}) = 500$

642 models are in non-dominated set of this experiment. Table 5.25 shows 5 models whose FP rate is less than 5% over BIG_DS. As it can be seen in Table 5.25, model 9710 has the minimum FP rate over BIG_DS which is 4.27%; but its norm of linear weights is infinity. As a result we look for the second least FP rate over BIG_DS within Table 5.25 which belongs to model 430 and is equal to 4.70%. Comparing model 430 with model 1951 from the previous experiment, shown in Table 5.23, one can see that there is no improvement in FP and FN rates over BIG_DS. Hence, we will stop the process of replacing normal convex points with random normal non-convex points in $TR_{DS(6_a)}$ and start to focus on reducing FN rate over BIG_DS.

Table 5.25 Models of scenario 6-third round of active learning whose of FP rate is less than 5% in BIG_DS

Model No.	$TR_{DS(6_c)}$			$TE_{DS(6_c)}$			$V_{DS(6_c)}$			BIG_DS			MC
	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	
430	2,66	4,28	3,47	3,20	10,80	7,00	6,00	10,00	8,00	**4,70**	**8,61**	4,83	300
7281	0,71	0,78	0,75	3,60	6,00	4,80	5,20	8,40	6,80	4,93	7,17	5,00	690
4850	2,40	1,88	2,14	6,40	6,00	6,20	4,80	7,20	6,00	4,85	6,37	4,90	400
9710	1,04	1,95	1,49	4,80	8,40	6,60	4,80	8,40	6,60	**4,27**	7,76	4,39	480
9250	0,97	0,97	0,97	4,40	6,00	5,20	5,20	6,80	6,00	4,94	6,04	4,98	621

5.2.3.4 Active learning – Substituting a fraction of abnormal convex points with abnormal non-convex points

In this experiment we are going to replace 250 abnormal convex points in $TR_{DS(6_c)}$ with 250 randomly selected abnormal non-convex points from $BIG\_DS - \{cvh_{BIG\_DS} \cup DS(6_c)\}$ in order to reduce FN rate over BIG_DS. Test and validation sets are left unchanged which means $V_{DS(6_d)} = V_{DS(6_c)}$ and $TE_{DS(6_d)} = TE_{DS(6_c)}$ where $DS(6_d)$ is the dataset of this experiment represented in Table 5.26.

Table 5.26 $DS(6_d)$ specification

$DS(6_d) = \{TR_{DS(6_d)} \cup TE_{DS(6_d)} \cup V_{DS(6_d)}\}$ $size(DS(6_d))$ =4082	$R11 = cvh^{abnrm}_{TR_{DS(6_c)}}$ $R12 = BIG\_DS - \{DS(6_c) \cup cvh_{BIG\_DS}\}$ $TR_{DS(6_d)} = \{TR_{DS(6_c)} - rndm_{R11}\} \cup rndm^{abnrm}_{R12}$ $size(TR_{DS(6_d)}) = 3082$, $size(TR_{DS(6_c)}) = 3082$ $size(rndm_{R11}) = 250$, $size(rndm^{abnrm}_{R12}) = 250$

	$TE_{DS(6_d)} = TE_{DS(6_c)}$, $size(TE_{DS(6_d)}) = 500$
	$V_{DS(6_d)} = V_{DS(6_c)}$, $size(V_{DS(6_d)}) = 500$

646 models are in non-dominated sat of this experiment. Tables 5.27 and 5.28 show the top 1% models in terms of FN and FP rates in BIG_DS respectively. Highlighted rows represent best models. As it can be seen, replacing some abnormal convex point with randomly selected abnormal non-convex points, has reduced the FN rate over BIG_DS, achieving 3.17% for model 2561, whose FP rate is 6.16%. Since we have to select a model which performs well on classifying both normal and abnormal pixels, model 9442, shown in Table 5.28, might be a better choice. The FP and FN rates of this model over BIG_DS are 4.98% and 4.99% respectively.

Table 5.27 Top 1% models of scenario 6 - 4th round of active learning in terms of FN rate in BIG_DS

Model No.	$TR_{DS(6_d)}$			$TE_{DS(6_d)}$			$V_{DS(6_d)}$			BIG_DS			MC
	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	FP (%)	FN (%)	FD (%)	
2529	1,56	1,04	1,30	8,00	2,80	5,40	6,40	4,80	5,60	5,99	3,64	5,91	464
5356	1,04	0,84	0,94	7,20	1,60	4,40	9,60	2,40	6,00	6,08	3,30	5,98	648
2561	0,84	0,58	0,71	7,20	2,40	4,80	4,40	2,80	3,60	**6,16**	**3,17**	6,05	522
7346	1,43	0,84	1,14	9,60	1,20	5,40	6,80	6,00	6,40	7,24	3,62	7,12	609
2506	1,30	0,58	0,94	6,40	2,80	4,60	8,00	3,60	5,80	7,18	3,33	7,04	660
6721	1,49	0,91	1,20	6,00	2,40	4,20	7,60	4,40	6,00	7,00	3,65	6,89	494
8074	2,40	1,17	1,78	10,00	4,00	7,00	9,60	3,60	6,60	7,68	3,60	7,54	440

Table 5.28 Top 1% models of scenario 6 -4th round of active learning in terms of FP rate in BIG_DS

Model No.	$TR_{DS(6_d)}$			$TE_{DS(6_d)}$			$V_{DS(6_d)}$			BIG_DS			MC
	FP (%)	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	
4715	1,43	2,08	1,75	4,80	4,80	4,80	3,20	4,80	4,00	4,94	5,50	4,96	522
347	1,95	2,60	2,27	5,60	5,60	5,60	3,60	4,40	4,00	5,41	5,22	5,40	325
4092	1,69	2,53	2,11	6,00	3,60	4,80	7,20	5,20	6,20	5,43	4,87	5,41	400
4704	2,40	3,63	3,02	4,00	8,00	6,00	4,00	5,60	4,80	5,22	6,75	5,28	272
1718	2,79	5,06	3,93	4,80	10,40	7,60	6,40	6,40	6,40	5,43	7,99	5,52	216
9442	1,88	2,73	2,30	5,60	3,20	4,40	5,20	5,20	5,20	**4,98**	**4,99**	4,98	357
7592	1,49	2,14	1,82	6,00	4,80	5,40	5,60	4,40	5,00	5,23	4,75	5,22	440

In order to see if we could further improve the obtained result of this experiment, some restrictions will be used for FP_{TR} and FN_{TR} objectives. We are going to restrict the number of False Positives and False Negatives in the training set to 3.5% for normal points (i.e., $\frac{1541\times3.5}{100} \cong 54$) and 1.5% for abnormal points (i.e., $\frac{1541\times1.5}{100} \cong 23$) in $TR_{DS(6_d)}$ respectively. As a result, 556 models have obtained as non-dominated models. Table 5.29 shows 7 non-dominated models whose FP and FN rates over BIG_DS is less than 5.5%. As it can be seen, the minimum FP rate is 5.10% and belongs to model 937 with FN rate 4.81%. The minimum FN rate is 3.78% which corresponds to model 1689 with a FP rate 5.44%. The result shows that incorporating restrictions on FP_{TR} and FN_{TR} objectives has not resulted in further improvement.

Table 5.29 Models of scenario 6-4th round of active learning with restricted MOGA objectives whose of FP and FN rates over BIG_DS are less than 5.5%

Model No.	$TR_{DS(6_d)}$			$TE_{DS(6_d)}$			$V_{DS(6_d)}$			BIG_DS			MC
	FP (%)	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	
3230	0,52	0,52	0,52	5,20	2,80	4,00	5,20	4,80	5,00	5,40	4,52	5,37	690
1689	1,17	1,43	1,30	4,80	2,80	3,80	5,20	4,80	5,00	5,44	**3,78**	5,38	560
937	1,10	1,43	1,27	3,60	4,40	4,00	5,60	6,00	5,80	**5,10**	4,81	5,09	500
1371	0,65	0,84	0,75	3,60	3,60	3,60	5,60	4,00	4,80	5,42	5,20	5,41	780
7014	1,10	0,78	0,94	5,20	4,00	4,60	6,00	6,40	6,20	5,17	4,54	5,15	552
6903	1,88	2,60	2,24	4,80	5,60	5,20	6,00	5,20	5,60	5,47	5,27	5,46	442
9631	1,10	1,23	1,17	4,40	6,00	5,20	8,00	6,40	7,20	5,36	4,94	5,34	550

5.2.4 Using all convex points of the whole dataset in MOGA (Scenario 7)

In this experiment all convex points of BIG_DS are included in training set. Recall that after applying Approxhull on BIG_DS, 13023 convex points have been obtained, within which 1291 data samples are abnormal and 11732 are normal. These convex points along with 6977 random data samples (50% normal and 50% abnormal) constitute our training set whose size is 20,000. After excluding training data samples from BIG_DS, 6666 random data samples were selected as test and 6666 random data samples were selected as validation sets. In both test and validation sets 50% of data samples are normal and 50% are abnormal. As a result, the MOGA input dataset of this experiment, $DS(7)$, has 33,332 data samples including 60% training, 20% test and 20% validation data samples. Table 5.30 shows the composition of $DS(7)$.

Table 5.30 $DS(7)$ specification

$DS(7) = \{TR_{DS(7)} \cup TE_{DS(7)} \cup V_{DS(7)}\}$ $size(DS(7)) = 33332$	$R13 = BIG\_DS - cvh_{BIG\_DS}$ $TR_{DS(7)} = \{cvh_{BIG\_DS}^{nrm} \cup cvh_{BIG\_DS}^{abnrm} \cup rndm_{R13}^{nrm} \cup rndm_{R13}^{abnrm}\}$ $size(TR_{DS(7)}) = 20000$, $size(cvh_{BIG\_DS}^{nrm}) = 11732$ $size(cvh_{BIG\_DS}^{abnrm}) = 1291$, $size(rndm_{R13}^{nrm}) = 3488$ $size(rndm_{R13}^{abnrm}) = 3489$
	$R14 = BIG\_DS - TR_{DS(7)}$ $TE_{DS(7)} = \{rndm_{R14}^{nrm} \cup rndm_{R14}^{abnrm}\}$ $size(TE_{DS(7)}) = 6666$, $size(rndm_{R14}^{nrm}) = 3333$ $size(rndm_{R14}^{abnrm}) = 3333$
	$R15 = BIG\_DS - \{TR_{DS(7)} \cup TE_{DS(7)}\}$ $V_{DS(7)} = \{rndm_{R15}^{nrm} \cup rndm_{R15}^{abnrm}\}$, $size(V_{DS(7)}) = 6666$ $size(rndm_{R15}^{nrm}) = 3333$, $size(rndm_{R15}^{abnrm}) = 3333$

406 models were in non-dominated set of this experiment. (i.e., since there are no restrictions on the objectives of this experiment, its preferable set is the same as non-dominated set). Table 5.31 shows Minimum, average and maximum FP and FN rates as well as model complexity of 406 non-dominated models of scenario 7.

Table 5.31Min, Avg. and Max false positive and false negative rates as well as model complexity of 406 non-dominated models of scenario 7.

	TR_{MOGA_DS}			TE_{MOGA_DS}			V_{MOGA_DS}			BIG_DS			MC
	FP (%)	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	
Min.	0	1.86	1.08	0	1.80	2.39	0	2.28	2.91	0	2.20	2.33	6
Avg.	2.13	23.41	7.21	3.83	21.50	12.67	4.16	21.60	12.88	4.09	21.78	4.71	199.8

Max.	8.47	100	24.16	12.27	100	50.03	13.47	100	50	12.49	100	12.74	900

Table 5.32 shows 2 models whose FP and FN rates over BIG_DS are less than 3%. As it can be seen, we have obtained a significant improvement comparing to all previous scenarios. Both models 1371 and 6009 have an equal percentage of FP (i.e., 2.96%) within BIG_DS but the FN percentage of model 1371 within BIG_DS is a bit smaller than that of model 6009.

Table 5.32 Models of scenario 7 whose of FP and FN rates over BIG_DS are less than 3%

Model No.	$TR_{DS(7)}$			$TE_{DS(7)}$			$V_{DS(7)}$			BIG_DS			MC
	FP (%)	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	
1371	0.80	2.70	1.25	2.43	2.58	2.51	3.27	3.24	3.26	**2.96**	**2.88**	2.96	702
6009	0.74	2.53	1.17	2.85	2.46	2.66	3.36	3.06	3.21	**2.96**	**2.89**	2.96	870

In order to see if we could further improve the obtained result of this experiment, we conducted another experiment, $scenario\ 7_b$, in which the FP_{TR} and FN_{TR} objectives have been restricted based on the statistics of the best model obtained in scenario 7 (i.e., model 1371). As it can be seen from Table 5.33, model 1371 has 121 FP and 129 FN within $TR_{DS(7)}$. As a result, FP_{TR} and FN_{TR} MOGA objectives have been restricted to these values (i.e., $FP_{TR} < 121\ and\ FN_{TR} < 129$).

Table 5.33 The model of scenario 7 whose statistics were used as restriction.

Model No.	$TR_{DS(7)}$			$TE_{DS(7)}$			$V_{DS(7)}$			BIG_DS			MC
	FP	*FN*	*FD*	*FP*	*FN*	*FD*	*FP*	*FN*	*FD*	*FP*	*FN*	*FD*	
1371	121	129	250	81	86	167	109	108	217	53442	1868	55310	702

The non-dominated set of $scenario\ 7_b$ contains 281 models from which 69 models are in preferable set. Table 5.34 shows Minimum, average and maximum FP and FN rates as well as model complexity of 69 models in preferable set of $scenario\ 7_b$.

Table 5.34. Min, Avg. and Max false positive and false negative rates as well as model complexity of 69 models in preferable set of $scenario\ 7_b$.

	TR_{MOGA_DS}			TE_{MOGA_DS}			V_{MOGA_DS}			BIG_DS			MC
	FP (%)	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	
Min.	0.49	1.38	0.77	2.19	2.06	2.36	2.24	1.75	2.05	2.40	1.98	2.40	750
Avg.	0.60	1.90	0.89	2.76	2.65	2.71	2.71	2.45	2.58	2.78	2.43	2.76	862.3
Max.	0.67	2.51	1.04	3.37	3.37	3.25	3.24	2.97	2.97	3.20	2.91	3.17	900

Table 5.35 shows 4 models whose FP and FN rates over BIG_DS are less than 2.6%. Analyzing Table 5.35, one can see that incorporating restrictions on FN_{TR} and FP_{TR} resulted in models with of the smallest FP and FN rates on BIG_DS. Among them, model 3726 has the minimum percentage of FP and model 3055 has minimum percentage of FN on BIG_DS.

Table 5.35 Models of $scenario\ 7_b$ whose FP and FN rates over BIG_DS are less than 2.6%

Model No.	$TR_{DS(7_b)}$			$TE_{DS(7_b)}$			$V_{DS(7_b)}$			BIG_DS			MC
	FP (%)	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	
3726	0.60	1.79	0.87	2.27	2.90	2.58	2.58	2.37	2.48	**2.40**	**2.34**	2.40	870
4812	0.60	1.87	0.89	2.71	2.30	2.50	2.61	2.71	2.66	2.60	2.43	2.59	900
3863	0.59	1.52	0.80	2.32	2.71	2.52	2.43	2.56	2.49	2.55	2.45	2.55	900
3055	0.50	1.73	0.77	2.71	2.84	2.78	2.43	2.09	2.26	2.56	**2.31**	2.55	900

5.2.5 Comparing best models of all scenarios

Table 5.36 shows the statistics of best models of all conducted scenarios considering the summary of all scenarios tested as follows:

1. Maintaining the ratio within normal and abnormal pixels (Scenario 1).

2. Balanced amount of normal and abnormal pixels (Scenario 2).
    a. Active learning - increasing the size of the training set:
        i. Importing an imbalanced amount of normal and abnormal data samples to the training set (Scenario 3).
        ii. Importing a balanced amount of normal and abnormal data samples to training set:
            1. First round (Scenario 4_a).
            2. Second round (Scenario 4_b).
    b. Active learning - fixing the size of the training set:
        i. First round (Scenario 5_a).
        ii. Second round (Scenario 5_b).
3. Incorporating a fraction of the convex points of BIG_DS to the training set (Scenario 6).
    a. Active learning – Adding random non-convex points to the training set (Scenario 6_a).
    b. Active learning – Substituting a fraction of normal convex points with normal non-convex points:
        i. First round (Scenario 6_b).
        ii. Second round (Scenario 6_c).
    c. Active learning – Substituting a fraction of abnormal convex points with abnormal non-convex points:
        i. Without any restriction on MOGA objectives (Scenario 6_d).
        ii. Restrict MOGA objectives based on the best model obtained in Scenario 6_d.
4. Using all convex points of the whole dataset in MOGA:
    a. Without any restriction on MOGA objectives (Scenario 7).
    b. Restrict MOGA objectives based on the best model obtained in Scenario 7 (Scenario 7_b).

Table 5.36 Comparing best models of all scenarios

Scenario	Model No.	TR_{DS1}			TE_{DS1}			V_{DS1}			BIG_DS			MC
		FP (%)	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	
1	4311	0.00	6.33	0.25	0.00	29.41	1.01	0.41	62.50	1.41	0.99	**43.00**	2.45	435
1	5080	0.00	74.68	2.94	0.00	64.71	2.22	0.00	100.00	1.62	**0.08**	80.83	2.88	34
2	4874	0.68	0.84	0.76	6.14	3.19	4.51	6.12	4.53	5.29	**5.81**	**3.92**	5.75	493
3	5738	2.21	5.08	3.33	4.82	5.32	5.10	5.31	6.04	5.69	**3.91**	**6.51**	4.00	462
4_a	791	2.85	1.76	2.32	7.89	2.84	5.10	6.53	4.53	5.49	**6.50**	**4.15**	6.42	609
4_b	4870	3,66	1,11	2,42	7,02	1,06	3,73	10,61	1,89	6,08	7,80	**1,94**	7,60	780
4_b	2015	2,98	1,37	2,20	7,89	1,77	4,51	10,61	3,77	7,06	**6,84**	2,99	6,71	690
5_a	2972	5,69	6,50	6,06	9,21	5,67	7,25	10,20	6,42	8,24	**7,68**	**6,96**	7,65	255
5_b	5504	4,29	2,20	3,33	13,16	4,61	8,43	13,06	4,15	8,43	**10,80**	**5,17**	10,60	616
6	5098	2.32	0.77	1.55	25.20	0.00	12.60	33.60	0.00	16.80	29.74	**0.70**	28.73	338
6	8029	4.88	2.71	3.80	14.00	5.20	9.60	16.40	6.00	11.20	**14.91**	4.83	14.56	190
6_a	4271	0,97	0,32	0,65	6,40	4,80	5,60	8,00	3,60	5,80	**8,68**	**3,11**	8,48	506
6_b	1951	1,04	0,58	0,81	4,40	5,60	5,00	6,80	6,00	6,40	**4,62**	**5,32**	4,65	609
6_c	430	2,66	4,28	3,47	3,20	10,80	7,00	6,00	10,00	8,00	**4,70**	**8,61**	4,83	300
6_d	2561	0,84	0,58	0,71	7,20	2,40	4,80	4,40	2,80	3,60	**6,16**	**3,17**	6,05	522
6_d	9442	1,88	2,73	2,30	5,60	3,20	4,40	5,20	5,20	5,20	**4,98**	**4,99**	4,98	357
6_d objectives restricted	1689	1,17	1,43	1,30	4,80	2,80	3,80	5,20	4,80	5,00	5,44	**3,78**	5,38	560
6_d objectives restricted	937	1,10	1,43	1,27	3,60	4,40	4,00	5,60	6,00	5,80	**5,10**	4,81	5,09	500
7	1371	0.80	2.70	1.25	2.43	2.58	2.51	3.27	3.24	3.26	**2.96**	**2.88**	2.96	702
7_b	3726	0.60	1.79	0.87	2.27	2.90	2.58	2.58	2.37	2.48	**2.40**	**2.34**	2.40	870

5.2.6 Ensemble of models in preferable set of $scenario\ 7_b$

Having reached a model (model 3726 shown in Table 5.35) with acceptable rates of specificity 97.60% (i.e., 2.40 % FP) and sensitivity 97.66% (i.e., 2.34% FN) at pixel level, an ensemble of the preferable models obtained in $scenario\ 7_b$ was used as a classifier. Each data sample is fed to all 69 preferable models and then a majority vote determines whether the pixel is considered normal or abnormal. In other words, if 35 (i.e., $\left\lfloor \frac{69}{2} \right\rfloor + 1$) preferable models agree that the pixel is abnormal, the pixel will be considered as abnormal; Otherwise it will be considered as a normal one. Table 5.37 shows the results obtained on $MOGA\_DS$ and BIG_DS. Comparing the results with the ones obtained from model 3726, shown in Table 5.35, one can see that there are 0.41% and 0.56% reductions in the FP and FN rates over BIG_DS, respectively. Hence, the ensemble achieves a specificity of 98.01% (i.e., 1.99 % FP) and a sensitivity of 98.22% (i.e., 1.78% FN) at pixel level over BIG_DS.

Table 5.37 The result of applying ensemble of preferable models of $scenario\ 7_b$ on $MOGA\_DS$ and BIG_DS

Ensemble of preferable models in $scenario\ 7_b$	TR_{MOGA_DS}			TE_{MOGA_DS}			V_{MOGA_DS}			BIG_DS		
	FP (%)	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***
	0.44	1.18	0.61	1.90	2.03	1.96	1.93	1.74	1.83	**1.99**	**1.78**	**1.99**

5.3 Comparing results with support vector machine

In order to compare the obtained results with a Support Vector Machine, the MATLAB SVM tool with Gaussian Radial Basis Function kernel was used. For this experiment, we used the dataset of scenario 7 (i.e., from which our best MOGA model was obtained). For determining the best penalty parameter C (i.e., recall from section 2.2 that C is the penalty parameter to control the sensitivity of SVM to possible outliers) and the best spread σ for RBFs, 121 possible combinations obtained by selecting 2 values from the set {0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300} were considered for SVM training and the combination (C=3, spread=1) whose error on test set was

minimum, was selected. In this experiment, 69.8% of data samples in training set were considered as support vectors. Table 5.38 shows the FP and FN rates when this SVM was applied.

Table 5.38 FP and FN rates using SVM

$TR_{DS(7_b)}$			$TE_{DS(7_b)}$			$V_{DS(7_b)}$			BIG_DS		
FP (%)	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*	*FP (%)*	*FN (%)*	*FD (%)*
0.16	0	0.13	2.6	2.42	2.51	2.32	2.26	2.29	**2.5**	**2.37**	2.5

Comparing the results with the ones obtained with the ensemble, shown in Table 5.37, and also with model 3726, shown in Table 5.35, one can see that even with a huge complexity of the SVM model (13,960 support vectors), its FP and FN rates in BIG_{DS} are not only higher than that of ensemble of preferable models but also than model 3726. Notice that a SVM model, with Gaussian Kernel can be considered a RBF model, where the centers of the Gaussians are the support vectors, and with a common spread to all the neurons. In this case, all the features (51) were considered as inputs and 13960 support vectors were employed. This is translated into a complexity of 711,960 parameters, determined by the SVM algorithm, compared with a complexity of 870 (around 0.1%), for model 3726 in Table 5.35.

5.4 Comparing the results with other works

The authors in [7] presented a Computer Aided Detection method for early detection of CVAs from CT images where, in the same way as this work, in a preprocessing phase artifacts are removed and tilted CT images are realigned. In order to find the regions that have higher probability of being a lesion, a Circular Adaptive Region Of Interest (CAROI) algorithm is applied on each CT slice, which aims to draw a circular border around areas with sudden change of intensity values. Each circular region is then compared with its corresponding region in the other side of the brain using the Pearson Correlation Coefficient (PCC). Those circular areas which have the smallest PCC values are selected for further investigation. Eight second order features are calculated from the GLCM matrix of previously selected circular regions and are passed to a 3-layer feed-forward back propagation neural network which was trained using 10 normal and 20 abnormal cases in a round robin (leave-one-out) manner. The output of the neural network identifies whether the circular

region is a lesion or not. In order to evaluate their CAD system, 31 positive cases containing 82 ischemic strokes (i.e., 39 acute and 43 chronic) were used as validation set. A sensitivity of 76.92% (i.e., 30/39 lesion areas correctly detected) for acute ischemic strokes and a sensitivity of 90.70% (i.e., 39/43 lesion areas correctly detected) for chronic strokes were reported. This gives a total sensitivity of 84.14% (i.e., $\frac{30+39}{82} \times 100$).

The detection in our work is done at pixel level (i.e., rather than drawing circular areas as lesions) which provides the possibility of specifying the actual contour of the lesion. In spite of the differences of both approaches, in order to be able to compare the accuracy of our work with the one presented in [7] in terms of lesions sensitivity, this measure has been calculated. A total number of 35 ischemic lesions within 150 CT images were marked by our collaborating neuroradiologist. The ensemble of preferable models in $scenario\ 7_b$ detected 30 lesions correctly, which is translated in a sensitivity of 85.71%.

The authors in [8] developed a CAD system for detecting hemorrhagic strokes in CT images. After removing the artifacts and realigning the tilted images, the hemorrhagic areas are segmented by employing a threshold on the pixels' intensity values. To detect the edema regions, a higher contrast ratio of a given CT image is firstly improved using a local histogram equalization. A thresholding method is then applied to segment the edema region from the normal tissue. The accuracy of the CAD system is evaluated by comparing the Area of Bleeding Region (ABR) and Area of Edema Region (AER) that are detected by the CAD and the ones that are marked by the doctor using data from 8 spontaneous hemorrhagic stroke patients. It is reported that the average difference of ABRs is 8.8%, and the average of the degree of coincidence is 86.4%, while the average difference of AERs is 14.1%, and the average of degree of coincidence is 77.4%.

The results obtained by the last approach cannot be exactly compared with the approach presented here, as [8] deals with hemorrhagic strokes, which typically are much easier to detect and mark than ischemic strokes. In spite of that, the average difference of the areas as well as the average degree of coincidence have been computed for the cases presented here, for the lesions both marked by the doctor and detected by our system. The average difference is 11.4%, and the average degree of coincidence is 88.6%. These figures are better than the values obtained for AER, in approach [8].

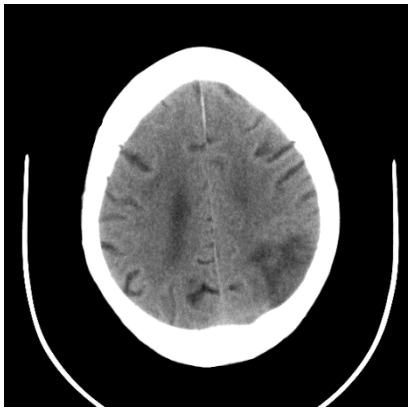
5.5 Visualizing abnormal regions in CT images using ensemble of preferable models obtained by MOGA in $scenario\ 7_b$

CT images from the patients' sample were submitted to the ensemble of preferable models of $scenario\ 7_b$. In order to reduce the computational load when submitting images to the neural network, the following procedure was carried out. Instead of feeding all the pixels to the network, a grid is considered on the intracranial area of each CT slice. Only those pixels which reside on the intersection points of the grid were submitted to the classifier (i.e., an intersection point is any point on the grid which is formed by the intersection of one horizontal and one vertical line). When one of these was reported as corresponding to pathology, all its neighbors were also submitted to the network. The process is repeated in a recursive way until no neighbor pixels are reported as pathological. In the implementation, the distance between each two adjacent vertical lines and each two adjacent horizontal lines of the grid were set to 15 pixels.

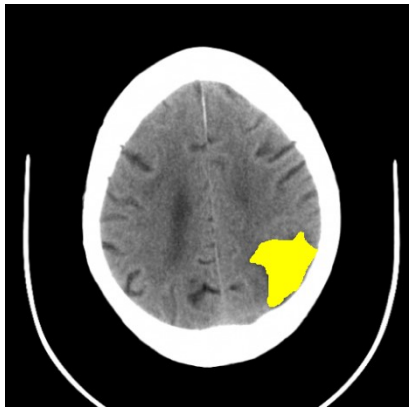
Fig. 5.2 shows the results of applying ensemble of preferable models of $scenario\ 7_b$on 11 CT images, where the output images of the classifier were marked with different colors, depending on the percentage of preferable models with a positive output for each tested pixel. The color code is shown in Table 5.39 The first 9 CT images shown in Fig. 5.2 have some lesions while the remaining are completely normal. The left column of Fig. 5.2 shows the original images. In the middle column the lesions marked by the Neuroradialogist are shown. In the right column the pixels are marked by the classifier.

Table 5.39. Colour code used for marking pixels based on the percentage of preferable models with a positive output

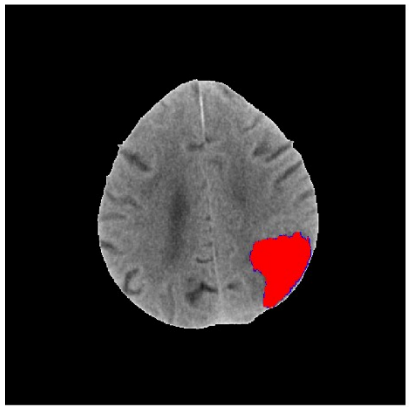
Percentage of preferable models with a positive output	Colour code	Description
[66% 100%]	Red	Clear presence of pathology
[50% 66%)	Blue	Cannot decide whether the pixel is normal or abnormal
[0% 50%)	---	Clear absence of pathology



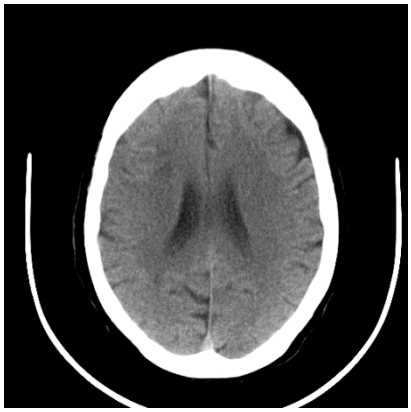
(1)



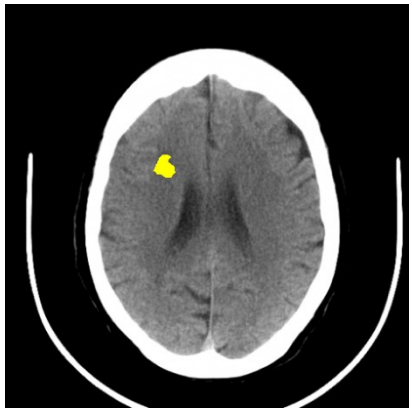
(2)



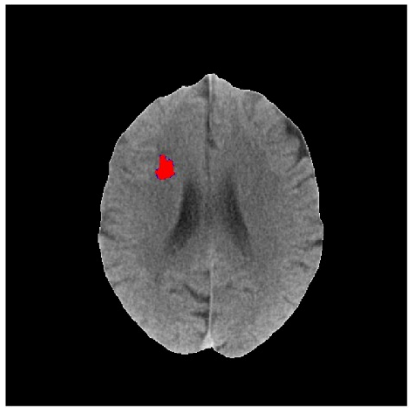
(3)



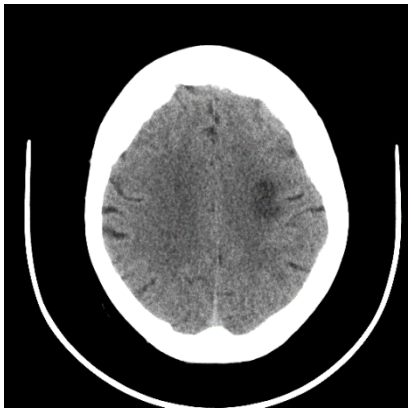
(4)



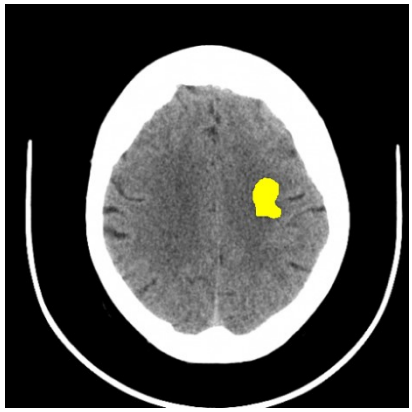
(5)



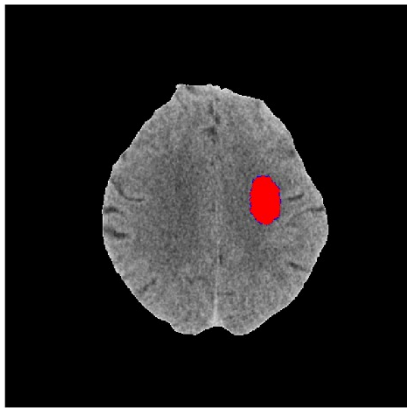
(6)



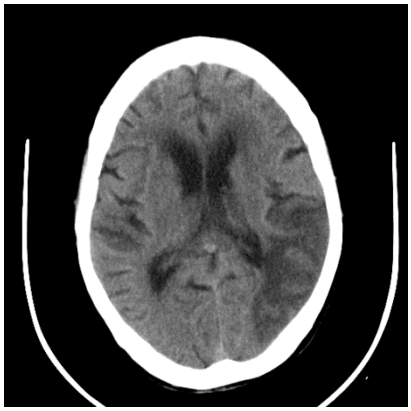
(7)



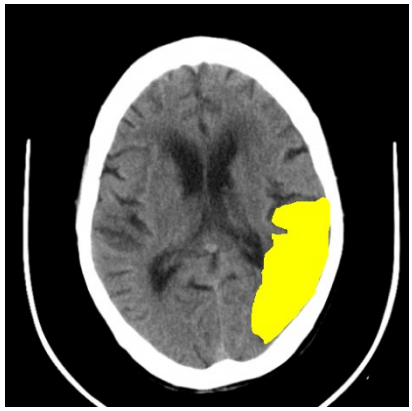
(8)



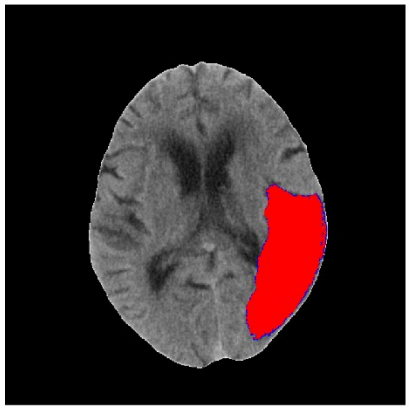
(9)



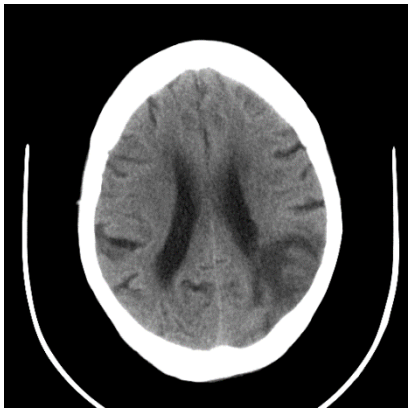
(10)



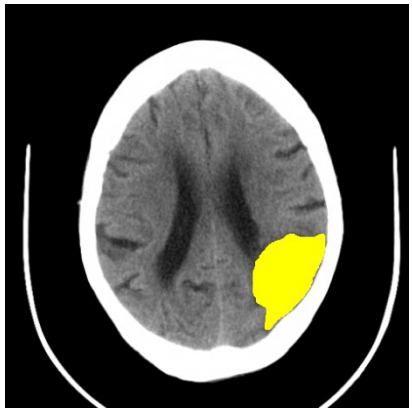
(11)



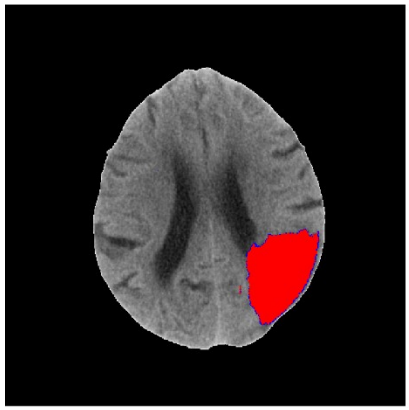
(12)



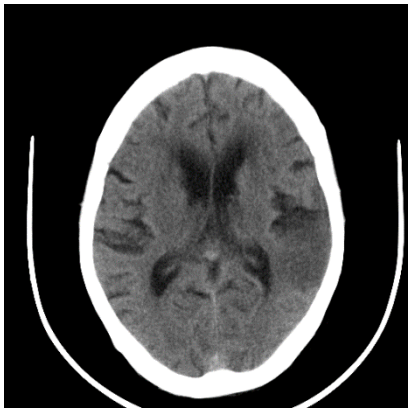
(13)



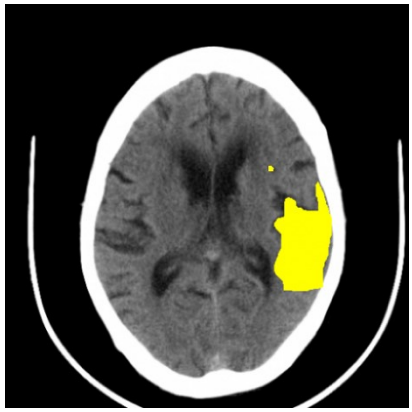
(14)



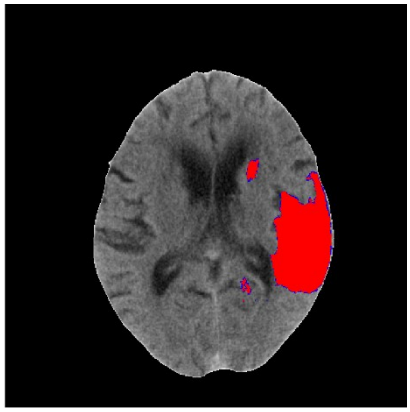
(15)



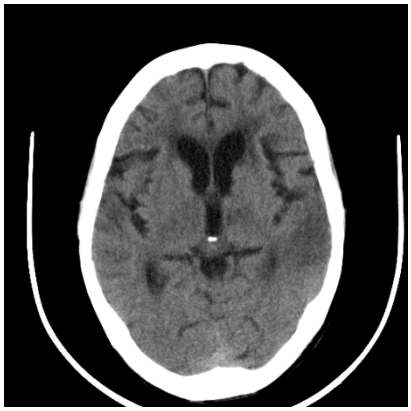
(16)



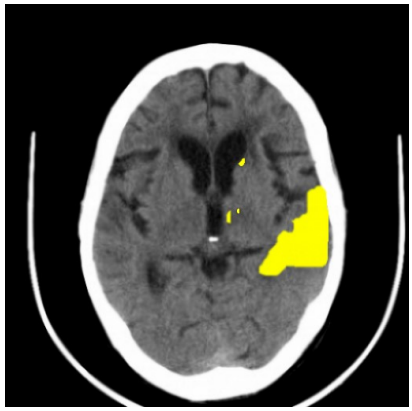
(17)



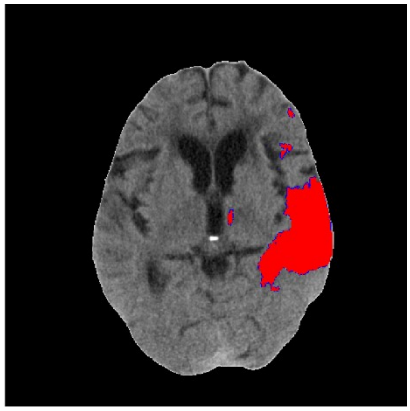
(18)



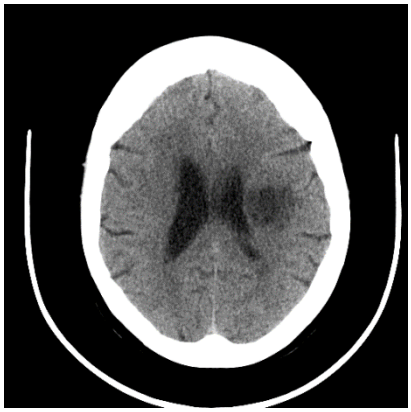
(19)



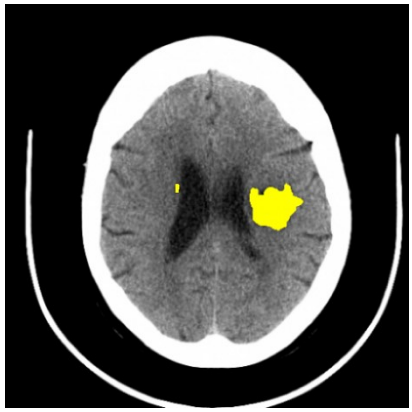
(20)



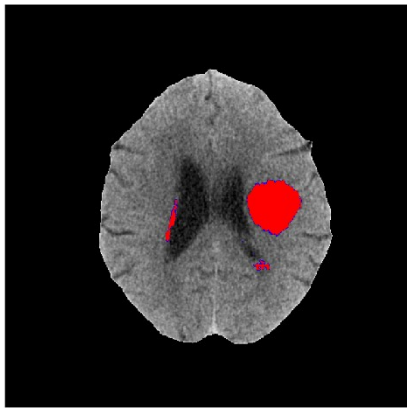
(21)



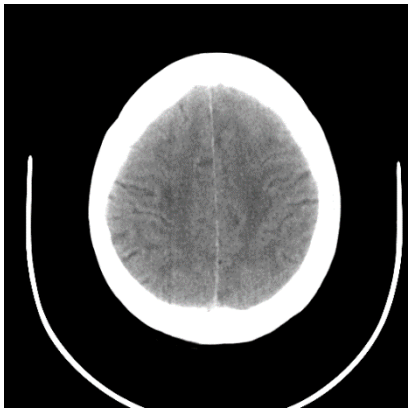
(22)



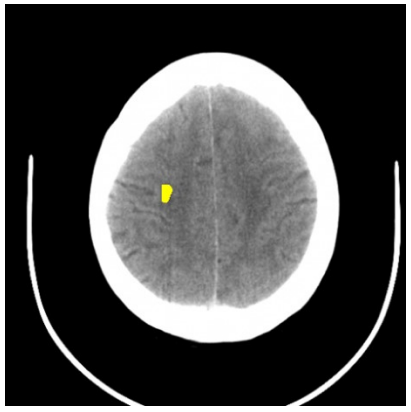
(23)



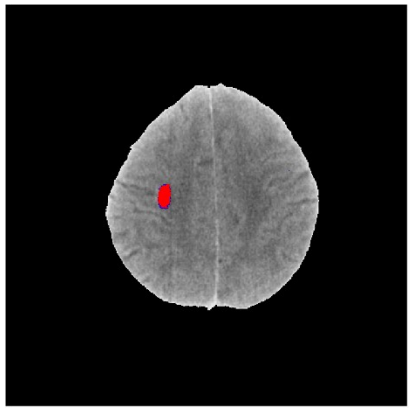
(24)



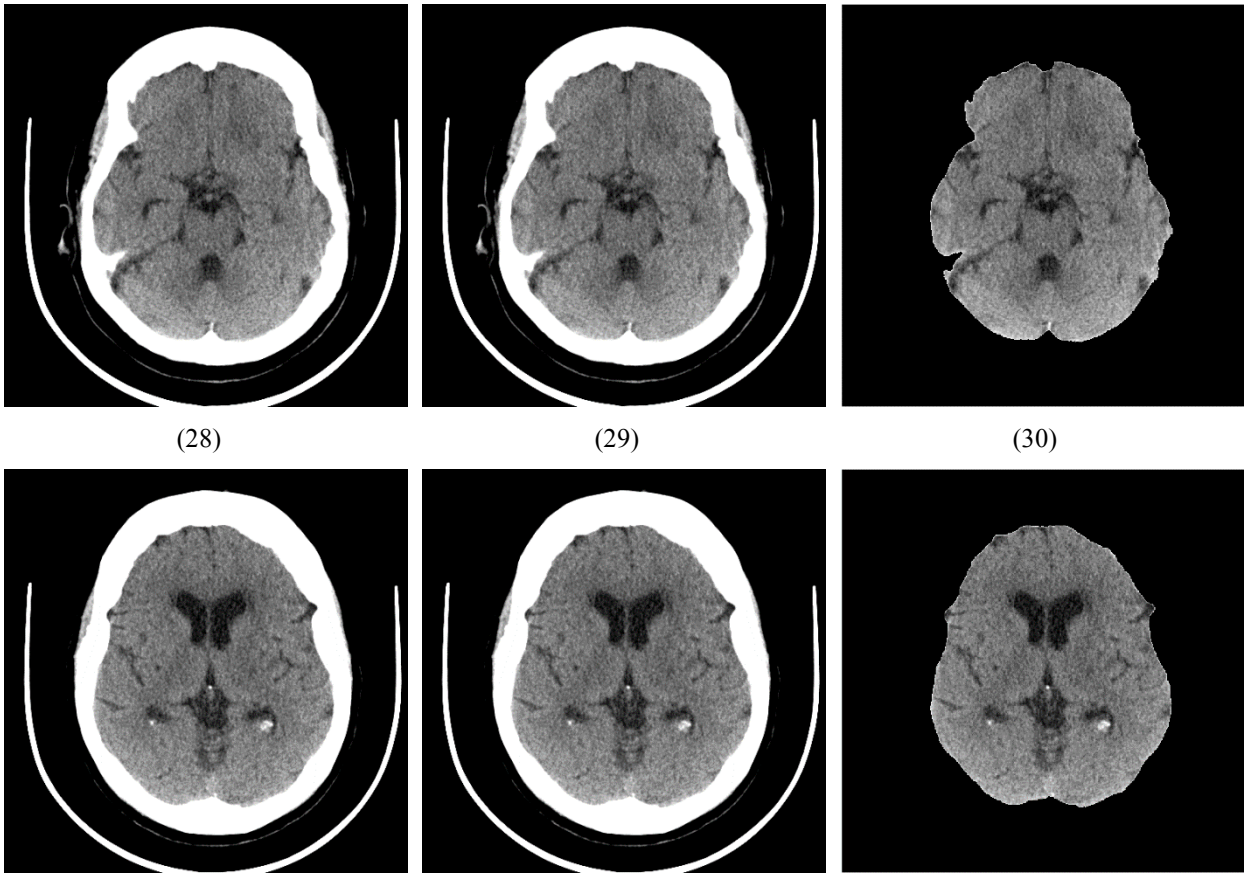
(25)



(26)



(27)



(28) (29) (30)

(31) (32) (33)

Fig. 5.2 The result of applying the ensemble of preferable models of $scenario\ 7_b$ on 11 CT images. The left column shows the original images. In the middle column the lesions marked by the Neuroradialogist are shown. In the right column the pixels are marked by the classifier.

As it can be seen in Fig. 5.2, the classifier is able to detect the great majority of the lesions, but sometimes will identify small false lesions (e.g., Figs. 5.2-21 and 5.2-24). This could be due to imbalance nature of the existing dataset (i.e., the number of abnormal pixels is much smaller than the number of normal pixels).

5.6 Discussion on the discrimination power of the most frequent features in the preferable models of the best scenario

To understand which features are the most frequent ones in preferable models of $scenario\ 7_b$, the normalized frequency of each feature fi within 69 preferable models is calculated by eq. (5.1) and their corresponding histogram is presented in Fig. 5.3.

$$normalized\ frequency(fi) = \frac{frequency(fi)}{n} * 100 \quad (5.1)$$

Where n is the number of preferable models in $scenario\ 7_b$ and $frequency(fi)$ indicates the number of preferable models that used feature fi as their input. From Fig. 5.3, one can see that, among the allowable 30 features within the 51 features considered, features $\{f2, f4, f5, f7, f12, f33, f41, f42, f44, f45\}$ are the ones that have been repeated in more than 80% of preferable models. Among this set, features $\{f2, f4, f5, f7, f12, f41\}$ are from the set of first order statistics, feature $f33$ is from the set of second order statistics and features $\{f42, f44, f45\}$ are from the symmetry features.

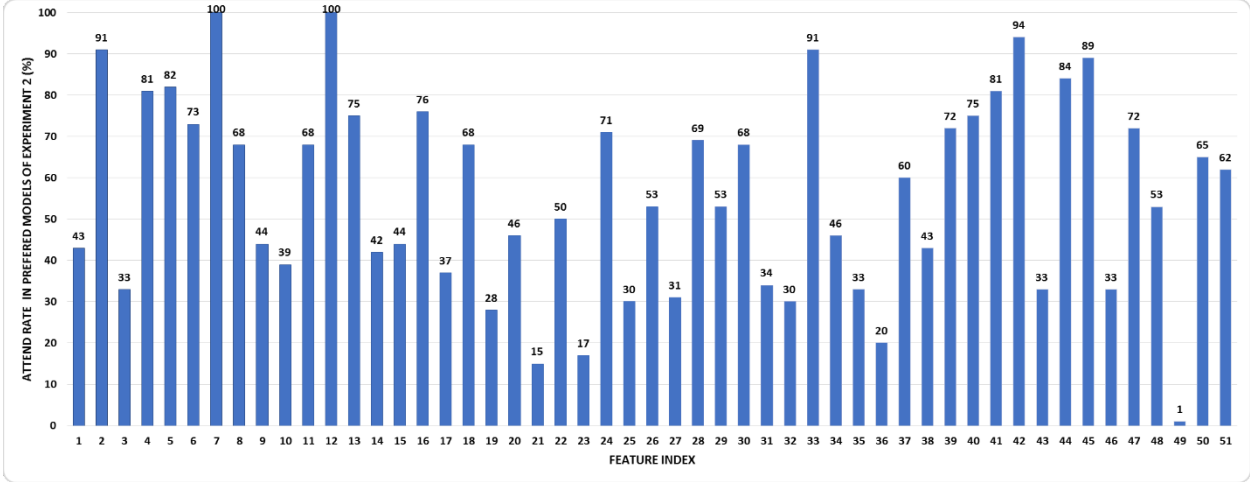


Fig. 5.3 Normalized frequency of each feature in preferable models of $scenario\ 7_b$.

In order to see the discrimination capability of the most frequent features in preferable set of $\boldsymbol{scenario\ 7_b}$, we investigated whether these features will also have a high rank value while using the feature selection method proposed in [121]. This method is based on the Amplitude Distribution Histograms (ADHs) of the data samples of each class (e.g., normal and abnormal data samples) and gives a higher rank value to the features that have greater difference between normalized ADHs of the data samples of each class. To achieve the normalized ADHs of each class, $\boldsymbol{ADH_{norm}}$, the original histograms were divided by the number of samples in each class, as in eq. (5.2).

$$ADH_{normj} = \frac{ADH_j}{ns \times w} \quad (5.2)$$

where $\boldsymbol{ADH_j}$ is the original histogram of class $\boldsymbol{j}$, $\boldsymbol{ns}$ is the number of data samples of class $\boldsymbol{j}$, and $\boldsymbol{w}$ is the bin-width. The features were normalized to fall inside the interval [0 1], and the feature axis was discretized into 100 equally spaced bins ($\boldsymbol{n = 100, w = 0.01}$), as required for calculating ADHs. The net area under each normalized ADH is therefore one. The common area between two normalized ADHs of a two-class problem, $\boldsymbol{C_{ADHs}}$, can thus be calculated as eq. (5.3).

$$C_{ADHs} = w \times \sum_{i=1}^{n} \min(ADH_{norm1}, ADH_{norm2}) \tag{5.3}$$

where $\boldsymbol{n}$ is the number of bins, and $\boldsymbol{i}$ indexes the bins where the values of two classes are distributed. The $\boldsymbol{C_{ADHs}}$ has a value in the real interval [0 1]. The difference of normalized ADHs for a two-class problem is defined by eq. (5.4).

$$D_{ADHs} = 1 - C_{ADHs} \tag{5.4}$$

Higher $\boldsymbol{D_{ADHs}}$ values represent higher separability between samples of different classes, for a given feature. So, features with high $\boldsymbol{D_{ADHs}}$are more likely to improve prediction performance [121]. Fig. 5.4 shows $\boldsymbol{D_{ADHs}}$ values for 51 features in our feature space.

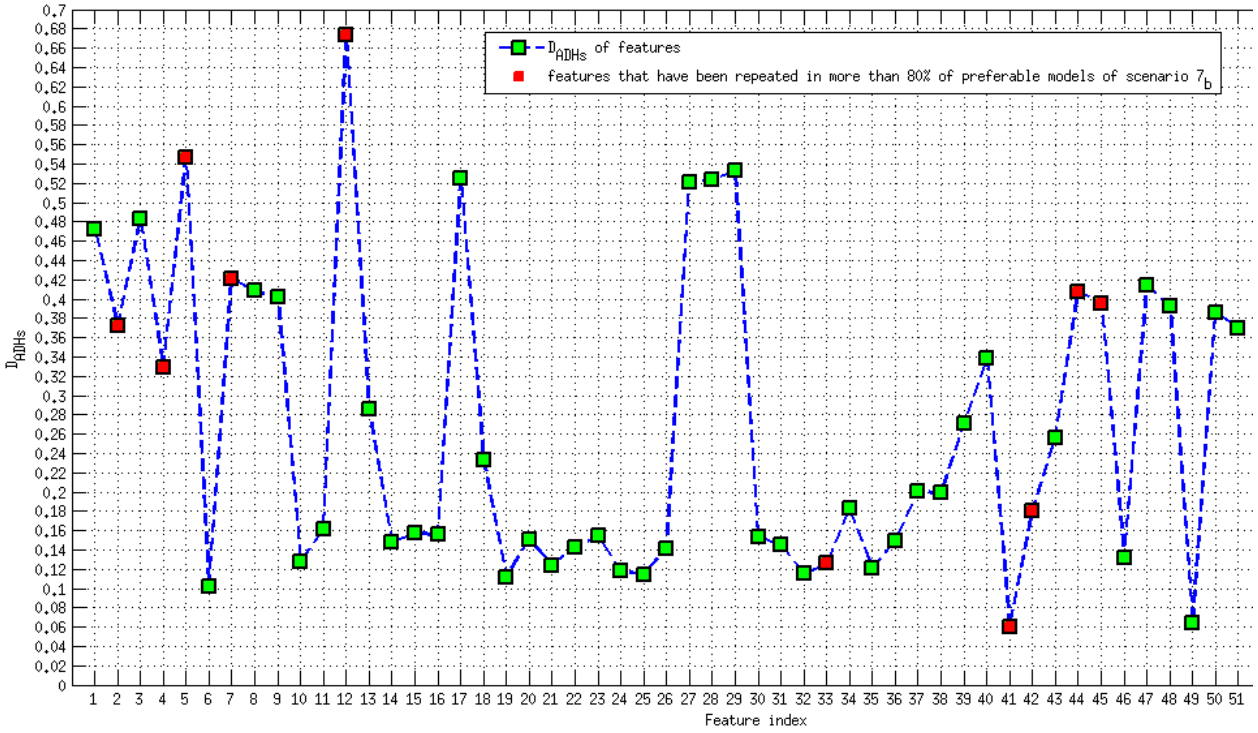


Fig. 5.4 $\boldsymbol{D_{ADHs}}$ values for 51 features in our feature space.

As it is stated in Fig. 5.4, red points are features $\{f2, f4, f5, f7, f12, f33, f41, f42, f44, f45\}$ which have been repeated in more than 80% of preferable models of $scenario\ 7_b$. Comparing Figures 5.3 and 5.4, one can see that within the most frequent features of the preferable models of $scenario\ 7_b$, we have features with high and low D_{ADHs} (e.g., feature $f12$ which appeared in all preferable models of $scenario\ 7_b$ with D_{ADHs} value near to 0.68 and feature $f41$ which appeared in 94% of preferable models of $scenario\ 7_b$ with D_{ADHs} value near to 0.06). This reveals the fact that although some features have a low discrimination power when considering them alone, their linear combination with other features, after mapping them all to a new feature space using the Radial Basis Functions, could provide a high discrimination power between normal and abnormal pixels.

6. Final comments and future work

6.1 Conclusions

Haemorrhagic stroke or blood vessel blockage due to blood clots (ischemic stroke or transient ischemic attack if clots are of short dimensions) are the main cause of Cerebral Vascular Accidents. Occurrence of these pathologies are still responsible for a large index of mortality and morbidity among developed and developing countries. Typically, CVA diagnosis is performed through Computer Tomography images, where each examination is composed of several brain slice images. Prompt diagnosis is not always available due to lack of full-time specialized clinicians, or even, at early stages of CVA, subtle changes in CT image's tones may not be perceived by the human eye. Therefore the existence of a computational intelligent application capable of assisting the neuroradiologist in the analysis of CT scan images, would greatly improve triggering the pathologic occurrence. In this thesis, an RBFNN based diagnosis system for automatic identification of CVAs through analysis of brain CT images was presented.

In chapter 2 the basic concepts of data driven modeling techniques that are used for developing the proposed intelligent support system were reviewed.

For detecting CVA abnormalities from head CT slices we have to focus on the intracranial part of the images. Other parts including the scalp, the skull and the U-shaped head holder are considered as artefacts and should be removed. Moreover, those slices which have been taken from the lower part of the head have too much noise from other organs like the eyes and the nose and contain a very small portion of intracranial area, which means that this kind of slices are not suitable for CVA detection. Another issue that must be considered lies with the problem of tilted head position in some CT images, which can happen due to patient movement during imaging process or as a part of the clinical process. Additionally, in order to extract symmetry features we need to detect the actual midline of the brain and rotate the tilted images to make the actual midsagittal line perpendicular to the x-axis. Chapter 3 addresses these problems and the corresponding applied solutions. A thorough review on the features was also been done in this chapter.

To train, test and validate the neural network models for classifying pathologic areas within brain CT images, it was decided to acquire the opinion of Neuroradiologists, and use it as the gold

standard. In order to remotely collect this information in an accurate and convenient way, a web-based tool was developed whose functionalities are described in chapter 4.

Having a set of CT images at hand whose lesion areas have been already identified by doctor, after removing their artefacts and realigning the tilted ones, we were able to construct our dataset by extracting features from normal and abnormal pixels. The process is explained in chapter 5. Moreover, in order to understand to what extent each feature alone can discriminate between normal and abnormal pixels and to detect and illustrate the location and variation changes between different groups of data, the bi-histogram and box plot of each feature for normal and abnormal groups of pixels are plotted.

Employing a set of 51 features composed of first order, second order and symmetry features, the MOGA design framework is then employed to find the best possible RBFNN structure and its corresponding parameters. Several experiments were conducted in MOGA which are explained in chapter 6. The best result is obtained from an ensemble of preferable models of $scenario\ 7_b$, where the FN_{TR} and FP_{TR} objectives were restricted based on the results obtained by the best model from $scenario\ 7$. Values of specificity of 98.01% (i.e., 1.99 % FP) and sensitivity of 98.22% (i.e., 1.78% FN) were obtained at pixel level, in a set of 150 CT slices (1,867,602 pixels).

Comparing the classification results with SVM over BIG_DS, we were able to conclude that, despite the huge complexity of the SVM model, the accuracy of the selected model in $scenario\ 7_b$, as well as of the ensemble of preferable models is superior to that of the SVM model. The present approach compares also favorably with other similar (although with not the same specifications) published approaches, achieving, on the one hand, improved sensitivity at lesion level, and, on the other hand, superior average difference and degree of coincidence between lesions marked by the doctor and marked by the automatic system.

As the number of abnormal pixels is much smaller than the number of normal pixels in the existing dataset, the classifier is able to detect the great majority of the lesions, but sometimes will identify false lesions.

6.2 Future work

A developed system may always be improved by adding new capabilities or trying to increase the accuracy of its functions. Below, we have identified two future research directions that helps to enhance the accuracy of the already available system.

6.2.1 Adding region specific classifiers to reduce the number of false positives

As the number of abnormal pixels is much smaller than the number of normal pixels in the existing dataset, at the present stage the classifier is able to detect the great majority of the lesions, but sometimes will identify false lesions. One possibility to improve these results, which was not used in this work, is using a general classifier, working in the whole brain (as employed in the current approach, followed by specific classifiers, operating in specific part of the brain. The general classifier would be designed with the aim of not missing existing lesions (therefore giving preference to the minimization of false negatives over the minimization of false positives) while the specific classifiers, would be designed to give preference to the minimization of false positives over the minimization of false negatives, therefore aiming to not to produce false lesions in the areas where real lesions have already been identified. This can be obtained by assigning different priorities, in MOGA, to the minimization of false positives and negatives, or restricting one objective and minimizing the other.

6.2.2 Using online adaptation techniques to improve the classifier as new unseen data arrives

The classifier is designed off-line, with structure and parameters kept fixed afterwards. But in spite of the excellent performance obtained, classification errors might increase when applied to other CT images of different patients. One way to tackle this problem is to adapt on-line the classifier, whenever classification errors are found. This can be done, for example, by adapting the network parameters in the case where the input data lies outside the current convex-hull.

References

[1] A. Ropper, M. Samuels, and J. Klein, Adams and Victor's Principles of Neurology 10th Edition. McGraw-Hill Education, 2014.

[2] P. M. Ferreira and A. E. Ruano, "Evolutionary Multiobjective Neural Network Models Identification: Evolving Task-Optimised Models," (in English), New Advances in Intelligent Signal Processing, Proceedings Paper vol. 372, pp. 21-53, 2011.

[3] E. Hajimani, C. A. Ruano, M. G. Ruano, and A. E. Ruano, "A software tool for intelligent CVA diagnosis by cerebral computerized tomography," in 8th IEEE International Symposium on Intelligent Signal Processing (WISP), 2013, pp. 103-108.

[4] E. Hajimani, A. Ruano, and G. Ruano, "The Effect of Symmetry Features on Cerebral Vascular Accident Detection Accuracy," presented at the RecPad 2015, the 21th edition of the Portuguese Conference on Pattern Recognition, Faro, Portugal, 2015.

[5] H. Khosravani, A. Ruano, and P. Ferreira, "A Convex Hull-based Data Selection Method for Data Driven Models," Applied Soft Computing, 2016 (in press).

[6] A. Ruano, H. R. Khosravani, and P. M. Ferreira, "A Randomized Approximation Convex Hull Algorithm for High Dimensions," IFAC-PapersOnLine, vol. 48, no. 10, pp. 123 - 128, 2015.

[7] K.-s. D. Ng, "Computer aided detection method for early detection of cerebrovascular accident," PhD, Dept. of Health Technology and Informatics, The Hong Kong Polytechnic University, The Hong Kong Polytechnic University, 2009.

[8] J.-G. Gan, Y.-W. Wang, J.-L. Su, L. Chan, and Ieee, "The Development of CAD system for Hemorrhagic Stroke in Computed Tomography Images," 2014 Ieee International Symposium on Bioelectronics and Bioinformatics (Isbb), 2014 2014.

[9] C. Metin and D. O. Frost, "Visual Responses of Neurons in Somatosensory Cortex of Hamsters with Experimentally Induced Retinal Projections to Somatosensory Thalamus," Proceedings of the National Academy of Sciences of the United States of America, vol. 86, no. 1, pp. 357-361, Jan 1989.

[10] A. W. Roe, S. L. Pallas, Y. H. Kwon, and M. Sur, "Visual Projections Routed to the Auditory Pathway in Ferrets - Receptive-Fields of Visual Neurons in Primary Auditory-Cortex," Journal of Neuroscience, vol. 12, no. 9, pp. 3651-3664, Sep 1992.

[11] A. E. Ruano, "Artificial Neural Networks," ed. Faro, Portugal: Centre for Intelligent Systems, University of Algarve, p. 295.

[12] S. Haykin, Neural Networks: A Comprehensive Foundation. Prentice Hall PTR, 1998, p. 842.

[13] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron) - A review of applications in the atmospheric sciences," Atmospheric Environment, vol. 32, no. 14-15, pp. 2627-2636, Aug 1998.

[14] R. J. Kuo, T.-L. Hu, and Z.-Y. Chen, "Sales Forecasting Using an Evolutionary Algorithm Based Radial Basis Function Neural Network," Information Systems: Modeling, Development, and Integration: Third International United Information Systems Conference, Uniscon 2009, vol. 20, pp. 65-74, 2009 2009.

[15] J. Wang, Advances in Neural Networks - ISNN 2006: Pt. III: Third International Symposium on Neural Networks, ISNN 2006, Chengdu, China, May 28 - June 1, 2006, Proceedings. Springer Berlin Heidelberg, 2006.

[16] O. Kaynak, Artificial Neural Networks and Neural Information Processing - Icann/Iconip 2003: Joint International Conference Icann/Icinip 2003, Istanbul, Turkey, June 26-29, 2003, Proceedings. Springer Berlin Heidelberg, 2003.
[17] K. L. Du and N. S. Swamy, Neural Networks and Statistical Learning. Springer London, 2013.
[18] W. Sun and Y. X. Yuan, Optimization Theory and Methods: Nonlinear Programming (Springer Optimization and Its Applications). Springer US, 2006.
[19] D. Tsegay and A. Mebrahtu, "Multidimensional and Multi-Parameter Fortran-Based Curve FittingTools," MEJS, vol. 1, no. 1, p. 20, 2009.
[20] W. W. Hsieh, Machine Learning Methods in the Environmental Sciences: Neural Networks and Kernels. Cambridge University Press, 2009.
[21] A. E. Ruano and I. o. E. Engineers, Intelligent Control Systems Using Computational Intelligence Techniques. Institution of Engineering and Technology, 2005.
[22] A. E. Ruano, D. Jones, and P. J. Fleming, "A new formulation of the learning problem for a neural network controller," in 30th IEEE conference on Decision and control, Brighton, England, 1991, vol. 1, pp. 865-6.
[23] Y. Wang, M. Chang, H. Chen, and M. Q. Wang, "Application of RBF Neural Network in Intelligent Fault Diagnosis System," Proceedings of International Conference on Soft Computing Techniques and Engineering Application, Icsctea 2013, vol. 250, pp. 561-566, 2014 2014.
[24] K. B. Kim and C. K. Kim, "Performance improvement of RBF network using ART2 algorithm and fuzzy logic system," Ai 2004: Advances in Artificial Intelligence, Proceedings, vol. 3339, pp. 853-860, 2004 2004.
[25] S. Papadimitriou, S. Mavroudi, L. Vladutu, and A. Bezerianos, "Generalized radial basis function networks trained with instance based learning for data mining of symbolic data," Applied Intelligence, vol. 16, no. 3, pp. 223-234, May-Jun 2002.
[26] C. Cortes and V. Vapnik, "Support-Vector Networks," Machine Learning, vol. 20, no. 3, pp. 273-297, Sep 1995.
[27] A. E. Ruano, G. Madureira, O. Barros, H. R. Khosravani, M. G. Ruano, and P. M. Ferreira, "Seismic detection using support vector machines," Neurocomputing, vol. 135, pp. 273-283, Jul 5 2014.
[28] T.-T. Frie, #223, N. Cristianini, and C. Campbell, "The Kernel-Adatron Algorithm: A Fast and Simple Learning Procedure for Support Vector Machines," presented at the Proceedings of the Fifteenth International Conference on Machine Learning, 1998.
[29] S. S. Haykin, Neural Networks: A Comprehensive Foundation (International edition). Prentice Hall, 1999.
[30] X. F. Zha and R. J. Howlett, Integrated Intelligent Systems for Engineering Design (Frontiers in Artificial Intelligence and Applications). IOS Press, 2006.
[31] D. Whitley, "A Genetic Algorithm Tutorial," Statistics and Computing, vol. 4, no. 2, pp. 65-85, Jun 1994.
[32] C. M. M. d. Fonseca, "Multiobjective genetic algorithms with application to control engineering problems," PhD, Department of Automatic Control and Systems Engineering, University of Sheffield, 1995.
[33] M. Safe, J. Carballido, I. Ponzoni, and N. Brignole, "On stopping criteria for genetic algorithms," Advances in Artificial Intelligence - Sbia 2004, vol. 3171, pp. 405-413, 2004 2004.

[34] C. M. Fonseca and P. J. Fleming, "Multiobjective Genetic Algorithms Made Easy: Selection, Sharing and Mating Restriction," presented at the Genetic Algorithms in Engineering Systems: Innovations and Applications, UK, 1995.
[35] C. A. Teixeira, M. G. Ruano, A. E. Ruano, and W. C. A. Pereira, "A soft-computing methodology for noninvasive time-spatial temperature estimation," IEEE Transactions on Biomedical Engineering, vol. 55, no. 2, pp. 572-580, Feb 2008.
[36] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part I: A unified formulation," IEEE Transactions on Systems Man and Cybernetics Part a-Systems and Humans, vol. 28, no. 1, pp. 26-37, Jan 1998.
[37] C. A. D. Teixeira, "Soft-computing techniques applied to artificial tissue temperature estimation," PhD, FCT, University of Algarve, 2008.
[38] A. P. Engelbrecht and R. Brits, "Supervised training using an unsupervised approach to active learning," Neural Processing Letters, vol. 15, no. 3, pp. 247-260, Jun 2002.
[39] M. Sugiyama and H. Ogawa, "Active learning for optimal generalization in trigonometric polynomial models," Ieice Transactions on Fundamentals of Electronics Communications and Computer Sciences, vol. E84A, no. 9, pp. 2319-2329, Sep 2001.
[40] M. Sugiyama and H. Ogawa, "Incremental active learning with bias reduction," in IEEE/INNS/ENNS International Joint Conference on Neural Networks (IJCNN 2000), Como, Italy, 2000, pp. 15-20, 2000.
[41] H. R. Khosravani, A. E. Ruano, and P. M. Ferreira, "A simple algorithm for convex hull determination in high dimensions," presented at the 8th International Symposium on Intelligent Signal Processing (WISP), Portugal, 2013.
[42] V. Ganganwar, "An overview of classification algorithms for imbalanced datasets," International Journal of Emerging Technology and Advanced Engineering, vol. 2, no. 4, p. 6, 2012.
[43] M. Ghanavati, R. K. Wong, F. Chen, Y. Wang, and C.-S. Perng, "An Effective Integrated Method for Learning Big Imbalanced Data," 2014 Ieee International Congress on Big Data (Bigdata Congress), pp. 691-698, 2014 2014.
[44] C. Beyan and R. Fisher, "Classifying imbalanced data sets using similarity based hierarchical decomposition," Pattern Recognition, vol. 48, no. 5, pp. 1653-1672, May 2015.
[45] Z. Yang, D. Gao, and Ieee, "An Active Under-sampling Approach for Imbalanced Data Classification," 2012 Fifth International Symposium on Computational Intelligence and Design (Iscid 2012), Vol 2, pp. 270-273, 2012 2012.
[46] H. Han, W. Y. Wang, and B. H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," Advances in Intelligent Computing, Pt 1, Proceedings, vol. 3644, pp. 878-887, 2005 2005.
[47] H. He and E. A. Garcia, "Learning from Imbalanced Data," Ieee Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263-1284, Sep 2009.
[48] Z. H. Zhou, J. X. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," (in English), Artificial Intelligence, Article; Proceedings Paper vol. 137, no. 1-2, pp. 239-263, May 2002, Art. no. Pii s0004-3702(02)00190-x.
[49] R. Providencia, L. Goncalves, and M. J. Ferreira, "Cerebrovascular mortality in Portugal: Are we overemphasizing hypertension and neglecting atrial fibrillation?," Revista Portuguesa De Cardiologia, vol. 32, no. 11, pp. 905-913, Nov 2013.
[50] A. S. Go et al., "Heart disease and stroke statistics--2013 update: a report from the American Heart Association," (in eng), Circulation, vol. 127, no. 1, pp. e6-e245, Jan 1 2013.

[51] D. Cuete, "Subacute middle cerebral artery infarct," ed: **http://radiopaedia.org/**.
[52] U. Radiology, "Haemorrhagic stroke - basal ganglia ", ed: http://radiopaedia.org/.
[53] (2013). Computed Tomography (CT). Available: http://www.nibib.nih.gov/science-education/science-topics/computed-tomography-ct
[54] A. Berger, "Magnetic resonance imaging," BMJ : British Medical Journal, vol. 324, no. 7328, pp. 35-35, 2002.
[55] A. H. Hielscher et al., "Near-infrared diffuse optical tomography," Disease Markers, vol. 18, no. 5-6, pp. 313-337, 2002 2002.
[56] J. Tian, "Diffuse Optical Tomography," in Molecular Imaging(Advanced Topics in Science and Technology in China: Springer-Verlag Berlin Heidelberg, 2013, p. 699.
[57] A. Berger, "Positron emission tomography," BMJ : British Medical Journal, vol. 326, no. 7404, pp. 1449-1449, 2003.
[58] B. R. Bendok and A. M. Naidech, Hemorrhagic and Ischemic Stroke: Medical, Imaging, Surgical and Interventional Approaches. Thieme, 2011.
[59] W. J. Powers and A. R. Zazulia, "The use of positron emission tomography in cerebrovascular disease," Neuroimaging Clinics of North America, vol. 13, no. 4, pp. 741-+, Nov 2003.
[60] M. Larobina and L. Murino, "Medical Image File Formats," Journal of Digital Imaging, vol. 27, no. 2, pp. 200-206, 12/13 2014.
[61] R. A. Poldrack, J. A. Mumford, and T. E. Nichols, Handbook of Functional MRI Data Analysis. Cambridge University Press, 2011.
[62] R. E. Hales, S. C. Yudofsky, L. W. Roberts, and D. J. Kupfer, The American Psychiatric Publishing Textbook of Psychiatry. American Psychiatric Publishing, 2014.
[63] X. Gang and S. Jing, "Automatic extraction of brain Mid-sagittal Line from normal and abnormal CT images," presented at the Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on, Chengdu, 2010. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5564810
[64] C.-C. Liao, F. Xiao, J.-M. Wong, and I. J. Chiang, "Automatic recognition of midline shift on brain CT images," Computers in Biology and Medicine, vol. 40, no. 3, pp. 331-339, Mar 2010.
[65] F.-h. Tang, D. K. S. Ng, and D. H. K. Chow, "An image feature approach for computer-aided detection of ischemic stroke," Computers in Biology and Medicine, vol. 41, no. 7, pp. 529-536, Jul 2011.
[66] R. Liu et al., "Hemorrhage Slices Detection in Brain CT Images," in 19th International Conference on Pattern Recognition (ICPR 2008), Tampa, FL, 2008, pp. 2189-2192, 2008.
[67] X. Qi et al., "Ideal Midline Detection Using Automated Processing of Brain CT Image," Open Journal of Medical Imaging, vol. 3, no. 2, p. 9, 2013.
[68] L. He, Y. Chao, K. Suzuki, and K. Wu, "Fast connected-component labeling," Pattern Recognition, vol. 42, no. 9, pp. 1977-1987, Sep 2009.
[69] J. H. Brown, E. S. Lustrin, M. H. Lev, C. S. Ogilvy, and J. M. Taveras, "Reduction of aneurysm clip artifacts on CT angiograms: A technical note," American Journal of Neuroradiology, vol. 20, no. 4, pp. 694-696, Apr 1999.
[70] X. Qi et al., "Actual Brain Midline Detection using Level Set Segmentation and Window Selection," presented at the The Eighth International Multi-Conference on Computing in the Global Information Technology, Nice, France, 2013. Available: http://www.thinkmind.org/index.php?view=article&articleid=iccgi_2013_6_30_10313

[71] S. K. Weeratunga and C. Kamath, "An investigation of implicit active contours for scientific image segmentation," in Conference on Visual Communications and Image Processing 2004, San Jose, CA, 2004, vol. 5308, pp. 210-221, 2004.
[72] Wenan Chen, K. Najarian, and K. Ward, "Actual Midline Estimation from Brain CT Scan Using Multiple Regions Shape Matching," in Pattern Recognition (ICPR), 2010 20th International Conference on, pp. 2552-2555.
[73] W. Chen, R. Smith, S.-Y. Ji, K. R. Ward, and K. Najarian, "Automated ventricular systems segmentation in brain CT images by combining low-level segmentation and high-level template matching," BMC Medical Informatics and Decision Making, vol. 9, no. 1, p. 14, 2009.
[74] X. Llado et al., "Segmentation of multiple sclerosis lesions in brain MRI: A review of automated approaches," Information Sciences, vol. 186, no. 1, Mar 1 2012.
[75] T. Hachaj and M. R. Ogiela, "CAD system for automatic analysis of CT perfusion maps," Opto-Electronics Review, vol. 19, no. 1, Mar 2011.
[76] L. E. Poh, V. Gupta, A. Johnson, R. Kazmierski, and W. L. Nowinski, "Automatic Segmentation of Ventricular Cerebrospinal Fluid from Ischemic Stroke CT Images," Neuroinformatics, vol. 10, no. 2, Apr 2012.
[77] W. L. Nowinski, G. Qian, K. N. B. Prakash, Q. Hu, and A. Aziz, "Fast Talairach Transformation for magnetic resonance neuroimages," Journal of Computer Assisted Tomography, vol. 30, no. 4, pp. 629-641, Jul-Aug 2006.
[78] N. Otsu, "A Threshold Selection Method From Gray-level Histogram," IEEE Transactions on Systems, Man, and Cybernetics, 1978.
[79] M. Gao and S. Chen, "Fully Automatic Segmentation of Brain Tumour in CT Images," European Journal of Cancer, vol. 47, Sep 2011.
[80] A. P. Nanthagopal and R. S. Rajamony, "Automatic classification of brain computed tomography images using wavelet-based statistical texture features," Journal of Visualization, vol. 15, no. 4, pp. 363-372, Nov 2012.
[81] T. J. Devadas and R. Ganesan, "Analysis of CT Brain images using Radial Basis Function Neural Network," Defence Science Journal, vol. 62, no. 4, Jul 2012.
[82] O. Freifeld, H. Greenspan, and J. Goldberger, "Lesion detection in noisy MR brain images using constrained GMM and active contours," 4th IEEE International Symposium on Biomedical Imaging : Macro to Nano, Vols 1-3, pp. 596-599, 2007.
[83] H. Greenspan, A. Ruf, and J. Goldberger, "Constrained Gaussian mixture model framework for automatic segmentation of MR brain images," IEEE Transactions on Medical Imaging, vol. 25, no. 9, pp. 1233-1245, Sep 2006.
[84] K. N. B. Prakash, S. Zhou, T. C. Morgan, D. F. Hanley, and W. L. Nowinski, "Segmentation and quantification of intra-ventricular/cerebral hemorrhage in CT scans by modified distance regularized level set evolution technique," International Journal of Computer Assisted Radiology and Surgery, vol. 7, no. 5, Sep 2012.
[85] C. Li, C. Xu, C. Gui, and M. D. Fox, "Distance Regularized Level Set Evolution and Its Application to Image Segmentation," Ieee Transactions on Image Processing, vol. 19, no. 12, pp. 3243-3254, Dec 2010.
[86] V. Harati, R. Khayati, and A. Farzan, "Fully automated tumor segmentation based on improved fuzzy connectedness algorithm in brain MR images," Computers in Biology and Medicine, vol. 41, no. 7, Jul 2011.

[87] B. J. Bedell and P. A. Narayana, "Automatic segmentation of gadolinium-enhanced multiple sclerosis lesions," Magnetic Resonance in Medicine, vol. 39, no. 6, pp. 935-940, Jun 1998.
[88] A. O. Boudraa et al., "Automated segmentation of multiple sclerosis lesions in multispectral MR imaging using fuzzy clustering," Computers in Biology and Medicine, vol. 30, no. 1, pp. 23-40, Jan 2000.
[89] D. Mortazavi, A. Z. Kouzani, and H. Soltanian-Zadeh, "Segmentation of multiple sclerosis lesions in MR images: a review," Neuroradiology, vol. 54, no. 4, Apr 2012.
[90] K. Van Leemput, F. Maes, D. Vandermeulen, A. Colchester, and P. Suetens, "Automated segmentation of multiple sclerosis lesions by model outlier detection," Ieee Transactions on Medical Imaging, vol. 20, no. 8, pp. 677-688, Aug 2001.
[91] Y. Wu et al., "Automated segmentation of multiple sclerosis lesion subtypes with multichannel MRI," Neuroimage, vol. 32, no. 3, pp. 1205-1215, Sep 2006.
[92] P. Anbeek, K. L. Vincken, M. J. P. van Osch, R. H. C. Bisschops, and J. van der Grond, "Probabilistic segmentation of white lesions in MR imaging," Neuroimage, vol. 21, no. 3, pp. 1037-1044, Mar 2004.
[93] R. Khayati, M. Vafadust, F. Towhidkhah, and S. M. Nabavi, "Fully automatic segmentation of multiple sclerosis lesions in brain MR FLAIR images using adaptive mixtures method and markov random field model," Computers in Biology and Medicine, vol. 38, no. 3, pp. 379-390, Mar 2008.
[94] E. Hajimani, M. G. Ruano, and A. E. Ruano, "MOGA design for neural networks based system for automatic diagnosis of Cerebral Vascular Accidents," in 9th IEEE International Symposium on Intelligent Signal Processing (WISP), 2015, pp. 1-6.
[95] J. K. Udupa, L. Wei, S. Samarasekera, Y. Miki, M. A. vanBuchem, and R. I. Grossman, "Multiple sclerosis lesion quantification using fuzzy-connectedness principles," Ieee Transactions on Medical Imaging, vol. 16, no. 5, pp. 598-609, Oct 1997.
[96] F. Admiraal-Behloul et al., "Fully automatic segmentation of white matter hyperintensities in MR images of the elderly," Neuroimage, vol. 28, no. 3, pp. 607-617, Nov 15 2005.
[97] G. Stockman and L. G. Shapiro, Computer Vision. Prentice Hall PTR, 2001, p. 608.
[98] G. N. Srinivasan and G. Shobha, "Statistical Texture Analysis," in PROCEEDINGS OF WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY, 2008, vol. 36, pp. 1264-1269.
[99] A. Usinskas, R. A. Dobrovolskis, and B. F. Tomandl, "Ischemic stroke segmentation on CT images using joint features," Informatica, vol. 15, no. 2, pp. 283-290, 2004.
[100] L. Ribeiro, A. E. Ruano, M. G. Ruano, and P. M. Ferreira, "Neural networks assisted diagnosis of ischemic CVA's through CT scan," IEEE International Symposium on Intelligent Signal Processing, Conference Proceedings Book, pp. 223-227, 2007.
[101] F. R. de Siqueira, W. R. Schwartz, and H. Pedrini, "Multi-scale gray level co-occurrence matrices for texture description," Neurocomputing, vol. 120, pp. 336-345, Nov 23 2013.
[102] A. Khademi, S. Krishnan, and Ieee, "Multiresolution analysis and classification of small bowel medical images," in 29th Annual International Conference of the IEEE-Engineering-in-Medicine-and-Biology-Society, Lyon, FRANCE, 2007, pp. 4524-4527, 2007.
[103] A. Khademi, S. Krishnan, and Ieee, "Medical image texture analysis: A case study with small bowel, retinal and mammogram images," in Canadian Conference on Electrical and Computer Engineering, Niagara Falls, CANADA, 2008, pp. 1861-1866, 2008.

[104] A. Khademi, S. Krishnan, and A. Venetsanopoulos, "Shift-Invariant DWT for Medical Image Classification," in Discrete Wavelet Transforms - Theory and Applications, D. J. T. Olkkonen, Ed.: InTech, 2011.
[105] A. Padma and R. Sukanesh, "A Wavelet Based Automatic Segmentation of Brain Tumor in CT Images Using Optimal Statistical Texture Features," International Journal of Image Processing (IJIP), vol. 5, no. 5, p. 11, 2011.
[106] A. PADMA and R. Sukanesh, "Automatic Classification and Segmentation of Brain Tumor in CT Images using Optimal Dominant Gray level Run length Texture Features," International Journal of Advanced Computer Science and Applications(IJACSA), vol. 2, no. 10, p. 7, 2011.
[107] Wei-Li Zhang and Xi-Zhao Wang, "Feature Extraction and Classification for Human Brain CT Images," in International Conference on Machine Learning and Cybernetics, 2007, vol. 2, pp. 1155-1159.
[108] H. Wu et al., "Combination of radiological and gray level co-occurrence matrix textural features used to distinguish solitary pulmonary nodules by computed tomography," (in eng), J Digit Imaging, vol. 26, no. 4, pp. 797-802, Aug 2013.
[109] R. M. Haralick, Shanmuga.K, and I. Dinstein, "Textural Features for Image Classification," IEEE Transactions on Systems Man and Cybernetics, vol. SMC3, no. 6, pp. 610-621, 1973.
[110] A. Kassner, F. Liu, R. E. Thornhill, G. Tomlinson, and D. J. Mikulis, "Prediction of Hemorrhagic Transformation in Acute Ischemic Stroke Using Texture Analysis of Postcontrast T1-Weighted MR Images," Journal of Magnetic Resonance Imaging, vol. 30, no. 5, pp. 933-941, Nov 2009.
[111] A. A. Othman and H. R. Tizhoosh, "Segmentation of Breast Ultrasound Images Using Neural Networks," Engineering Applications of Neural Networks, Pt I, vol. 363, pp. 260-269, 2011 2011.
[112] N. H. Rajini and R. Bhavani, "Computer aided detection of ischemic stroke using segmentation and texture features," Measurement, vol. 46, no. 6, pp. 1865-1874, Jul 2013.
[113] D. A. Clausi, "An analysis of co-occurrence texture statistics as a function of grey level quantization," Canadian Journal of Remote Sensing, vol. 28, no. 1, pp. 45-62, Feb 2002.
[114] E. H. Linfoot, "An informational measure of correlation," Information and Control, vol. 1, no. 1, pp. 85-89, 1957/09/01 1957.
[115] A. Eguizabal et al., "Linear classifier and textural analysis of optical scattering images for tumor classification during breast cancer extraction," in Conference on Biomedical Applications of Light Scattering VII, San Francisco, CA, 2013, vol. 8592, 2013.
[116] P. Blondel, "Segmentation of the Mid-Atlantic Ridge south of the Azores, based on acoustic classification of TOBI data," Geological Society, London, Special Publications, vol. 118, no. 1, pp. 17-28, 1996.
[117] P. A. Puech, L. Boussel, S. Belfkih, L. Lemaitre, P. Douek, and R. Beuscart, "DicomWorks: Software for reviewing DICOM studies and promoting low-cost teleradiology," Journal of Digital Imaging, vol. 20, no. 2, pp. 122-130, Jun 2007.
[118] R. Pradhan, S. Kumar, R. Agarwal, M. P. Pradhan, and M. K. Ghose, "Contour line tracing algorithm for digital topographic maps," International Journal of Image Processing (IJIP), vol. 4, no. 2, p. 8, 2010.
[119] K. Saeed, M. Tabędzki, M. Rybnik, and M. Adamski, "K3M: A universal algorithm for image skeletonization and a review of thinning techniques," International Journal of Applied Mathematics and Computer Science, vol. 20, no. 2, p. 19, 2010.

[120] P. Ferreira, "Application of Computational Intelligence Methods to Greenhouse Environmental Control," PhD, FCT, University of Algarve, 2007.
[121] M. Bandarabadi, "Low-Complexity Measures for Epileptic Seizure Prediction and Early Detection Based on Classification," PhD, Faculty of Sciences and Technology, University of Coimbra, 2014.
[122] N. L. Johnson, S. Kotz, and N. Balakrishnan, Continuous univariate distributions (Wiley series in probability and mathematical statistics: Applied probability and statistics). Wiley & Sons, 1995.
[123] C. Forbes, M. Evans, N. Hastings, and B. Peacock, Statistical Distributions. Wiley, 2011.
[124] Y. Li, Y. Zhao, S. Peng, Q. Zhou, and L. Q. Ma, "Temporal and spatial trends of total petroleum hydrocarbons in the seawater of Bohai Bay China from 1996 to 2005," Marine Pollution Bulletin, vol. 60, no. 2, pp. 238-243, Feb 2010.
[125] O. Barnett and A. Cohen, "The histogram and boxplot for the display of lifetime data," Journal of Computational and Graphical Statistics, vol. 9, no. 4, pp. 759-778, Dec 2000.
[126] M. Bounessah and B. P. Atkin, "An application of exploratory data analysis (EDA) as a robust non-parametric technique for geochemical mapping in a semi-arid climate," Applied Geochemistry, vol. 18, no. 8, pp. 1185-1195, 8// 2003.
[127] T. Wegner, Applied Business Statistics: Methods and Excel-Based Applications. Juta, 2007.
[128] M. G. Bulmer, Principles of Statistics (Dover Books on Mathematics Series). Dover Publications, 1979.
[129] X. Yang et al., "Ultrasound GLCM texture analysis of radiation induced parotid gland injury in head-and-neck cancer radiotherapy An in vivo study of late toxicity," Medical Physics, vol. 39, no. 9, pp. 5732-5739, Sep 2012.

Appendix A - Exploratory feature analysis

Exploratory Data Analysis (EDA) is an approach that mostly uses visual methods to maximize the insight into the dataset at hand and reveal its main characteristics. Bi-histogram and box plot are the two graphical EDA tools that are used in this chapter to understand to what extent each feature can discriminate between normal and abnormal pixels. Sections A.1 and A.2 describe the bi-histogram and box plot. Section A.3 analyses the discrimination power of each feature by plotting its corresponding bi-histogram and box plot for normal and abnormal groups of pixels.

A.1 Bi-histogram plot

The bi-histogram is an Exploratory Data Analysis tool for assessing whether the histograms of two comparing groups of data have any differences in location, variation or distribution. Looking into our problem, in order to understand to what extent one feature can discriminate between normal and abnormal pixels, the bi-histogram of each feature for normal and abnormal groups of pixels is plotted. For each feature, to create the corresponding bi-histogram, the histogram of normal pixels which is shown in green is overlaid to the histogram of abnormal pixels, plotted in red. As a result, in the overlapped areas we have a mixture of green and red colors.

Moreover, in each bi-histogram plot we show the best distribution fit for both normal and abnormal pixels. To find the best distribution fit, the"allfitdist" function from MATLAB is used, which tries to fit all valid parametric probability distributions including: Beta, Birnbaum-Saunders, Exponential, Extreme value, Gamma, Generalized extreme value, Generalized Pareto, Inverse Gaussian, Logistic, Log-logistic, Lognormal, Nakagami, Normal, Rayleigh, Rician, t location-scale and Weibull. Afterwards, it will return the fittest distribution. For a detailed review on statistical distributions and their corresponding properties and parameters please refer to [122, 123]. The distribution of the features in our dataset is limited to a small variety of distributions which are briefly described in the following paragraphs:

Normal: The normal distribution is a two-parameter family of curves. The first parameter, μ, is the mean. The second,σ, is the standard deviation. The standard normal distribution sets μ to 0 and σ to 1. The probability density function (pdf) for normal distribution is stated in eq. (A.1).

$$y = f(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{A.1}$$

T location-scale: The probability density function of the t location-scale distribution is stated in eq. (A.2).

$$y = \frac{\Gamma\left(\frac{\upsilon+1}{2}\right)}{\sigma\sqrt{\upsilon\pi}\Gamma\left(\frac{\upsilon}{2}\right)}\left[\frac{\upsilon+\left(\frac{x-\mu}{\sigma}\right)^2}{\upsilon}\right]^{-\left(\frac{\upsilon+1}{2}\right)} \tag{A.2}$$

where $\Gamma(.)$ is the gamma function, μ is the location parameter, σ is the scale parameter, and υ is the shape parameter.

The t location-scale distribution is useful for modeling data distributions with heavier tails (more prone to outliers) than the normal distribution. It approaches the normal distribution as υ approaches infinity, and smaller values of υ yield heavier tails. The mean of the t location-scale distribution is only defined for shape parameter values $\upsilon > 1$ and is equal to the location parameter μ. The variance of the t location-scale distribution is only defined for values of $\upsilon > 2$ and is equal to eq. (A.3)

$$var = \sigma^2 \times \frac{\upsilon}{\upsilon-2} \tag{A.3}$$

Generalized Pareto: The probability density function for the generalized Pareto distribution with shape parameter k , scale parameter σ, and threshold parameter θ, is shown in eq. (A.4).

$$y = f(x|k,\sigma,\theta) = \left(\frac{1}{\sigma}\right)\left(1 + k\frac{(x-\theta)}{\sigma}\right)^{-1-\frac{1}{k}} \tag{A.4}$$

Generalized Pareto is a parametric distribution that agrees well with the data in areas of low density like the tail of other distributions. This is the main reason to use the generalized Pareto distribution. The beginning of the tail is identified by threshold parameter θ.

The generalized Pareto distribution has three basic forms, each corresponding to a limiting distribution of tail data from a different class of underlying distributions.

- Distributions whose tails decrease exponentially, such as the normal, lead to a generalized Pareto with $k = 0$.

- Distributions whose tails decrease as a polynomial, such as Student's t, lead to a positive shape parameter ($k > 0$).
- Distributions whose tails are finite, such as the beta, lead to a negative shape parameter ($k < 0$).

Generalized extreme value: The probability density function for the generalized extreme value distribution with location parameter μ, scale parameter σ, and shape parameter $k \neq 0$ is depicted in eq. (A.5)

$$y = f(x|k,\mu,\sigma) = \left(\frac{1}{\sigma}\right) exp\left(-\left(1 + k\,\frac{(x-\mu)}{\sigma}\right)^{-\frac{1}{k}}\right)\left(1 + k\,\frac{(x-\mu)}{\sigma}\right)^{-1-\frac{1}{k}} \qquad \text{(A.5)}$$

The generalized extreme value distribution is often used to model the smallest or largest value among a large set of independent, identically distributed random values representing measurements or observations. For example, one might have batches of 1000 washers from a manufacturing process. If the size of the largest washer in each batch is recorded, the data is known as block maxima (or minima in case of recording the smallest). The generalized extreme value distribution can be used as a model for those block maxima.

The generalized extreme value combines three simpler distributions into a single form, allowing a continuous range of possible shapes that includes all three of the simpler distributions. The three cases covered by the generalized extreme value distribution are often referred to as the Types I, II, and III. Each type corresponds to the limiting distribution of block maxima from a different class of underlying distributions. Distributions whose tails decrease exponentially, such as the normal, lead to the Type I ($k = 0$). Distributions whose tails decrease as a polynomial, such as Student's t, lead to the Type II ($k > 0$). Distributions whose tails are finite, such as the beta, lead to the Type III ($k < 0$). One can use any one of those distributions to model a particular dataset of block maxima. The generalized extreme value distribution allows the data decide which distribution is appropriate.

Extreme value: Like generalized extreme value distributions, extreme value distributions are also often used to model the smallest or largest value among a large set of independent, identically distributed random values representing measurements or observations. The extreme value distribution is appropriate for modeling the smallest value from a distribution whose tails decay

exponentially fast, for example, the normal distribution. It can also model the largest value from a distribution, such as the normal or exponential distributions, by using the negative of the original values. The probability density function for the extreme value distribution with location parameter μ and scale parameter σ is shown in eq. (A.6)

$$y = f(x|\mu,\sigma) = \sigma^{-1}\exp\left(\frac{x-\mu}{\sigma}\right) exp\left(-exp\left(\frac{x-\mu}{\sigma}\right)\right) \quad \text{(A.6)}$$

Logistic: The Logistic distribution resembles the normal distribution in shape but has heavier tails (higher kurtosis). The probability density function for Logistic distribution is stated in eq. (A.7).

$$y = f(x|\mu,\sigma) = \frac{exp\left\{\frac{x-\mu}{\sigma}\right\}}{\sigma\left(1+exp\left\{\frac{x-\mu}{\sigma}\right\}\right)^2} \;\; ; -\infty < x < +\infty \quad \text{(A.7)}$$

Where μ is the mean value and σ is the scale parameter.

In each bi-histogram plot, the best distribution fit for normal and abnormal subgroups of pixels is shown in blue and red color respectively. In order to discretize the continuous value of features, eq. (A.8), proposed by MATLAB, is used to calculate number of histogram bins.

$$Number\ of\ bins\ =\ max(min(length(data)/10{,}100){,}50) \quad \text{(A.8)}$$

A.2 Box plot

Box plot is also a tool for graphically describing groups of data in terms of their quartiles and gives the ability to detect and illustrate location and variation changes between different groups of data.

In order to draw a box plot of a group of data, we should first calculate the median and the quartiles (the lower quartile is the 25th percentile and the upper quartile is the 75th percentile) of dataset. Then a box is plotted between the lower and upper quartiles and the median is determined by a line inside the box. Next step is to draw the whiskers which show the variability outside the upper and lower quartiles. The lower whisker is a vertical line connecting the lower quartile to the lower adjacent value and the upper whisker is a line in between the upper quartile and the upper adjacent value. Lower adjacent value is the smallest value above the lower fence and upper adjacent value is the largest value below the upper fence. Lower and upper fences are calculated as shown in eq. (A.9) and (A.10) respectively.

$$Lower\ fence = q1 - {}^{3}/_{2}\,(q3 - q1) \tag{A.9}$$

$$Upper\ fence = q3 + {}^{3}/_{2}\,(q3 - q1) \tag{A.10}$$

Where q1 and q3 are the 25th and 75th percentiles, respectively. Data points whose values are smaller than the lower fence or greater than the upper fence are considered as outliers. It should be noted that outliers are not necessarily "bad" data points; indeed they may be the most important and information-rich part of the dataset. Therefore, they should not be removed from the dataset but may deserve special consideration [124]. When the underlying distribution is normal, the interval between the fences covers 99.3% of the distribution. The upper and lower fences are actually estimators of the .9965 and .0035 quantiles, respectively [125]. Fig. A.1 shows the definition of boxplot features.

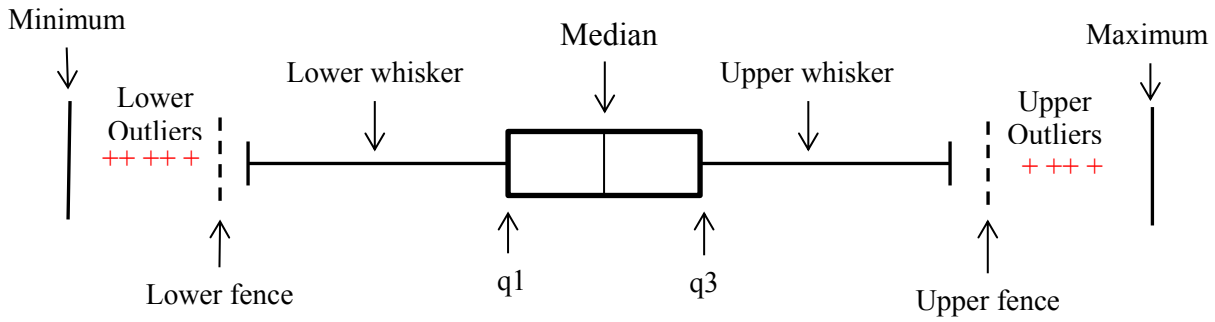


Fig. A.1 Definition of the boxplot features [126]

As stated in [127], we can use the boxplot to interpret the skewness of the data. The data is not skewed if its boxplot is symmetrical about the median which means that the quartiles are approximately equidistant from the median in both directions and the length of upper and lower whiskers are approximately equal to each other. However, if there is a long tail at the lower end of the boxplot, relative to the tail at the upper end, then the data is left skewed meaning that there are a few extremely small values in the dataset. Alternatively, if there is a long tail at the upper end of the boxplot, relative to the tail at the lower end, then the data is right skewed meaning that there are a few extremely large values in the dataset.

A.3 Feature analysis

Intensity(x, y)

The first feature in our dataset is the intensity value for normal and abnormal pixel $P(x, y)$. From the bi-histogram of the first feature which is shown in Fig. A.2-a, we can see that normal pixels are centered at a value of approximately -0.1 while abnormal pixels are centered at a value of approximately -0.3 (i.e., The center of a histogram is the location of its highest bin. In other words, the center of a histogram tells us which value is the most frequent value within our data). That indicates that the two subgroups are displaced by about 0.2 units. Thus whether a pixel is normal or abnormal has an effect on the location (most frequent value) for intensity feature. From the boxplot of intensity values for normal and abnormal pixels, that is shown in Fig. A.2-b, one can see that the median value for normal and abnormal pixels are also around -0.11 and -0.31 respectively. From Fig. A.3 it can be seen that the mean values of intensity feature for normal and abnormal pixels are around -0.14 and -0.3 respectively which shows that the abnormal pixel are darker than the normal pixels in our dataset.

With respect to variation (i.e., the variation or spread of a histogram is the range of values that bins of histogram cover), the spread of the normal pixels is more than the abnormal pixels. It is true; because in a normal CT series, we have the ventricles in the middle of the brain which appears very dark as well as the white matters which are quite lighter. This fact produces a high variation within intensity values of normal pixels. On the other hand, since most of the regions that is marked as abnormal in our dataset were ischemic stroke (i.e., which produces darker intensity values in CT images), the variation of intensity value within abnormal pixels is smaller than the normal group.

As we can see, the best distribution fit for both normal and abnormal pixels belongs to t location scale family. Moreover, normal pixels have heavier tails than abnormal pixels ($\nu_{normal} < \nu_{abnormal}$). If we take a look to the boxplot of this feature, we can see that, despite the normal pixels for which $|q2 - q1| > |q3 - q2|$, abnormal pixels have equidistant quartiles from the median. Moreover, for normal pixels $length(lower\ whisker) > length(upper\ whisker)$ while for abnormal $length(upper\ whisker) = length(lower\ whisker)$ which certifies that the distribution of normal pixels is left skewed but the distribution of abnormal pixels is more or less symmetrical. It makes sense; because if you look into a normal CT slice of a brain after removing

the skull and the other artifacts, there exists very dark pixels like the ones in ventricles but very light pixels are very rare which makes the shape of distribution for normal pixels left skewed.

Regarding feature 1, the bi-histogram, boxplot and mean plot reveal that there is a clear difference between the normal and abnormal sub-groups of pixels with respect to mean, median and location but their variation are not quite different. Moreover, their distributional shape belongs to the same family.

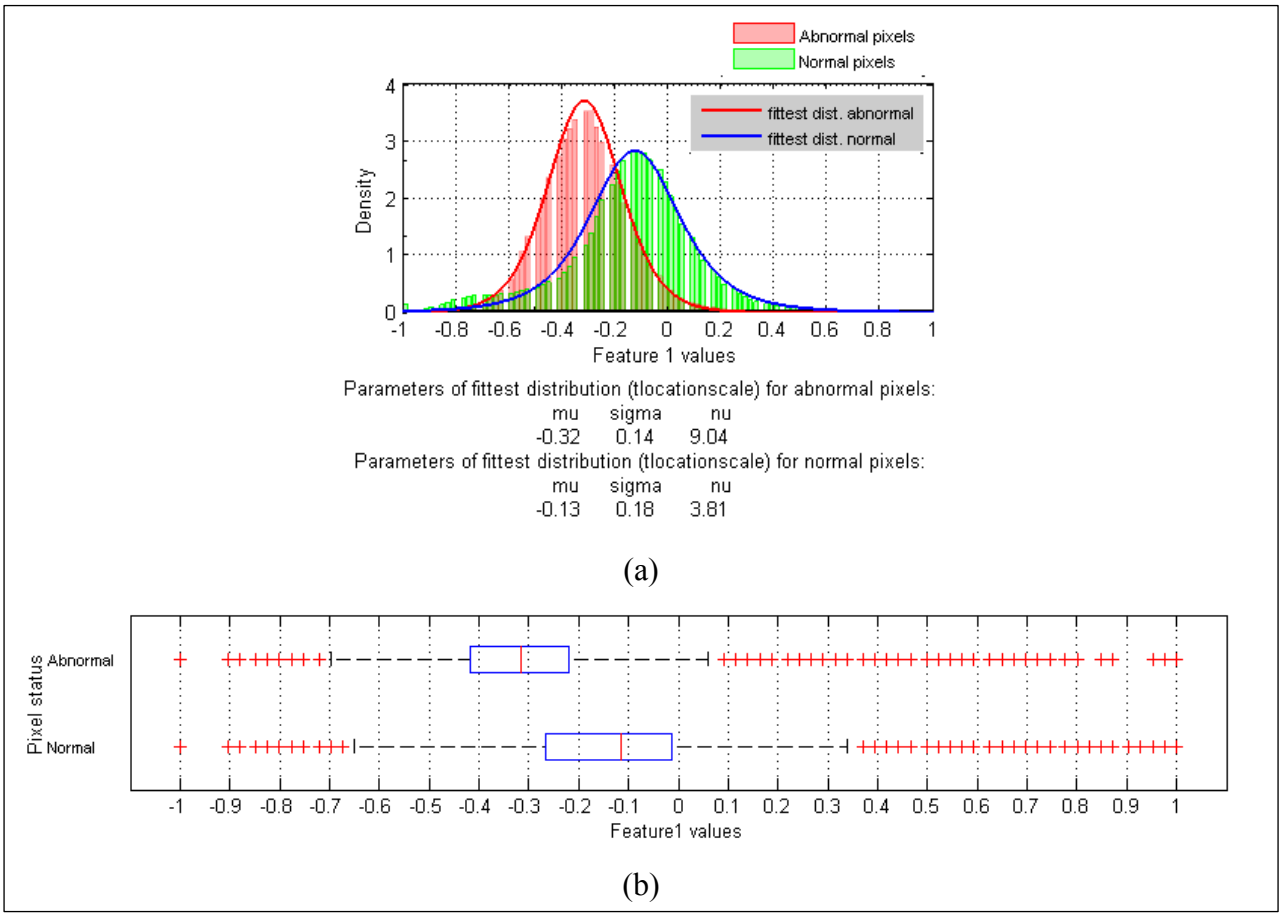


Fig. A.2: (a) bi-histogram of Feature 1; (b) box plot of Feature 1.

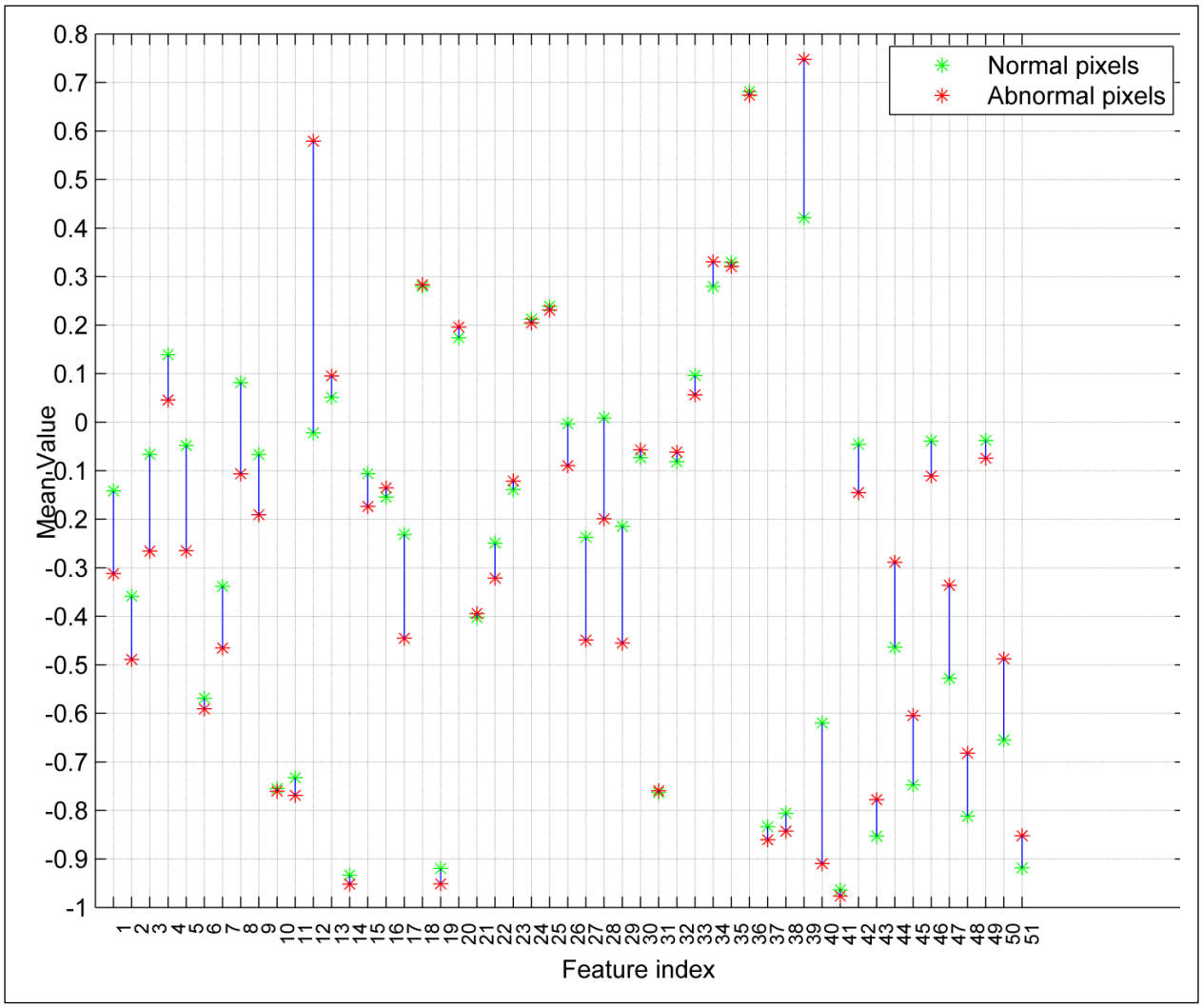


Fig. A.3 Mean values of each feature for normal and abnormal sub-groups of pixels

$$\underset{m,n \in w}{\mathbf{Min}} \; I(m,n)$$

The 2nd feature in our dataset is the minimum value of the intensity values within window w of size 31×31 centered at normal or abnormal pixel $P(x,y)$. From the bi-histogram of the second feature which is shown in Fig. A.4-a, we can see that that the most frequent value for normal pixels is -1 while the location of abnormal pixels is around -0.4. That indicates that the two subgroups are displaced by about 0.6 units. Thus whether a pixel is normal or abnormal has an effect on the location (most frequent value) of 2nd feature. From the boxplot of feature 2 that is shown in Fig. A.4-b, one can see that the median value for normal and abnormal pixels are around -0.3 and -0.47

respectively. From Fig. A.3 it can be seen that the mean values of feature 2 for normal and abnormal pixels are around -0.35 and -0.48 accordingly.

With respect to variation, the spread (variation) of the normal pixels is more than the abnormal pixels. Moreover, the best distribution fit for abnormal pixels belongs to t location scale family while normal pixels are modeled by a generalized Pareto distribution with a negative shape parameter ($k = -0.43$). If we take a look to the boxplot of this feature, we can see that for both normal and abnormal pixels $|q2 - q1| > |q3 - q2|$. On the other hand, regarding to whisker length we have $length(lower\ whisker) < length(upper\ whisker)$ for both normal and abnormal pixels. As a result there is no clue about whether the two subgroups are symmetric or skewed.

Regarding feature 2, the bi-histogram, boxplot and mean plot reveal that there is a clear difference between normal and abnormal sub-groups of pixels with respect to mean, median, location as well as distribution. Their variation is also to some extent different.

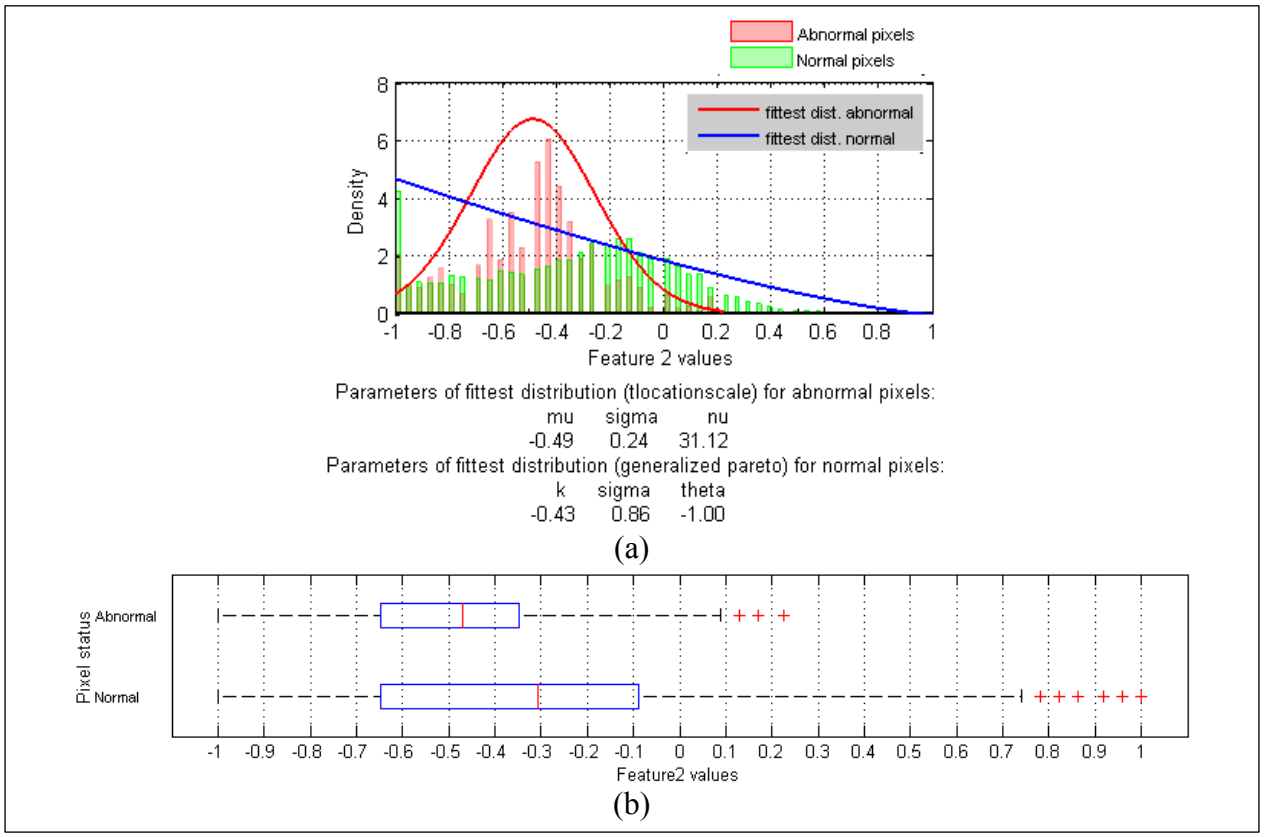


Fig. A.4: (a) bi-histogram of Feature 2; (b) box plot of Feature 2.

$$\underset{m,n \in w}{\textbf{Mean}}\, \boldsymbol{I(m,n)}$$

The 3rd feature in our dataset is the average intensity values within window w of size 31×31 centered at normal or abnormal pixel $P(x, y)$. From the bi-histogram of the 3rd feature which is shown in Fig. A.5-a, we can see that normal pixels are centered at a value of approximately -0.1 while abnormal pixels are centered at a value of approximately -0.3. That indicates that the two subgroups are displaced by about 0.2 units. Thus whether a pixel is normal or abnormal has an effect on the location (most frequent value) for feature 3. From the boxplot of feature 3 for normal and abnormal pixels, that is shown in Fig. A.5-b, one can see that the median value for normal and abnormal pixels are also around -0.05 and -0.29 respectively. From Fig. A.3 it can be seen that the mean values of Feature 3 for normal and abnormal pixels are around -0.06 and -0.26 respectively which shows that the pixels around the abnormal pixels are darker in average than the pixels around the normal ones in our dataset.

With respect to variation, the spread of normal pixels is more than the abnormal pixels. It is true; because in a normal CT series, we have the ventricles in the middle of the brain which appears very dark as well as the white matters which are quite lighter. This fact produces a high variation within the average intensity values around normal pixels. On the other hand, since most of the regions that is marked as abnormal in our dataset were ischemic stroke (i.e., which produces darker intensity values in CT images), the variation of the average intensity values around abnormal pixels is smaller than the normal group.

As we can see, the best distribution fit for normal pixels is logistic which resembles the normal distribution in shape but has heavier tails while the best distribution fit for abnormal pixels belongs to generalized extreme value family (i.e., Type III because of negative shape parameter). If we take a look to the boxplot of this feature, we can see that normal pixels have a symmetric boxplot. For abnormal pixels, the length of lower and upper whiskers is approximately the same but we have $|q2 - q1| < |q3 - q2|$. As a result, one can say that abnormal pixels is to some extent right skewed. It is probably because of the abnormal pixels that reside in the boundary of the lesion. In such cases, since the feature is getting the average of intensity values within a window centered at these boundary pixels, there will be some normal pixels within the window that are lighter and have greater intensity values which makes the average a larger value.

Regarding feature 3, the bi-histogram, mean plot and the boxplot reveal that there is a clear difference between normal and abnormal pixels with respect to mean, median, location, distribution as well as the variation.

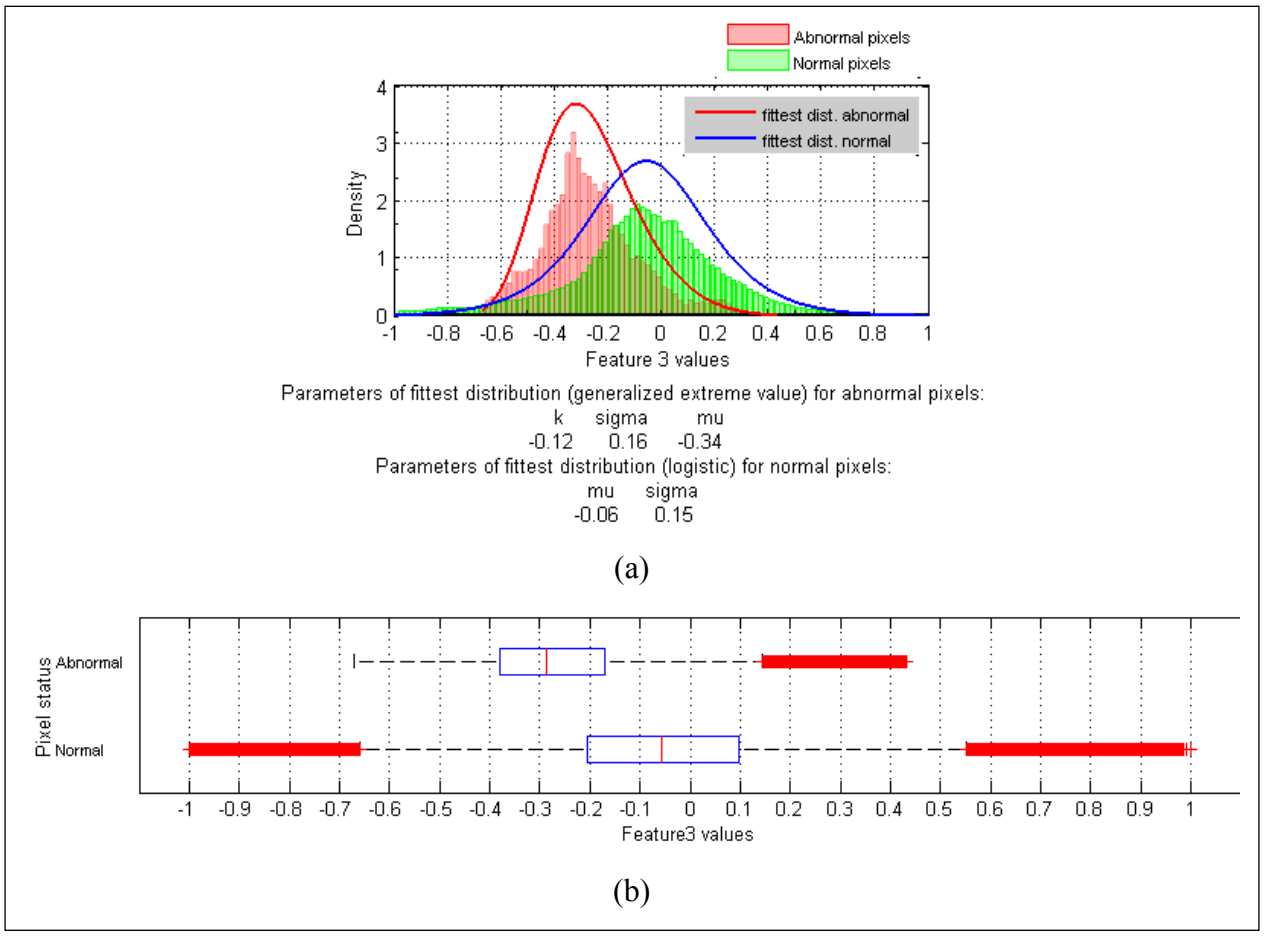


Fig. A.5: (a) bi-histogram of Feature 3; (b) box plot of Feature 3.

$$\underset{\mathbf{m},\mathbf{n} \in \mathbf{w}}{\mathbf{Max}}\ \mathbf{I}(\mathbf{m},\mathbf{n})$$

The 4th feature in our dataset is the maximum intensity values within window w of size 31×31 centered at normal or abnormal pixel $P(x, y)$. From the bi-histogram of the 4th feature which is shown in Fig. A.6-a, we can see that the histogram of both normal and abnormal pixels have edge peaked shape with skewness to the right. The peak on the right hand side of both histograms indicates that around 11% of normal data samples and 10% of abnormal data samples have at least one pixel within their 15×15 neighbor (i.e., $P(x, y)$ is located in the center of w) whose intensity value is maximum along the whole CT slice. The second histogram peak for normal and abnormal

pixels is located around -0.1 and -0.3 respectively. That indicates that the two subgroups are displaced by about 0.2 units. As we can see, both normal and abnormal pixels are modeled by generalized Pareto distribution family with negative shape parameters.

From the boxplot of this feature that is shown in Fig. A.6-b, we can see that the boxplot of normal pixels is right skewed because the upper whisker is obviously longer than the lower whisker and $|q2 - q1| < |q3 - q2|$. For abnormal pixels, however the upper and lower whiskers are more or less equidistant but $|q2 - q1| < |q3 - q2|$. As a result, one can say that abnormal pixels are also right skewed. The median value for normal and abnormal pixels are around -0.05 and -0.21 respectively.

From Fig. A.3 it can be seen that the average values of Feature 4 for normal and abnormal pixels are around 0.13 and 0.04 respectively which shows that the brightest pixels around the abnormal pixels are still darker than the brightest pixels around the normal pixels in our dataset. With respect to variation, the spread of normal pixels is more than abnormal pixels.

The bi-histogram, mean plot and the boxplot reveal that there is no clear difference between the maximum intensity values around the normal and abnormal pixels with respect to location and distribution but their mean, median and variation are different.

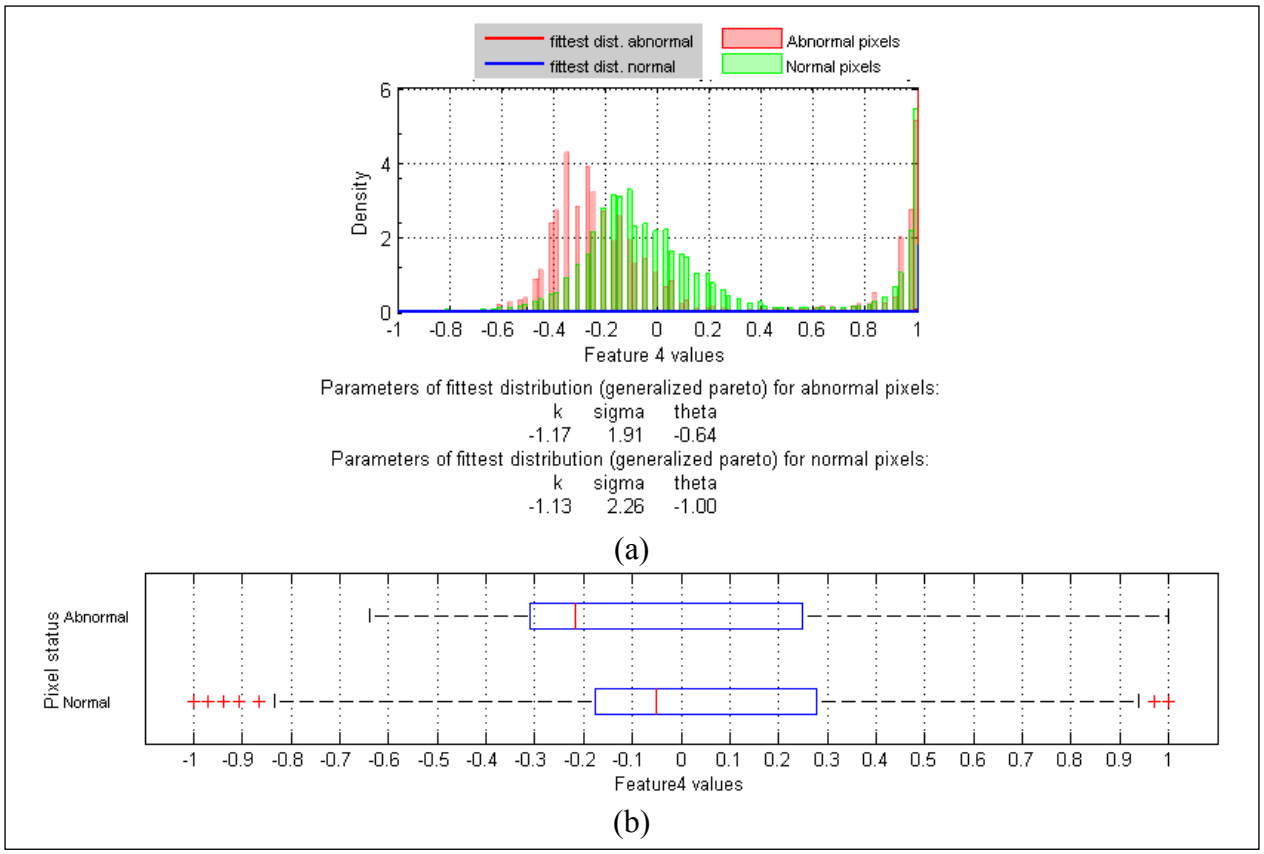


Fig. A.6: (a) bi-histogram of Feature 4; (b) box plot of Feature 4.

Median $\underset{m,n \in w}{\boldsymbol{I(m,n)}}$

The 5th feature in our dataset is the median intensity values within window w of size 31×31 centered at normal or abnormal pixel $P(x, y)$. From the bi-histogram of the 5th feature which is shown in Fig. A.7-a, we can see that the histogram of normal pixels is centered at a value of approximately -0.1 while abnormal pixels are peaked around -0.3. Moreover, the best distribution fit for normal pixels is t location scale while abnormal pixels are modeled by a Type III generalized extreme value distribution (i.e., negative shape parameter).

From the boxplot of Feature 5 that is shown in Fig. A.7-b, one can see that the median value for normal and abnormal pixels are around -0.01 and -0.28 respectively. Moreover, for both normal and abnormal pixels, the boxplots are symmetrical meaning that there is no skewness (i.e., $|q2 - q1| \cong |q3 - q2|$ and $h(lower\ whisker) \cong length(upper\ whisker)$).

From Fig. A.3 it can be seen that the average values of Feature 5 for normal and abnormal pixels are around -0.04 and -0.26 respectively. With respect to variation, the spread of the normal pixels is more than the abnormal pixels. It is true; because in a normal CT series, we have the ventricles in the middle of the brain which appears very dark as well as the white matters which are quite lighter. This fact produces a high variation within the median intensity values depending on the location of window *w*. On the other hand, since most regions that are marked as abnormal in our dataset were ischemic areas (i.e., which produces darker intensity values in CT images), the variation of the median values of window *w* around abnormal pixels is smaller than the normal group.

Regarding feature 5, the bi-histogram, mean plot and the boxplot reveal that there is a difference between normal and abnormal pixels with respect to median, mean, location, distribution as well as variation.

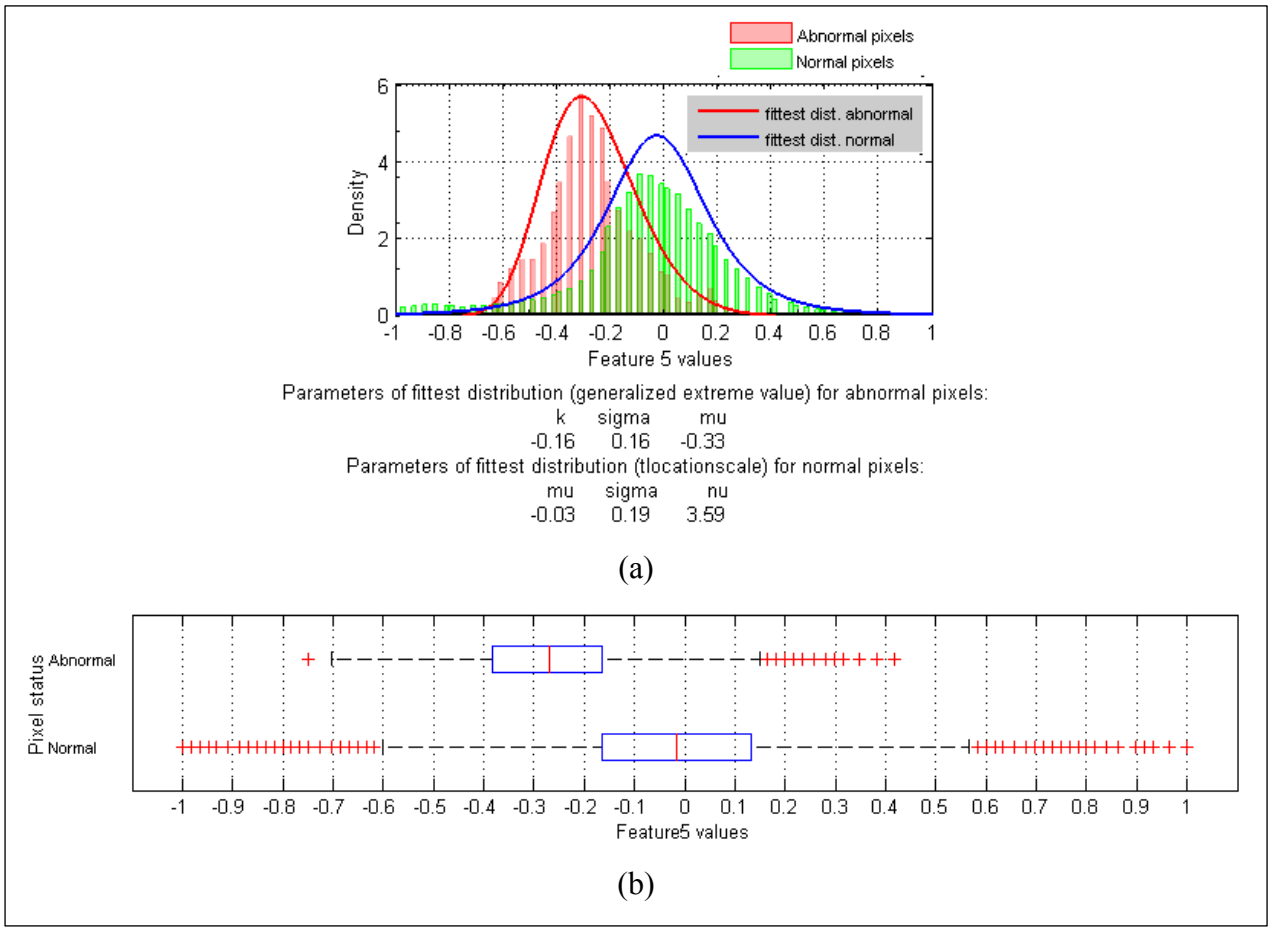


Fig. A.7: (a) bi-histogram of Feature 5; (b) box plot of Feature 5.

$\mathbf{Std}_w$

The 6th feature in our dataset is the standard deviation of intensity values within window w of size 31×31 centered at normal or abnormal pixel $P(x, y)$. From the bi-histogram of the 6th feature which is shown in Fig. A.8-a, we can see that the best distribution fit for both normal and abnormal pixels is Type II generalized extreme value ($k > 0$) whose tails decrease as a polynomial.

The boxplot that is shown in Fig. A.8-b certifies that the distribution of normal and abnormal pixels is right skewed because in both cases we have $|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) < length(upper\ whisker)$. Having right skewed distributions means that regardless of whether a pixel is normal or abnormal, in most cases, the intensity values within the 15×15 neighborhood area around pixel $P(x, y)$ are not very far away from the average value of the neighborhood. This fact makes the corresponding standard deviation small. In the case of having high standard deviation, it is probable that pixel $P(x, y)$ is located in a boundary region and the window that is specifying the neighborhood area is covering two different types of tissue (e.g., a pixel located in the boundary of ventricle and gray matter or a pixel located in the boundary of lesion). Moreover, the median value for normal and abnormal pixels are around -0.62 and -0.65 respectively.

From the histogram plot we can see that normal pixels are centered at a value of approximately -0.8 while abnormal pixels are centered at a value of approximately -0.7. The mean values of feature 6 for normal and abnormal pixels, shown in Fig. A.3, are around -0.56 and -0.59 respectively which are very close to each other. With respect to variation, the spread of the normal pixels is approximately the same as abnormal pixels.

Regarding feature 6, the bi-histogram, mean plot and the boxplot reveal that there is no clear difference between normal and abnormal pixels with respect to median, mean, location, distribution as well as the variation.

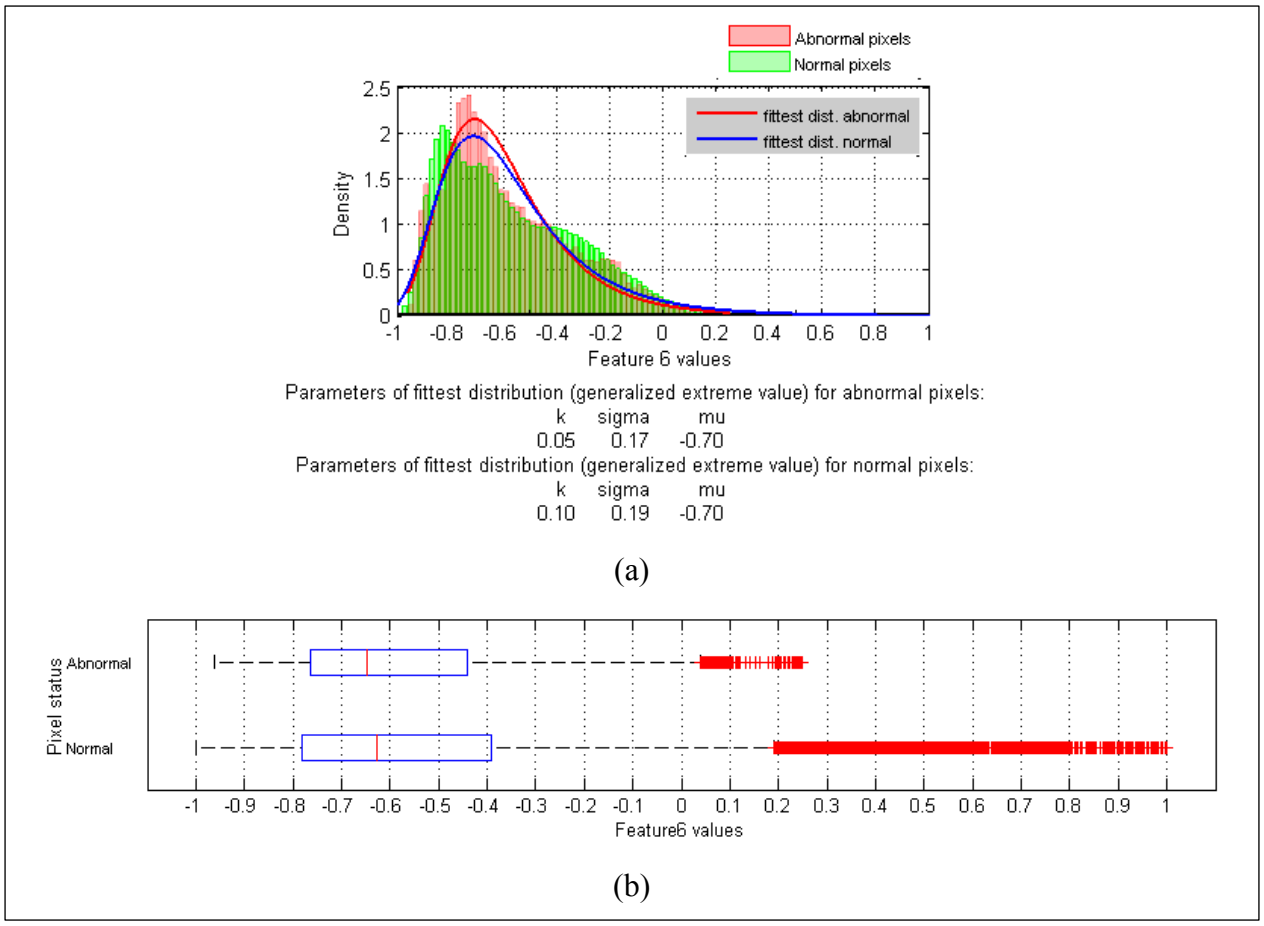


Fig. A.8: (a) bi-histogram of Feature 6; (b) box plot of Feature 6.

Mean value of the whole CT slice

The 7th feature is the mean value of the pixel intensities within the whole CT slice after removing the skull and other artifacts. As a result, regardless of whether a pixel is normal or abnormal, all pixels within a particular CT slice have the same value for this feature that is the average of intensity values of all cranial pixels within a CT slice.

From the bi-histogram of the 7th feature which is shown in Fig. A.9-a, we can see that the best distribution fit for both normal and abnormal pixels is generalized Pareto ($k < 0$) which agrees well with the data in low density regions

The most frequent value of this feature for normal pixels is -1 and for abnormal pixels is around -0.7. With respect to variation, the spread of the normal pixels is approximately the same as abnormal pixels. Moreover, if we take a look to the boxplot of this feature which is shown in Fig.

A.9-b, we can see that for normal pixels $|q2 - q1| > |q3 - q2|$ but $length(lower\ whisker) < length(upper\ whisker)$; as a result there is no clue about whether normal pixels are symmetric or skewed. For abnormal pixels, since there exists conditions $|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) < length(upper\ whisker)$, one can say that the corresponding distribution is right skewed. Furthermore, the boxplot indicates that the median value for normal and abnormal pixels are around -0.29 and -0.6 correspondingly.

The mean values of Feature 7 for normal and abnormal pixels, shown in Fig. A.3, are around -0.33 and -0.46 respectively.

Regarding to Feature 7, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between the normal and abnormal pixels with respect to distribution as well as the variation but their location, median and mean values are different.

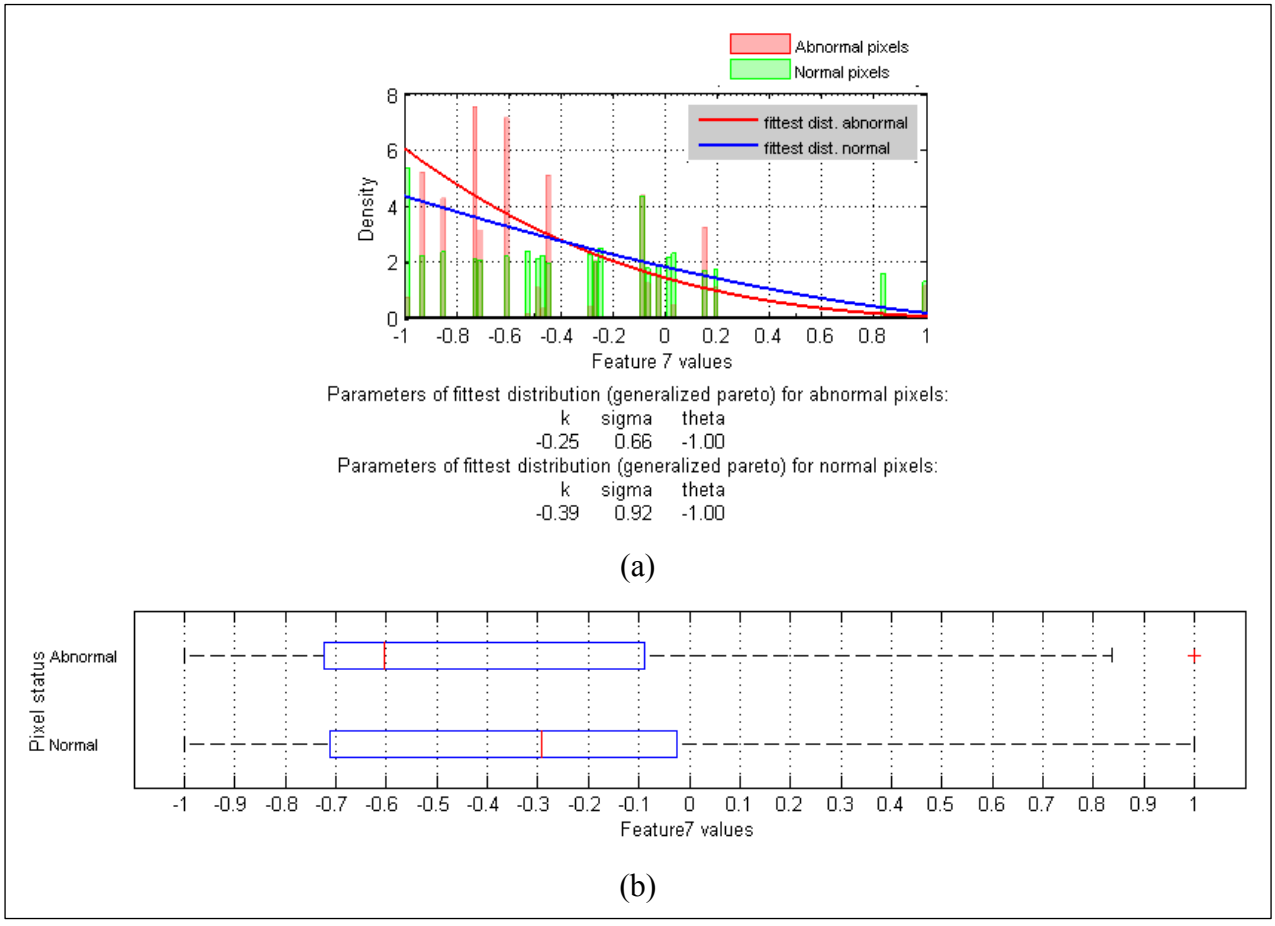


Fig. A.9: (a) bi-histogram of Feature 7; (b) box plot of Feature 7.

$\underset{m,n \in w}{\text{Mean}}\, I(m,n) - \text{Mean value of the whole CT slice}$

The 8th feature in our dataset wants to determine how far the average intensity value around a pixel under the study is from the average intensity value of the whole cranial part of the CT slice. As a result, this feature is obtained by subtracting Feature 7 (i.e., the average of intensity values of all cranial pixels within a CT slice) from Feature 3 (i.e., the average intensity values within window w of size 31×31 centered at normal or abnormal pixel (x, y)). From the bi-histogram of the 8th feature which is shown in Fig. A.10-a, we can see that normal pixels are centered at a value of approximately 0.1 while abnormal pixels are centered at a value of approximately -0.1. That indicates that the two subgroups are displaced by about 0.2 units.

From the boxplot of Feature 8 that is shown in Fig. A.10-b, one can see that the median value for normal and abnormal pixels are around 0.1 and -0.09 respectively. Moreover, the boxplot of both normal and abnormal pixels is approximately symmetrical meaning that there is no skewness. From Fig. A.3 it can be seen that the mean value of Feature 8 for normal and abnormal pixels are around 0.08 and -0.1 respectively.

With respect to variation, the spread of the normal pixels is more than abnormal pixels. It is true since, as it was previously mentioned in the analysis of Feature 3, in a normal CT series, we have the ventricles in the middle of the brain which appears very dark as well as the white matters which are quite lighter. This fact produces a high variation within the average intensity values around normal pixels. The high variation will be kept after subtracting a constant value that is the average intensity value of the whole cranial part. On the other hand, since most regions that are marked as abnormal in our dataset were ischemic areas (i.e., which produces darker intensity values in CT images), the variation of the average intensity values around abnormal pixels is smaller than the normal group which will be resulted in a lower variation of Feature 8 for abnormal pixels as well.

As we can see, the best distribution fit for normal pixels is logistic which resembles the normal distribution in shape but has heavier tails (higher kurtosis). Abnormal pixels are also modeled by a normal distribution.

Regarding to Feature 8, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between the normal and abnormal pixels with respect to distribution but their location, variation, median and mean values are different.

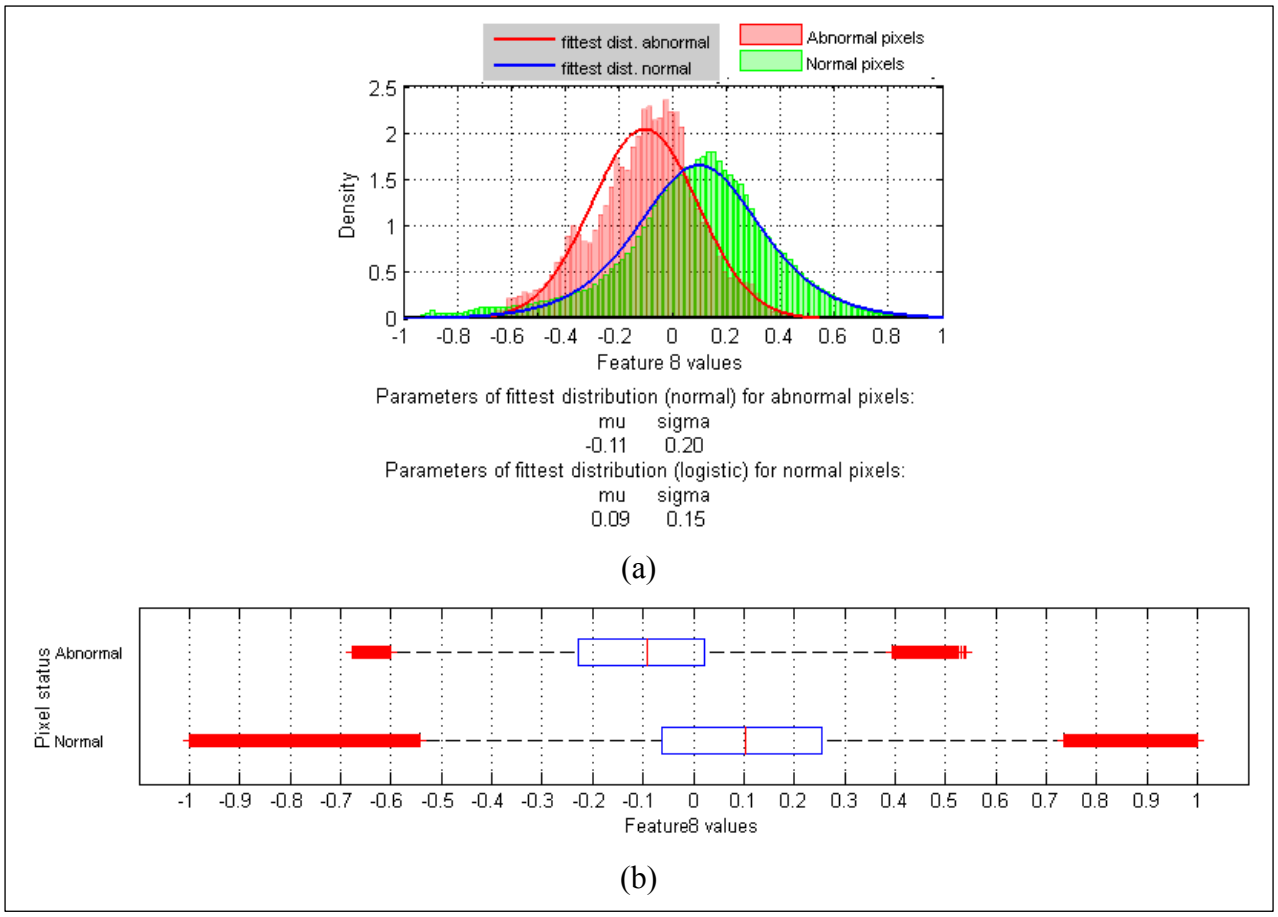


Fig. A.10: (a) bi-histogram of features 8; (b) box plot of features 8.

$\mathbf{Intensity(x,y) - Mean\ value\ of\ the\ whole\ CT\ slice}$

The 9th feature in our dataset wants to determine how far the intensity value of a pixel under the study is from the average intensity value of the whole cranial part of the CT slice. As a result, this feature is obtained by subtracting Feature 7 (i.e., the average of intensity values of all cranial pixels within a CT slice) from Feature 1 (i.e., $Intensity(x,y)$). From the bi-histogram of the 9th feature which is shown in Fig. A.11-a, we can see that normal pixels are centered at a value of 0 while abnormal pixels are centered at a value of approximately -0.15. That indicates that the two subgroups are displaced by about 0.15 units. Moreover, the best distribution fit for both normal and abnormal pixels belongs to t location scale family. The smaller ν value for normal pixels shows that its corresponding distribution has heavier tails than abnormal pixels.

From the boxplot of Feature 9 for normal and abnormal pixels, that is shown in Fig. A.11-b, one can see that the median value for normal and abnormal pixels are around -0.05 and -0.19 respectively. Furthermore, the boxplot of both normal and abnormal pixels seems to be symmetrical which means that there exists no skewness. From Fig. A.3 it can be seen that the mean value of Feature 9 for normal and abnormal pixels are around -0.06 and -0.19 respectively. With respect to variation, the spread of the normal pixels is more than abnormal pixels.

Regarding to Feature 9, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between the normal and abnormal pixels with respect to distribution but their location, variation, median and mean values are different.

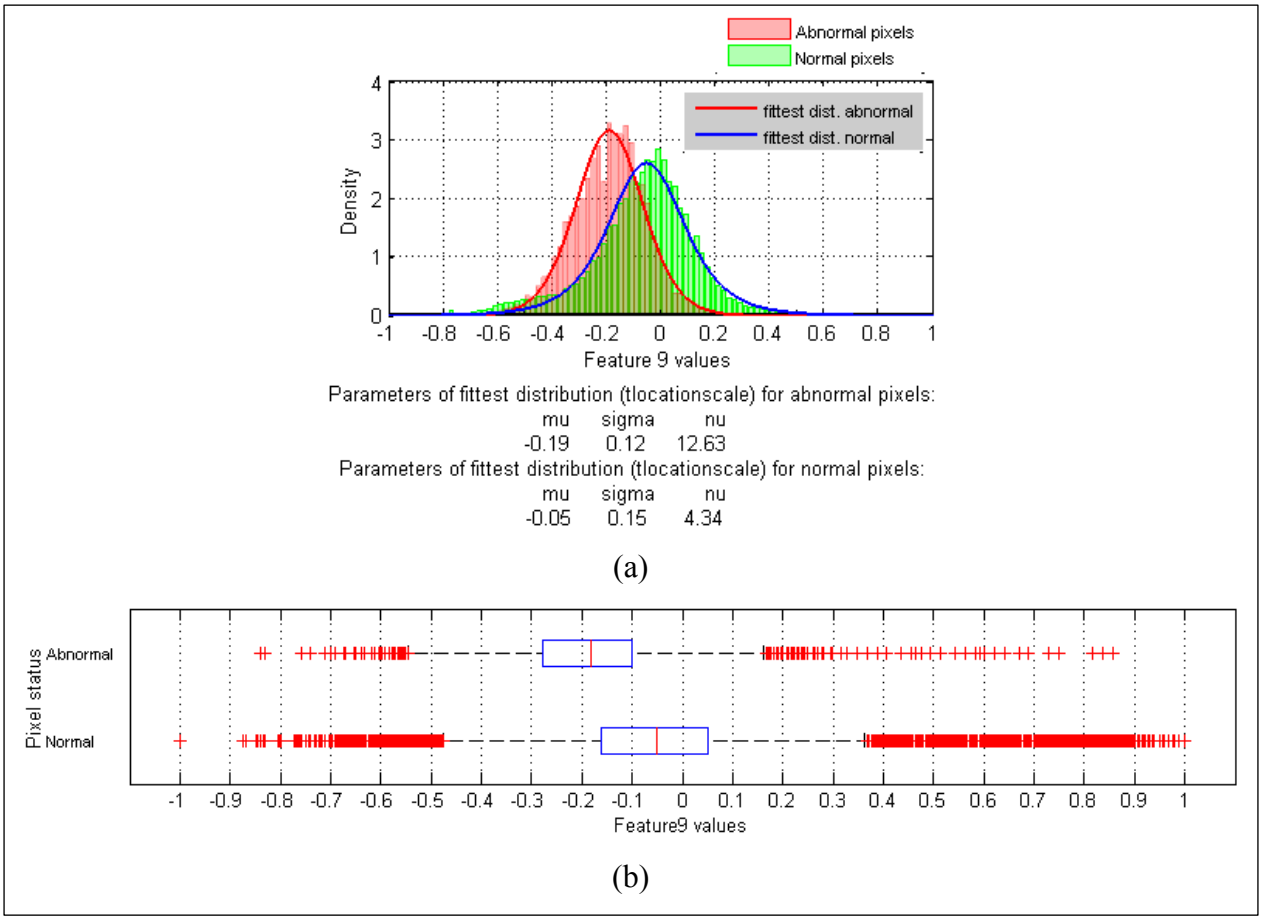


Fig. A.11: (a) bi-histogram of features 9; (b) box plot of features 9.

Plh

The 10th feature in our dataset is the accumulated differences between the intensities of a vector of horizontally adjacent pixels centered at the pixel $P(x, y)$ under the study. The vector has a length of 31 pixels. In fact, this feature wants to measure the degree of intensity homogeneity in a horizontal direction around the pixel under the study.

From the bi-histogram of the 10th feature which is shown in Fig. A.12-a, we can see that the best distribution fit for both normal and abnormal pixels belongs to type II generalized extreme value whose tails decrease as a polynomial ($k > 0$). Moreover, the most frequent value of both normal and abnormal pixels is about -0.8.

The boxplot that is shown in Fig. A.12-b certifies that the distribution of normal and abnormal pixels is right skewed because in both cases we have $|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) < length(upper\ whisker)$. Having right skewed distributions means that regardless of whether a pixel is normal or abnormal, in most cases, the intensity values within the horizontally adjacent pixels next to pixel $P(x, y)$ are not very far away from each other and they make a horizontally homogeneous texture. This fact makes the corresponding plh value small. Furthermore, from the boxplots we can see that the spread of normal pixels is more than the abnormal ones. The median value for normal and abnormal pixels is around -0.79 and -0.8 respectively.

The mean values of Feature 10 for normal and abnormal pixels, shown in Fig. A.3, are around -0.75 and -0.76 respectively which are very close to each other.

Regarding to Feature 10, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between the normal and abnormal pixels with respect to distribution, location, median and mean values but their spread are to some extent different.

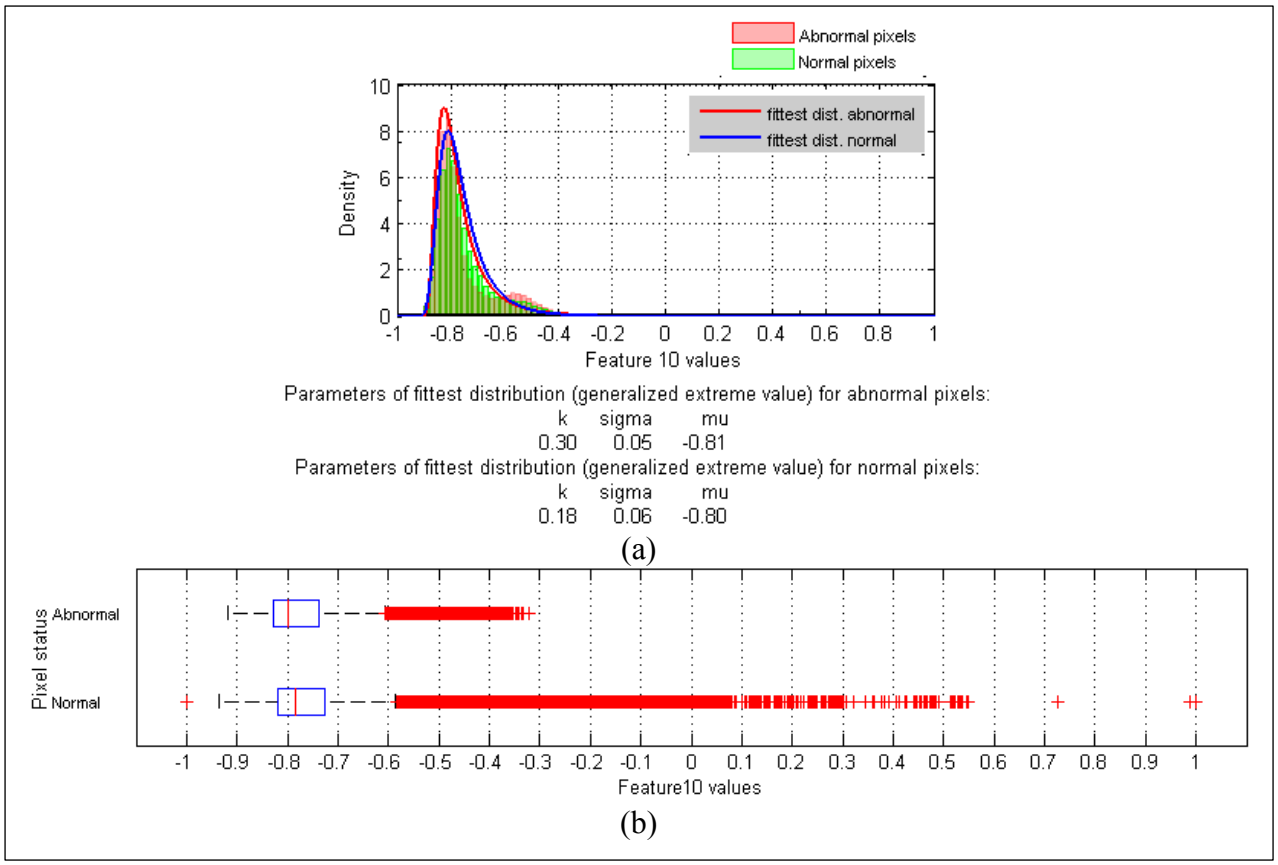


Fig. A.12: (a) bi-histogram of features 10; (b) box plot of features 10.

Plv

The 11th feature in our dataset is the accumulated differences between the intensities of a vector of vertically adjacent pixels centered at the pixel $P(x, y)$ under the study. The vector has a length of 31 pixels. In fact, this feature wants to measure the degree of intensity homogeneity in a vertical direction around the pixel under the study.

From the bi-histogram of the 11th feature which is shown in Fig. A.13-a, we can see that the best distribution fit for both normal and abnormal pixels belongs to type II generalized extreme value whose tails decrease as a polynomial ($k > 0$). Moreover, the most frequent value of both normal and abnormal pixels is about -0.8. If we take a closer look to the bi-histogram, we can see that the majority part of normal and abnormal data samples have their plv value within range $[-0.9, -0.7]$ which are all small values and reflect the fact of having vertically homogeneous texture in the close neighborhood of either normal or abnormal pixels.

As it can be seen in the boxplot of plv feature, shown in Fig. A.13-b, $|q2 - q1| < |q3 - q2|$ for both normal and abnormal pixels but the length of upper and lower whiskers, in both cases, are more or less equal. As a result one can say that both distributions are right skewed with a notice that normal pixels are more skewed than the abnormal ones, covering a bigger range of values (higher variation). Having right skewed distributions means that regardless of whether a pixel is normal or abnormal, in most cases, the intensity values within the vertically adjacent pixels next to pixel $P(x, y)$ are not very far away from each other and they make a vertically homogeneous texture. This fact makes the corresponding plv value small. Moreover, the boxplot indicates that the median value for normal and abnormal pixels is around -0.78 and -0.8 respectively. The mean value of Feature 11 for normal and abnormal pixels, shown in Fig. A.3, are around -0.73 and -0.76 respectively which are very close to each other.

Regarding to Feature 11, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between the normal and abnormal pixels with respect to distribution, location, median and mean values but their variation are different.

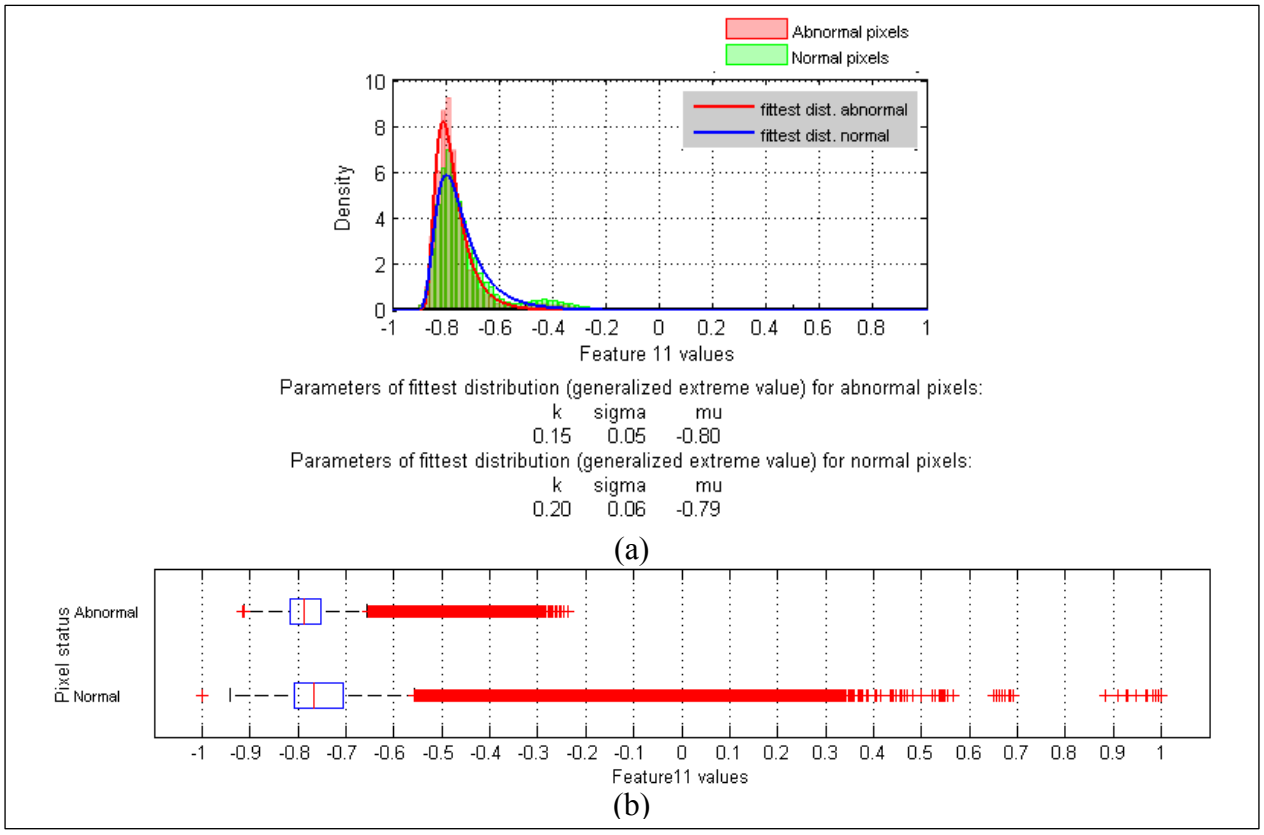


Fig. A.13: (a) bi-histogram of features 11; (b) box plot of features 11.

$x/512$

The 12th feature in our dataset determines how much the location of the pixel under the study tends to the left or right. Since brain CT images in DICOM format have a dimension of 512×512 pixels the value of Feature 12 will be within range $[0,1]$ in which 0 and 1 values represent the most left and the most right locations respectively. It should be noticed that our dataset is normalized between $[-1,1]$ afterwards and therefore range $[0,1]$ has been mapped to $[-1,1]$. As a result, -1 and 1 values represent the most left and the most right locations respectively.

From the bi-histogram of the 12th feature which is shown in Fig. A.14-a, we can see that the best distribution fit for normal pixels is type III generalized extreme value whose tails are finite ($k < 0$) while abnormal pixels are modeled extreme value distribution family. Looking closer into the bi-histogram, we can see that in our dataset the probability of having normal pixels in both sides of the brain is to some extent equal. On the other hand, the distribution shape of abnormal pixels seems to be bi-modal with the peaks on -0.4 and 0.6. As we can see, the bins around value 0.6 are quite longer than the ones around -0.4 which indicates that, in our dataset, most lesions are placed in the right side of the brain. The mean value of Feature 12 for normal and abnormal pixels, shown in Fig. A.3, are around -0.02 and -0.5 respectively.

From the boxplot of Feature 12 for normal and abnormal pixels, that is shown in Fig. A.14-b, one can see that the median value for normal and abnormal pixels is around -0.02 and 0.61 respectively. Moreover, the boxplot of normal pixels is symmetrical but for abnormal pixels we have $|q2 - q1| \cong |q3 - q2|$ and $length(lower\ whisker) > length(upper\ whisker)$; as a result there exists some skewness to the left.

Regarding to Feature 12, the bi-histogram, boxplot and the mean plot reveal that there is a clear difference between the normal and abnormal pixels with respect to distribution, location, median, mean as well as their variation.

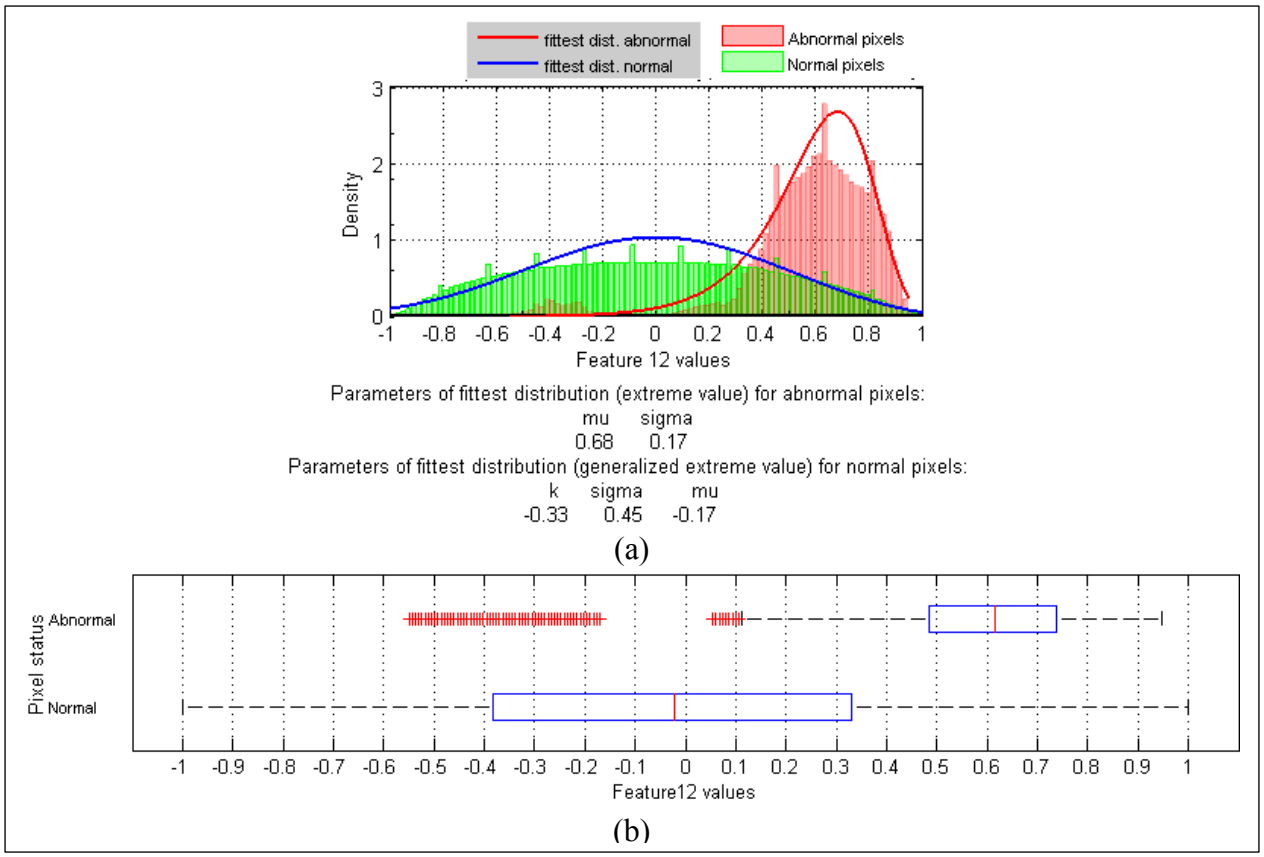


Fig. A.14: (a) bi-histogram of features 12; (b) box plot of features 12.

Skewness

Given pixel $P(x, y)$, the 13^{th} feature in our dataset wants to quantify how symmetrical the distribution of the intensity values in a 15×15 neighborhood area of $P(x, y)$ is. If the distribution is symmetrical, the skewness value will be 0. An asymmetrical distribution with a long tail to the right (higher values) has a positive skew while an asymmetrical distribution with a long tail to the left (lower values) has a negative skew.

From the bi-histogram of the 13^{th} feature which is shown in Fig. A.15-a, we can see that the best distribution fit for normal is logistic which resembles the normal distribution in shape but has heavier tails (higher kurtosis). After summing up the frequency values for normal bins residing within range$[-0.1,0.2]$, we saw that around 23% have their skewness value within $[-0.1,0]$, around 38% are within $[0,0.1]$; and around 22% are between $[0.1,0.2]$. According to eq. (A.11) which is depicted in [128], if we consider these ranges within their original scale (i.e., the original

of feature 13 is within $[-8.4, 7.09]$), we can conclude that around 38% of normal data samples are approximately symmetric (i.e., $skewness \in [-0.6,0.1]$). Around 23% are moderately left skewed (i.e., $skewness \in [-1.4, -0.6]$) and around 22% are moderately right skewed (i.e., $skewness \in [0.1,0.8]$). As a result we can say that in most cases (around 83%) there is no sudden change of intensity value in a 15×15 neighborhood area of the normal pixels, otherwise there would be highly left or right skewed of intensity distribution in their close neighborhood.

$$distribution = \begin{cases} highly\ left\ skewed & skewness < -1 \\ moderately\ left\ skewed & -1 < skewness < -\frac{1}{2} \\ approximately\ symmetric & -\frac{1}{2} < skewness < \frac{1}{2} \\ moderately\ right\ skewed & \frac{1}{2} < skewness < 1 \\ highly\ right\ skewed & skewness > 1 \end{cases} \quad \text{(A.11)}$$

The best distribution fit for abnormal pixels belongs to t location scale family. Taking a closer look into the histogram, we can see that the skewness of around 80% of abnormal pixels is within range$[0,0.2]$. Having mapped range $[0,0.2]$ to its original scale, we have the result of $skewness \in [-0.6,0.8]$ which indicates that around 80% of abnormal pixels have an approximately symmetric or moderately skewed distribution of intensity within their close neighborhood which is translated into the smooth change of intensity values.

The mean values of feature 13 for normal and abnormal pixels, shown in Fig. A.3, are around 0.05 and 0.09 respectively which are relatively close to each other.

From the boxplot of feature 13 for normal and abnormal pixels, that is shown in Fig. A.15-b, one can see that the median value for normal and abnormal pixels is around 0.05 and 0.09 respectively. Moreover, the boxplots of both normal and abnormal pixels are symmetrical meaning that there exists no skewness.

Regarding to feature 13, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between the normal and abnormal pixels with respect to location, median, mean but their best fit distribution family are different. Moreover, normal pixels have higher variation than abnormal ones.

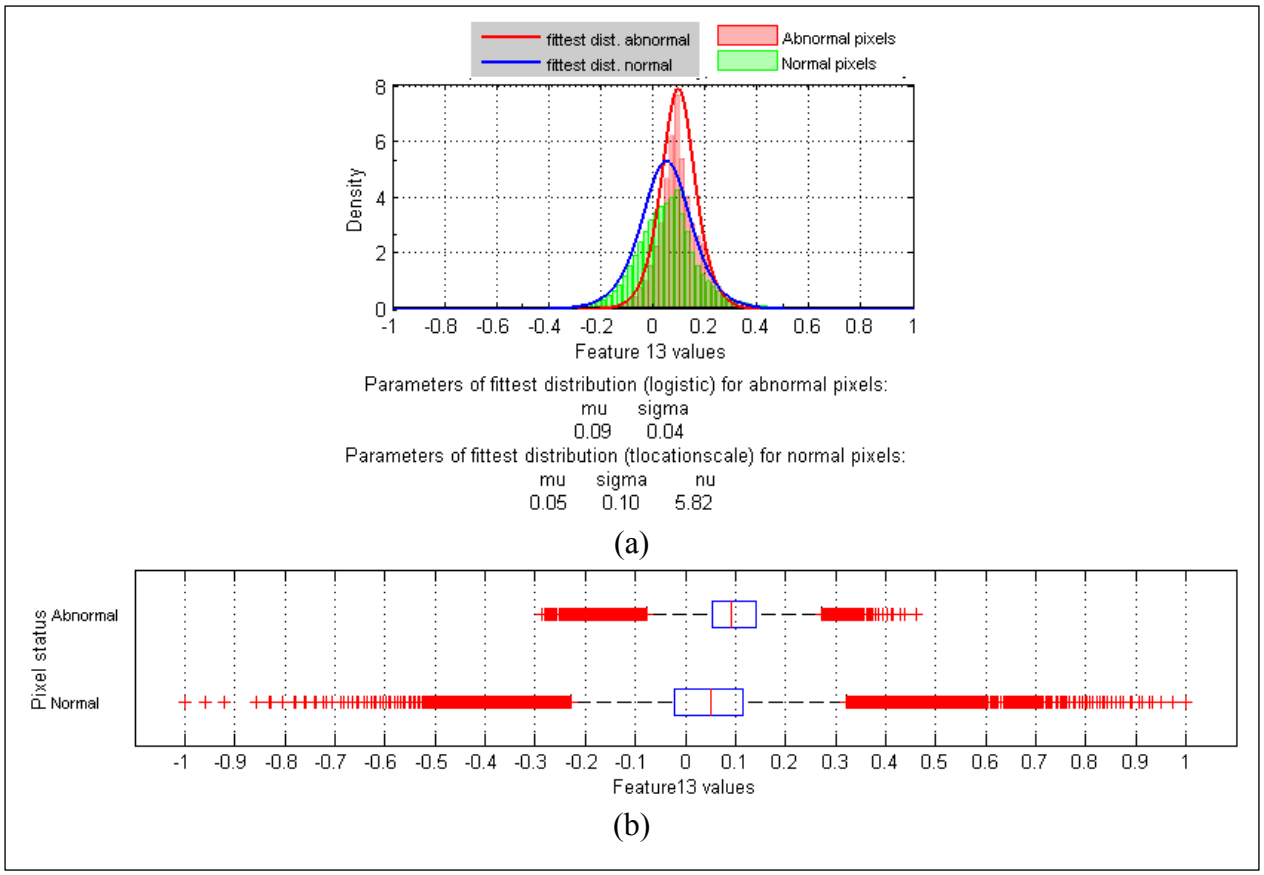


Fig. A.15: (a) bi-histogram of features 13; (b) box plot of features 13.

Kurtosis

Given pixel $P(x, y)$, the 14th feature in our dataset wants to quantify to what extent the shape of the distribution of the intensity values in a 15×15 neighborhood area of $P(x, y)$ matches the Gaussian distribution. A Gaussian distribution has a kurtosis of 3. A flatter distribution has a kurtosis less than 3 while a distribution that is more peaked than a Gaussian distribution has a kurtosis value greater than 3. The smallest possible value for kurtosis is 1, and the largest possible value is ∞.

From the bi-histogram of the 14th feature which is shown in Fig. A.16-a, we can see that the best distribution fit for both normal and abnormal pixels is type II generalized extreme value whose tails decrease as a polynomial ($k > 0$). Taking a closer look into the bi-histogram, we can see that a majority part of normal and abnormal pixels have their kurtosis value within range $[-1, -0.9]$; Having mapped range $[-1, -0.9]$ to its original scale, we have the result of $kurtosis \in [1, 6.27]$.

The mean values of feature 14 for normal and abnormal pixels, shown in Fig. A.3, are around -0.93 and -0.95 respectively which are very close to each other.

From the boxplot of feature 14, that is shown in Fig. A.16-b, one can see that the median value for normal and abnormal pixels is around -0.95 and -0.96 respectively. Moreover, for both normal and abnormal pixels, $|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) < length(upper\ whisker)$. As a result, the distribution of both normal and abnormal pixels is right skewed.

Regarding to Feature 14, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between the normal and abnormal pixels with respect to distribution, location, median, mean but the variation of normal pixels is bigger than the abnormal ones.

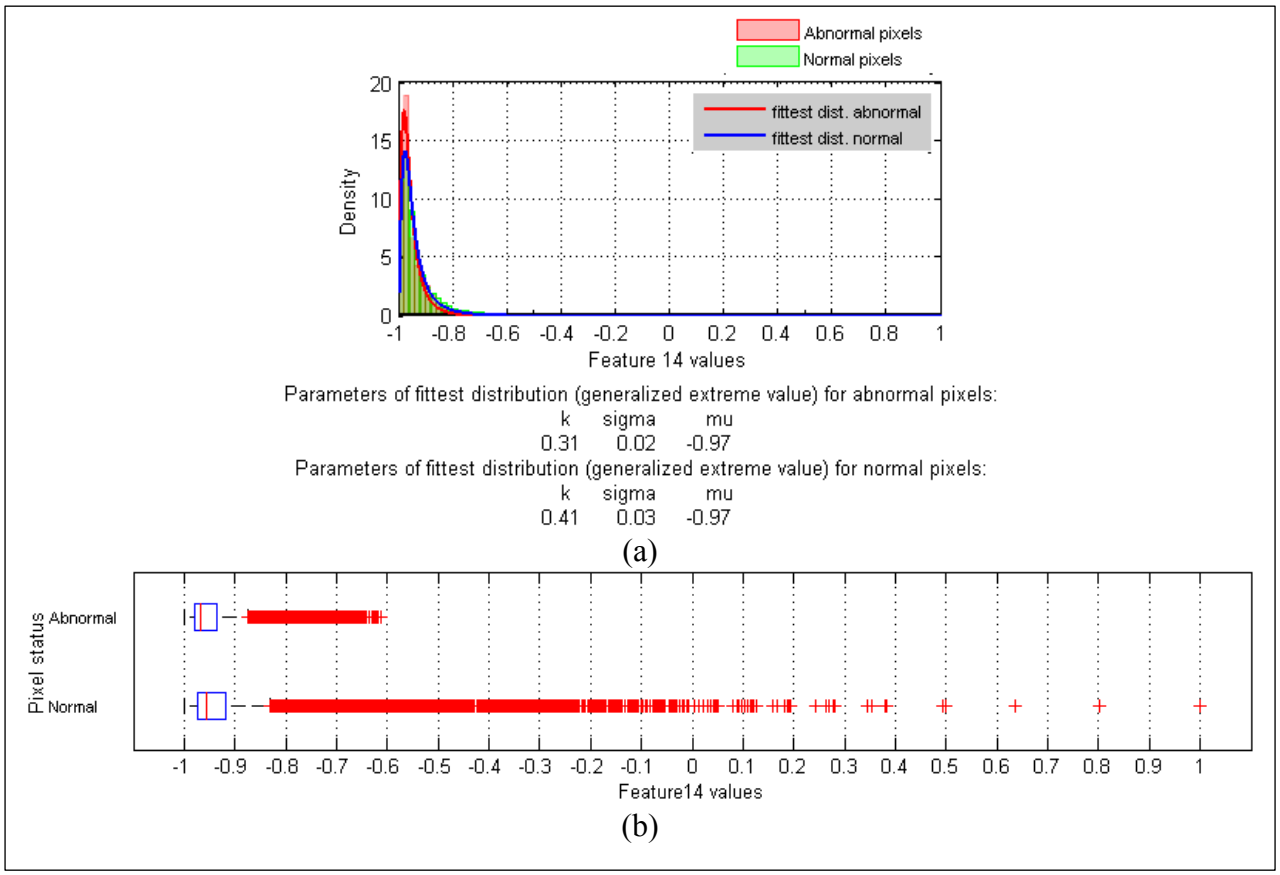


Fig. A.16: (a) bi-histogram of features 14; (b) box plot of features 14.

Energy

Given pixel $P(x, y)$, the 15th feature in our dataset wants to quantify the degree of intensity uniformity in a 15×15 neighborhood area of $P(x, y)$.

From the bi-histogram of the 15th feature, shown in Fig. A.17-a, we can see that both normal and abnormal pixels are centered at value -0.2. Moreover, the best distribution fit for both subgroups is type III generalized extreme value whose tails are finite ($k < 0$).

From the boxplot of Feature 15 that is shown in Fig. A.17-b, one can see that the median value for normal and abnormal pixels are around -0.15 and -0.2 respectively. Furthermore, for both normal and abnormal pixels, $|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) < length(upper\ whisker)$. As a result, the distribution of both normal and abnormal pixels is right skewed. From Fig. A.3 it can be seen that the mean value of feature 15 for normal and abnormal pixels are around -0.1 and -0.17 respectively. As it is shown in bi-histogram and boxplot, the spread of normal pixels is more than the abnormal pixels.

Regarding to feature 15, the bi-histogram, boxplot and the mean plot reveal that there is a difference between normal and abnormal pixels with respect to variation, median and mean values but their fittest distribution and location, are equal.

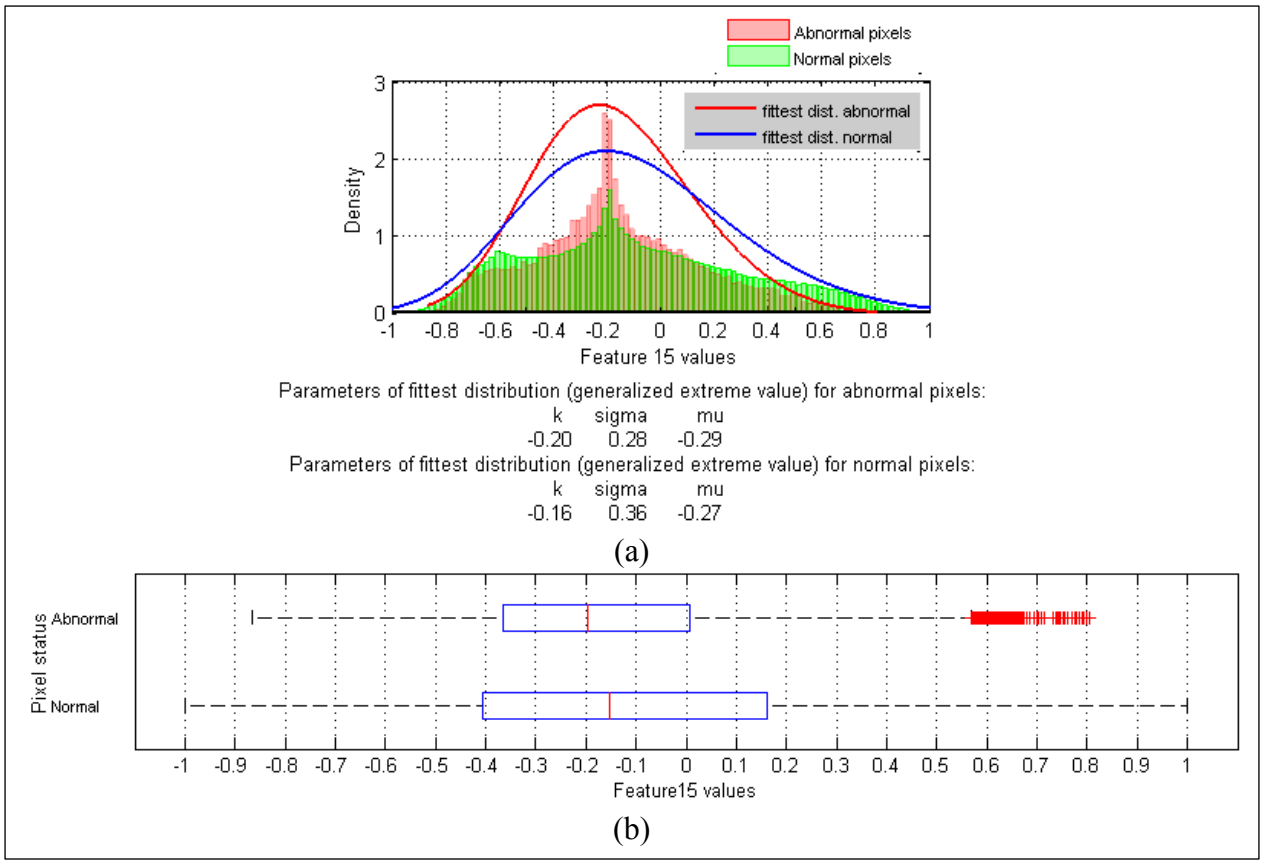


Fig. A.17: (a) bi-histogram of features 15; (b) box plot of features 15.

Entropy

Given pixel $P(x, y)$, the 16th feature in our dataset wants to quantify the degree of intensity disorder in a 15×15 neighborhood area of $P(x, y)$.

From the bi-histogram of the 16th feature, shown in Fig. A.18-a, we can see that both normal and abnormal pixels are approximately centered at value -0.3. Moreover, the best distribution fit for both subgroups is type III generalized extreme value whose tails are finite ($k < 0$).

From the boxplot of Feature 16 for normal and abnormal pixels, that is shown in Fig. A.18-b, one can see that the median value for both normal and abnormal pixels is around -0.2. Furthermore, for both normal and abnormal pixels, $|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) < length(upper\ whisker)$. As a result, the distribution of both normal and abnormal pixels is right skewed. From Fig. A.3 it can be seen that the mean values of feature 16 for normal and abnormal pixels are around -0.15 and -0.13 respectively which are relatively close to each other.

As it is shown in the boxplot, the spread of normal pixels is more than abnormal ones.

Regarding to feature 16, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between normal and abnormal pixels with respect to median, mean, location and their fittest distributional shape but their variation are different.

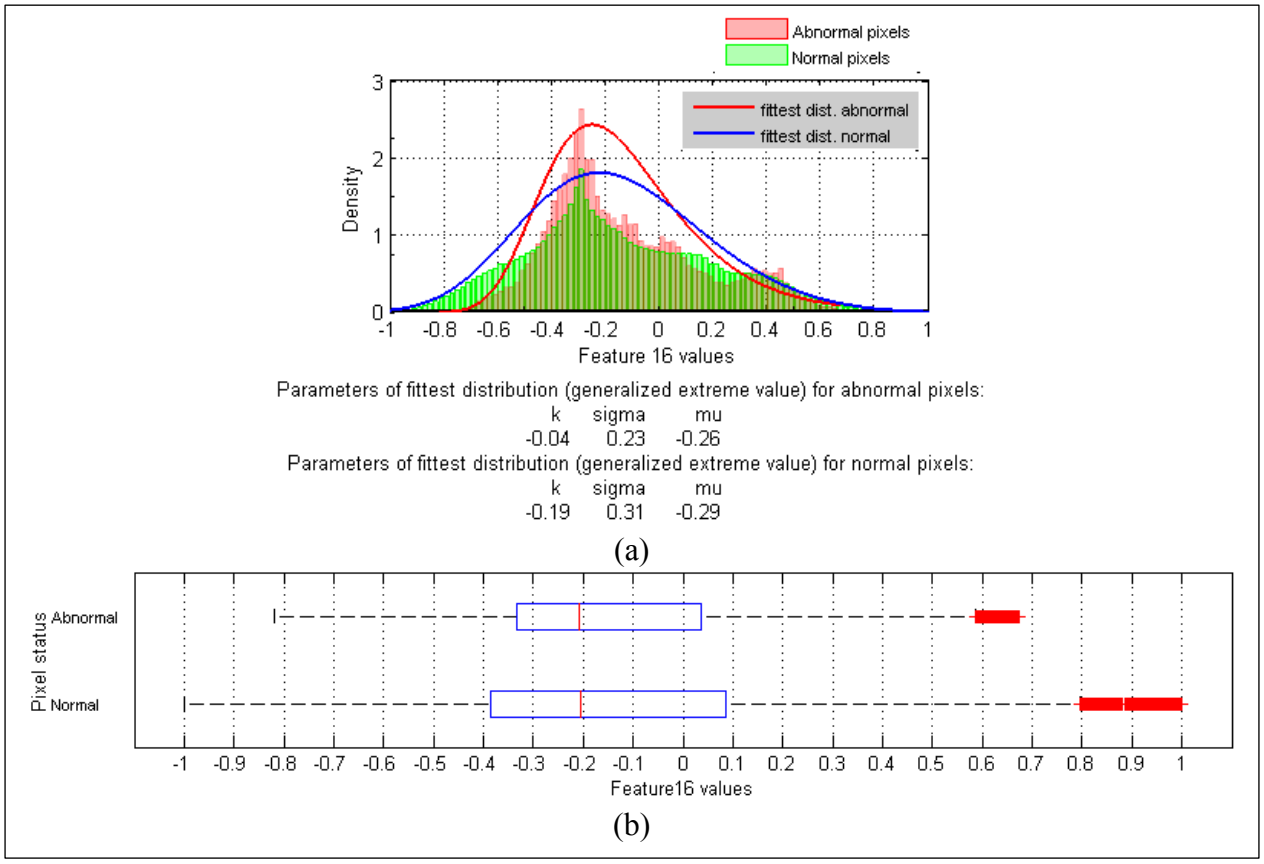


Fig. A.18: (a) bi-histogram of features 16; (b) box plot of features 16.

GLCM Autocorrelation

To calculate this feature, given window w of size 31×31 centered at pixel $P(x,y)$, we first obtained its corresponding GLCM matrix in 4 different directions using parameters $d = 1$ and $\theta = \{0,45,90,135\}$ and then made an average over them to obtain the final GLCM matrix direction invariant. Having the final GLCM matrix at hand, we used eq. (3.45) to calculate feature 17 value for pixel $P(x,y)$ which provides a measure of gray-tone linear-dependencies [115] between each pixel within window w and its immediate neighbors in 4 directions $0,45,90$ and 135.

From the bi-histogram of the 17th feature which is shown in Fig. A.19-a, we can see that normal and abnormal pixels are approximately centered at -0.2 and -0.5 respectively. That indicates that the two subgroups are displaced by about 0.3 units. Thus whether a pixel is normal or abnormal has an effect on the location for feature 17. Moreover, the best distribution fit for normal pixels is logistic which resembles the normal distribution in shape but has heavier tails. Abnormal pixels are also modeled by type III generalized extreme value whose tails are finite ($k < 0$).

From the boxplot of feature 17 that is shown in Fig. A.19-b, one can see that for abnormal pixels upper whisker is a bit longer than lower whisker. Moreover, $|q2 - q1| < |q3 - q2|$. As a result, abnormal distribution is a bit right skewed. On the other hand, the boxplot of normal pixels is approximately symmetric hence there exists no skewness. Also, the median value for normal and abnormal pixels are around -0.22 and -0.47 correspondingly. With respect to variation, the spread of the normal pixels is more than abnormal pixels. From Fig. A.3 it can be seen that the mean values of feature 17 for normal and abnormal pixels are around -0.23 and -0.44 respectively.

Regarding to Feature 17, the bi-histogram, boxplot and the mean plot reveal that there is a clear difference between normal and abnormal pixels with respect to median, mean, variation, location as well as their fittest distribution.

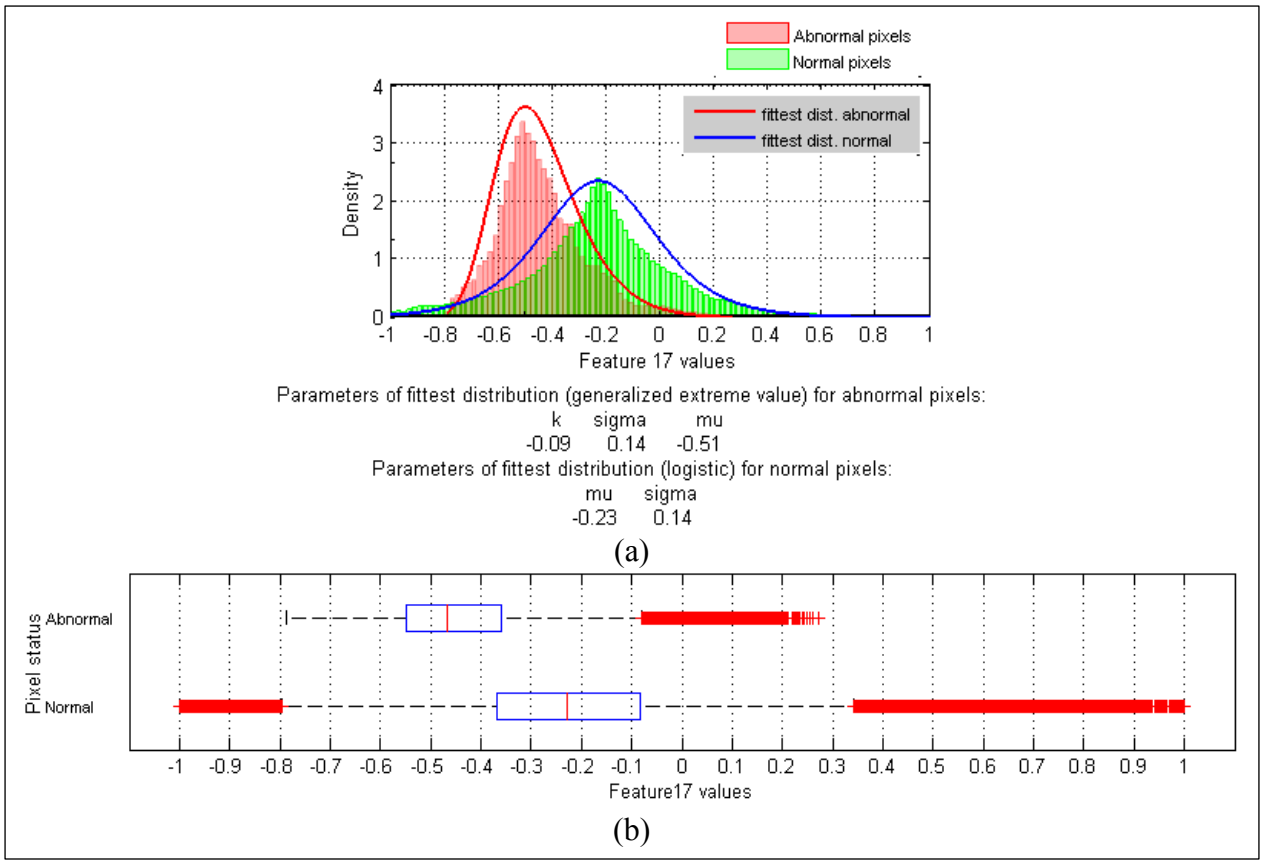


Fig. A.19: (a) bi-histogram of features 17; (b) box plot of features 17.

Correlation

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.29) to calculate correlation feature. Correlation also provides a measure of gray-tone linear-dependencies between each pixel within window w and its immediate neighbors in 4 directions $0,45,90\ and\ 135$. Correlation is 1 or -1 for a perfectly positively or negatively correlated image.

From the bi-histogram of the 18th feature which is shown in Fig. A.20-a, we can see that normal pixels are bimodal and have two peaks around 0.05 and 0.7. Abnormal pixels are centered at value 0.3. Moreover, the best distribution fit for both subgroups is type III generalized extreme value whose tails are finite ($k < 0$).

From the boxplot of feature 18, shown in Fig. A.20-b, one can see that the median value for normal and abnormal pixels are around 0.28 and 0.3 respectively. The boxplot shows left skewness for

both normal and abnormal pixels with a notice that the distribution of normal pixels is more skewed than abnormal pixels ($|q2 - q1| \cong |q3 - q2|$ and $length(lower\ whisker) > length(upper\ whisker)$). With respect to variation, the spread of the normal pixels is a bit more than the abnormal pixels. From Fig. A.3 it can be seen that the mean values of feature 18 for normal and abnormal pixels are around 0.27 and 0.28 respectively which are very close to each other.

Regarding to Feature 18, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between the normal and abnormal pixels with respect to median, mean and their fittest distributional model but their variation and location are different.

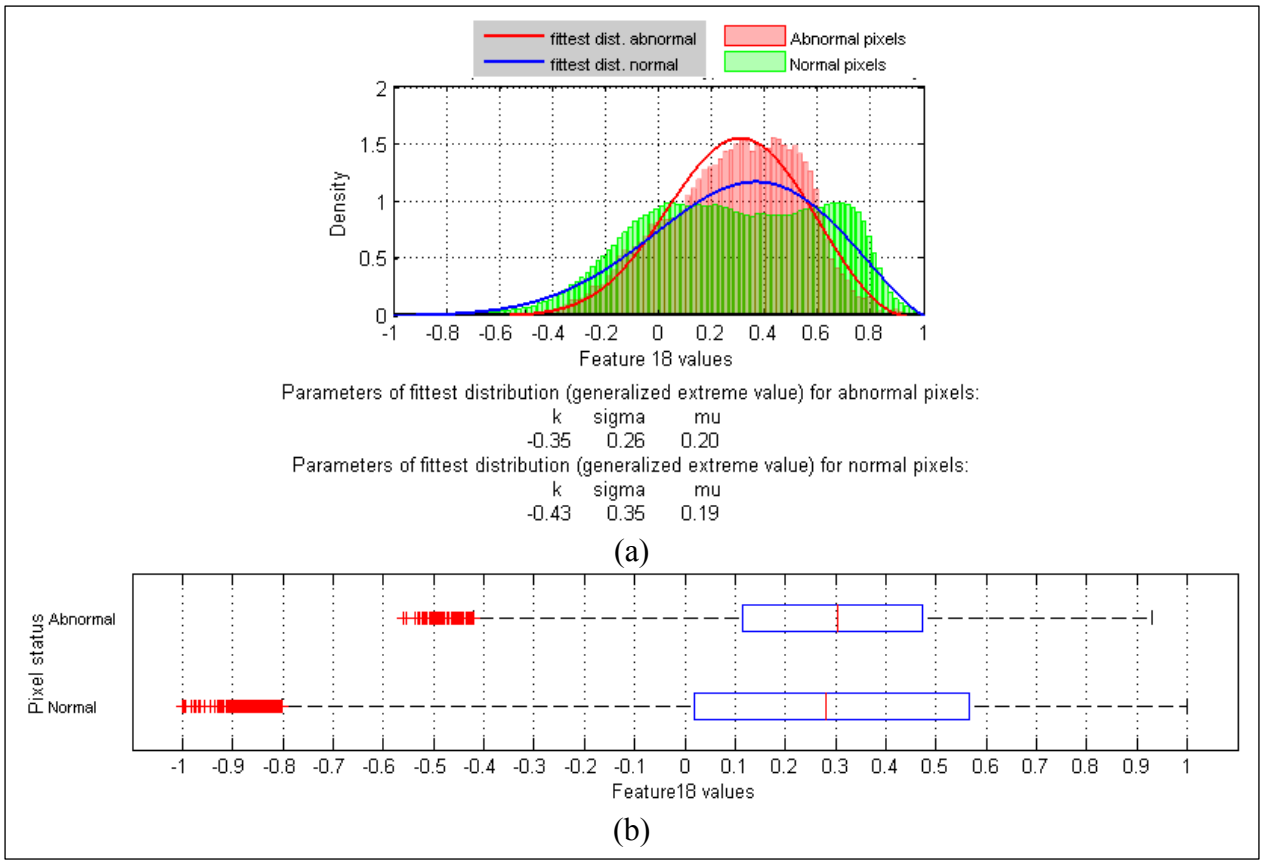


Fig. A.20: (a) bi-histogram of features 18; (b) box plot of features 18.

Prominence

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.26) to calculate prominence feature. Prominence provides a measure of

asymmetry within window w. The higher the prominence value is, the more asymmetric the image is. Moreover, a low prominence value indicates small variation in gray-scale [129].

From bi-histogram of feature 19, shown in Fig. A.21-a, we can see that the best distribution fit for both normal and abnormal pixels belongs to type II generalized extreme value whose tails decrease as a polynomial ($k > 0$). Moreover, both distributions are centered at -1. Summing up the frequencies within range $[-1, -0.9]$, we concluded that the prominence value of around 84% of normal pixels and 81% of abnormal pixels are overlapped and distributed within range $[-1, -0.9]$. Looking into the values before normalizing them within range $[-1,1]$ reveals the fact that the prominence value for 84% of normal pixels and 81% of abnormal pixels are around 25 and 11 respectively which are relatively small values. As a result, in most cases, regardless of whether $P(x, y)$ is normal or abnormal, there is a small variation of gray-scale values around a 15×15 neighborhood of $P(x, y)$.

From the boxplot of feature, shown in Fig. A.21-b, one can see that the median value for normal and abnormal pixels are both around -1. With respect to variation, the spread of the normal pixels is more than the abnormal pixels. From Fig. A.3 it can be seen that the mean values of feature 19 for normal and abnormal pixels are around -0.91 and -0.95 respectively which are very close to each other.

Regarding to Feature 19, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between normal and abnormal pixels with respect to median, mean, location and their distributional shapes but their variation are to some extent different.

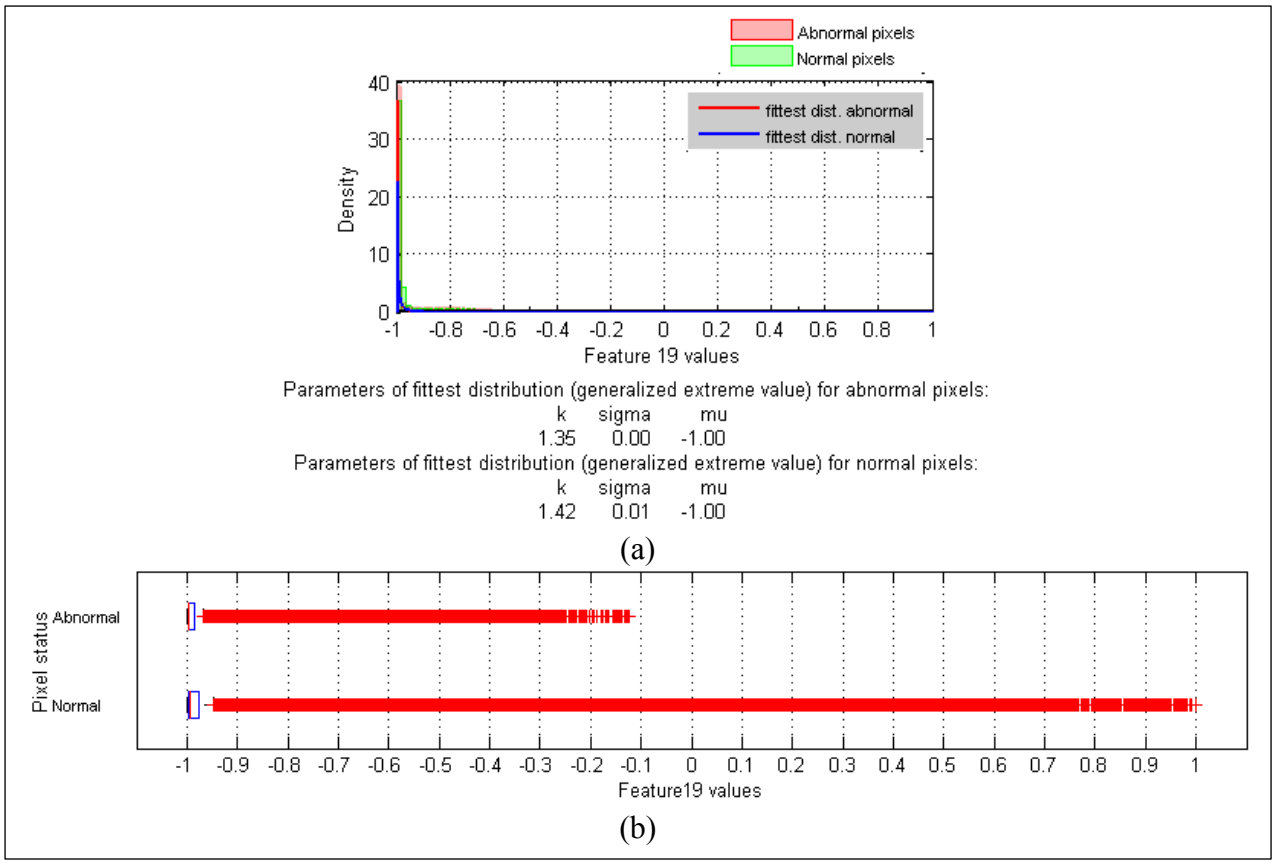


Fig. A.21: (a) bi-histogram of features 19; (b) box plot of features 19.

Shad

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.25) to calculate shade feature. This feature identifies the degree of gray-scale uniformity within window w. Like prominence, when the shade is high, the image is asymmetric [129].

From bi-histogram of feature 20 which is shown in Fig. A.22-a, we can see that the best distribution fit for both normal and abnormal pixels belongs to t location scale family. Moreover, both distributions are centered at 0.2. Summing up the frequencies within range $[0.2, 0.3]$, we concluded that the shade value of around 72% of normal pixels and 81% of abnormal pixels are overlapped and distributed within range $[0.2, 0.3]$. Looking into the values before normalizing them within range $[-1,1]$ reveals the fact that the shade value for 72% of normal pixels and 81% of abnormal pixels are around 3 and -0.3 respectively which are relatively small values. As a result, in most

cases, regardless of whether $P(x, y)$ is normal or abnormal, there is a small variation of gray-scale values around a 15×15 neighborhood of $P(x, y)$.

From the boxplot of feature 20, shown in Fig. A.22-b, one can see that the median value for normal and abnormal pixels are both around 0.21. With respect to variation, the spread of the normal pixels is more than the abnormal pixels. From Fig. A.3 it can be seen that the mean values of feature 20 for normal and abnormal pixels are around 0.17 and 0.18 respectively which are very close to each other.

Regarding to Feature 20, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between normal and abnormal pixels with respect to median, mean, location and their distributional shapes but their variation are to some extent different.

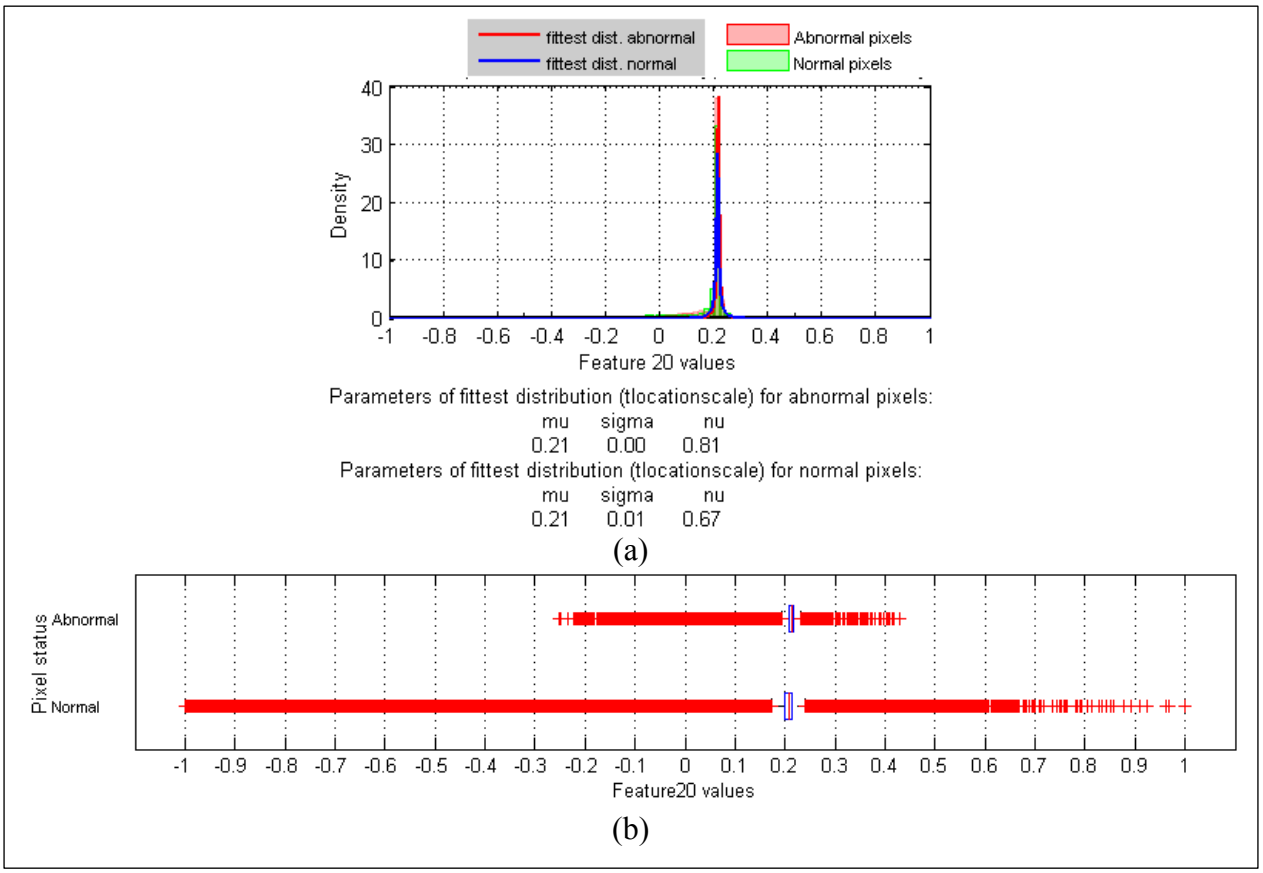


Fig. A.22: (a) bi-histogram of features 20; (b) box plot of features 20.

Dissimilarity

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.35) to calculate dissimilarity feature. In Dissimilarity the weights with which GLCM probabilities are multiplied increase linearly away from the diagonal (along which neighboring values are equal). In other words, when dissimilarity value of window w is high, it means that there exist a considerable amount of pixels within this window whose intensity values are quite different from their immediate neighbors (i.e., $d = 1$).

From the bi-histogram of feature 21 which is shown in Fig. A.23-a, we can see that the best distribution fit for normal pixels is type III generalized extreme value whose tails are finite ($k < 0$) while abnormal pixels are modeled by type II generalized extreme value whose tails decrease as a polynomial ($k > 0$). Moreover, the most frequent value for normal and abnormal pixels is -0.5 and -0.6 respectively.

From the boxplot of feature 21, shown in Fig. A.23-b, one can see that the median value for both normal and abnormal pixels is around -0.44. Furthermore, for both normal and abnormal pixels, $|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) < length(upper\ whisker)$. As a result, the distribution of both normal and abnormal pixels is right skewed which means that regardless of whether pixel $P(x, y)$ is normal or abnormal, only few data samples are centered in a window whose dissimilarity value is relatively big. In the case of having big dissimilarity, it is probable that pixel $P(x, y)$ is located in a boundary region and the window w is covering two different types of tissue (e.g., $P(x, y)$ is located in the boundary of ventricle and gray matter or is located in the boundary of a lesion). With respect to variation, the spread of normal pixels is bigger than abnormal pixels. The mean values of feature 21 for normal and abnormal pixels, shown in Fig. A.3, are around -0.40 and -0.39 respectively which are very close to each other.

Regarding to Feature 21, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between the normal and abnormal pixels with respect to mean and median values but their distributional shape, spread and centers are different from each other.

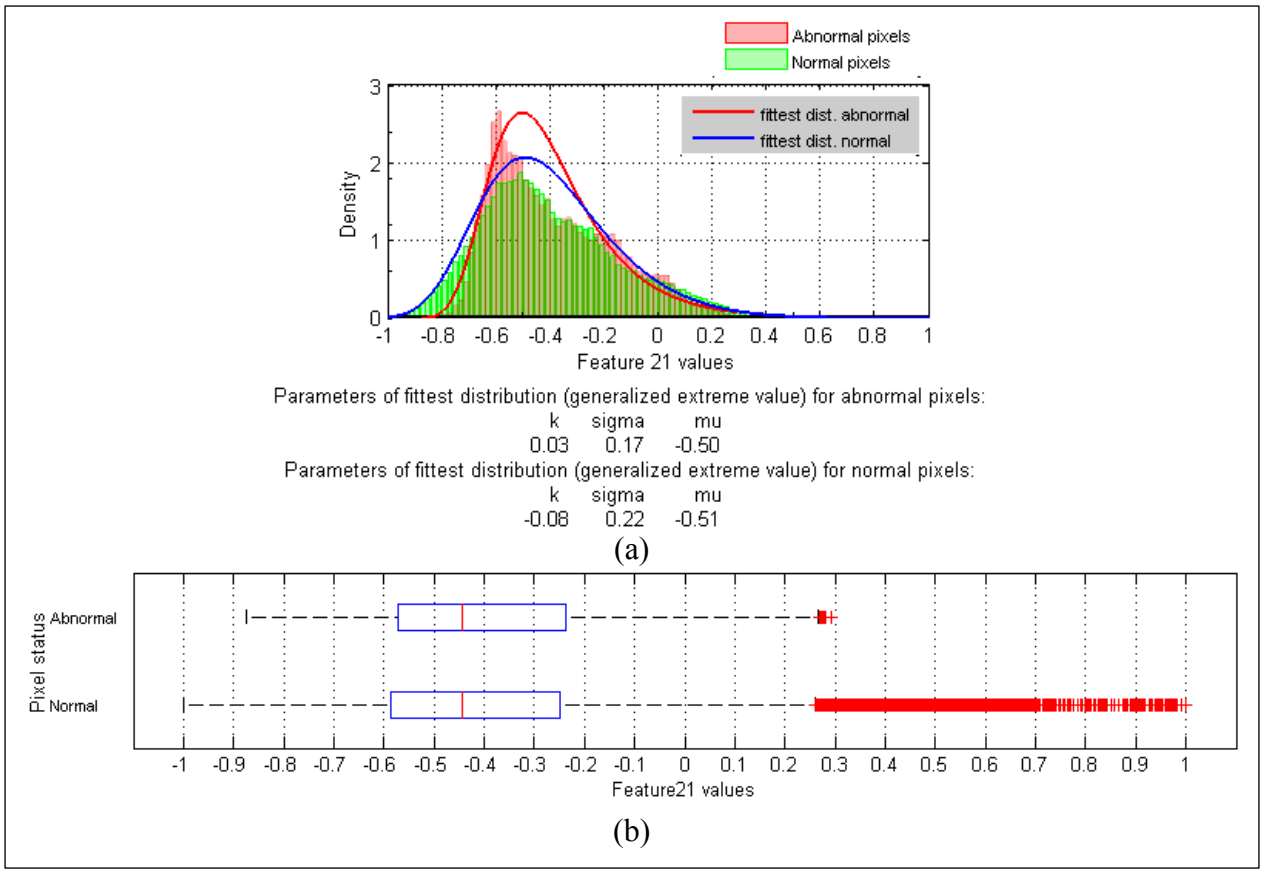


Fig. A.23: (a) bi-histogram of features 21; (b) box plot of features 21.

GLCM energy

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.21) to calculate energy or angular second moment feature. As mentioned before, the energy of a texture describes the uniformity of the texture. If window w has a homogeneous texture or pixels are very similar, the co-occurrence matrix has fewer entries of large magnitude and this lead to a larger energy feature.

From the bi-histogram of feature 22 which is shown in Fig. A.24-a, we can see that the best distribution fit for normal pixels is type II generalized extreme value whose tails decrease as a polynomial ($k > 0$) while abnormal pixels are modeled by type III generalized extreme value whose tails are finite ($k < 0$). Moreover, the most frequent value for both normal and abnormal pixels is around -0.45.

Considering the boxplot that is shown in Fig. A.24-b, the median value for normal and abnormal pixels are around -0.34 and -0.37 respectively. Moreover, for both normal and abnormal pixels, $|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) < length(upper\ whisker)$. As a result, the distribution of both normal and abnormal pixels is right skewed which means that regardless of whether pixel $P(x, y)$ is normal or abnormal, only few data samples are centered in a window whose energy value is relatively big and as a result are quite homogeneous.

From the definition of energy and dissimilarity features, we expect to have an inverse relationship between dissimilarity and energy values of window w centered at pixel $P(x, y)$. There may be raised a question of why we do not have a left skewed histogram for energy, considering the fact that the dissimilarity histogram is right skewed. To answer this question, we plotted the dissimilarity values against their corresponding energy value in their original scales which is shown in Fig. A.25. The inverse relationship between these two features is clearly visible in this plot. Moreover, if we had a huge amount of data samples in the extreme points, the histogram skewness of energy and dissimilarity features would be in the opposite direction. As we can see in Fig. A.25, the majority of data samples have a middle-range value of both dissimilarity and energy feature and hence the skewness of their histograms is not in the opposite direction.

The mean values of feature 22 for normal and abnormal pixels, shown in Fig. A.3, are around -0.24 and -0.32 respectively. With respect to variation, the spread of normal pixels is a little bit bigger than abnormal pixels.

Regarding to feature 22, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between the normal and abnormal pixels with respect to their location. Their fittest distributions belong to the different types of a same family. Their variation, mean and median values are also close to each other with a small difference.

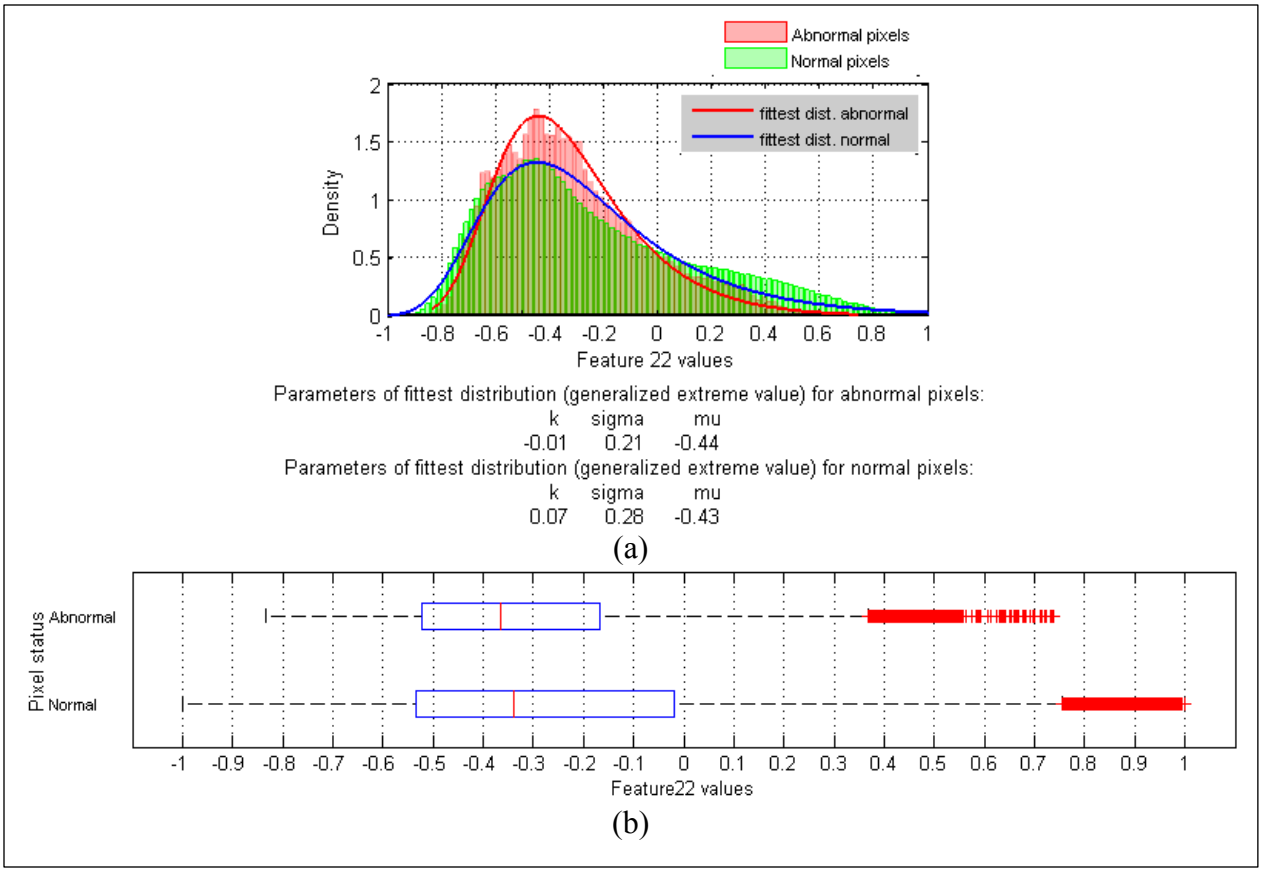


Fig. A.24: (a) bi-histogram of features 22; (b) box plot of features 22.

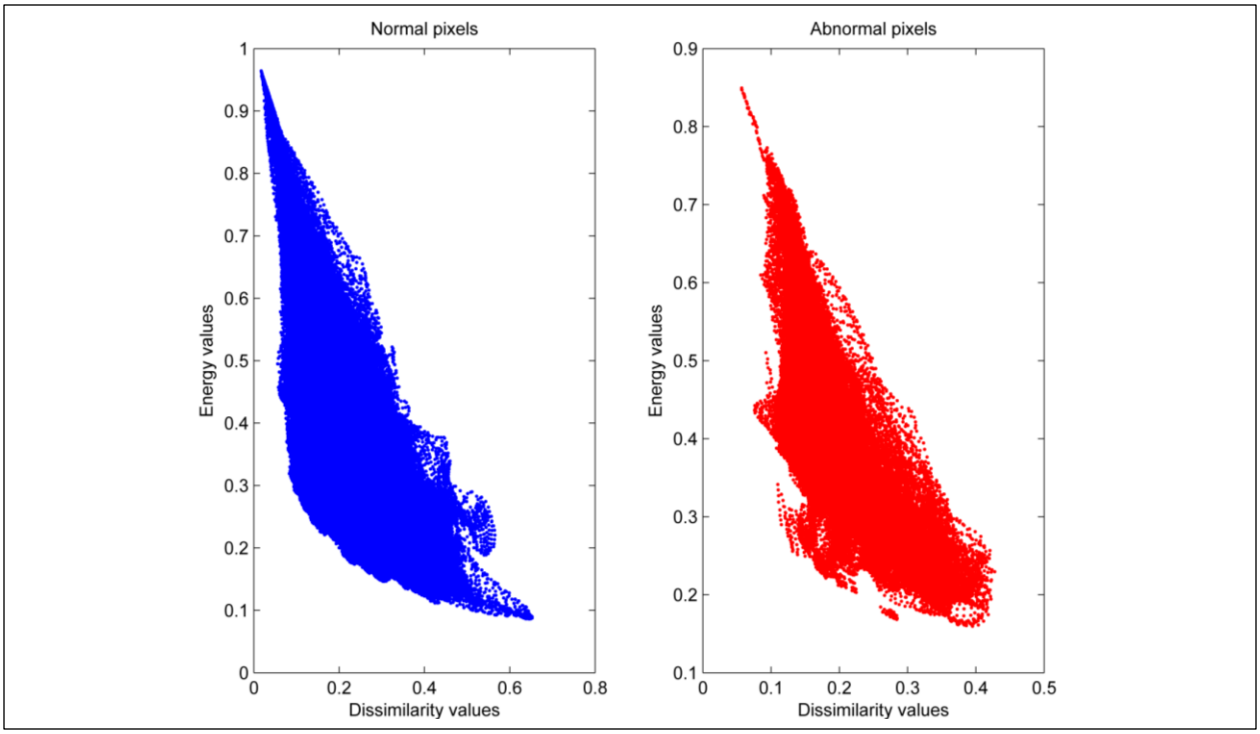


Fig. A.25 Visualizing inverse relationship between energy and dissimilarity features

GLCM entropy

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.22) to calculate the entropy. As mentioned before, the entropy of a texture describes the disorder or complexity of an image. Complex textures tend to have high entropy. Entropy is strongly, but inversely correlated to energy. This inverse relationship is shown in Fig. A.26. The values in Fig. A.26 are in their original scale.

From the bi-histogram of feature 23 which is shown in Fig. A.27-a, we can see that the best distribution fit for normal pixels is normal which is centered at -0.2. Abnormal pixels are modeled by type III generalized extreme value whose tails are finite ($k < 0$). Looking into the histogram of abnormal pixels, one can see that it has a multi-modal shape but the most frequent value is around -0.25.

From the boxplot of feature 23, that is shown in Fig. A.27-b, one can see that the median value for normal and abnormal pixels are around -0.15 and -0.18 respectively. Moreover, the boxplot of normal pixels is symmetric and shows no skewness. For abnormal pixels, although the length of upper and lower whiskers are equal, $|q2 - q1| < |q3 - q2|$. Hence, there the distribution of this subgroup has some skewness to the right. With respect to variation, the spread of normal pixels is more than the abnormal ones. From Fig. A.3 it can be seen that the mean values of feature 23 for normal and abnormal pixels are around -0.13 and -0.12 respectively which are very close to each other.

Regarding to feature 23, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between normal and abnormal pixels with respect to median and mean values but their variation and location are to some extent different. Furthermore, their fittest distributions belong to different families.

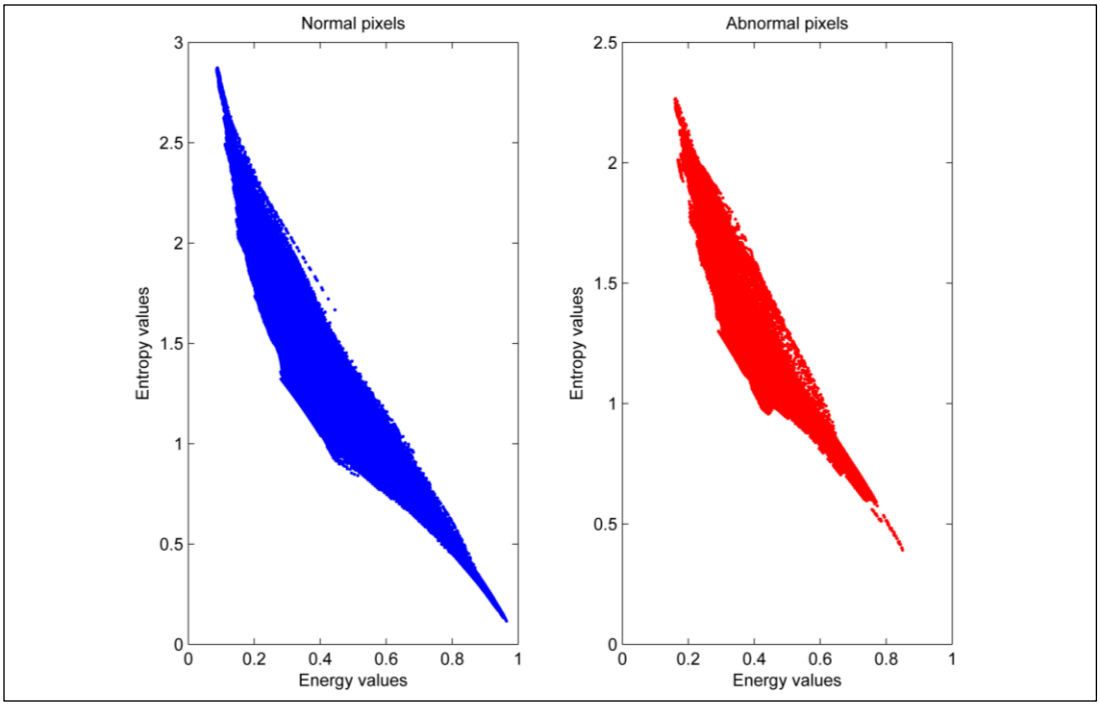


Fig. A.26 Visualizing inverse relationship between energy and entropy features

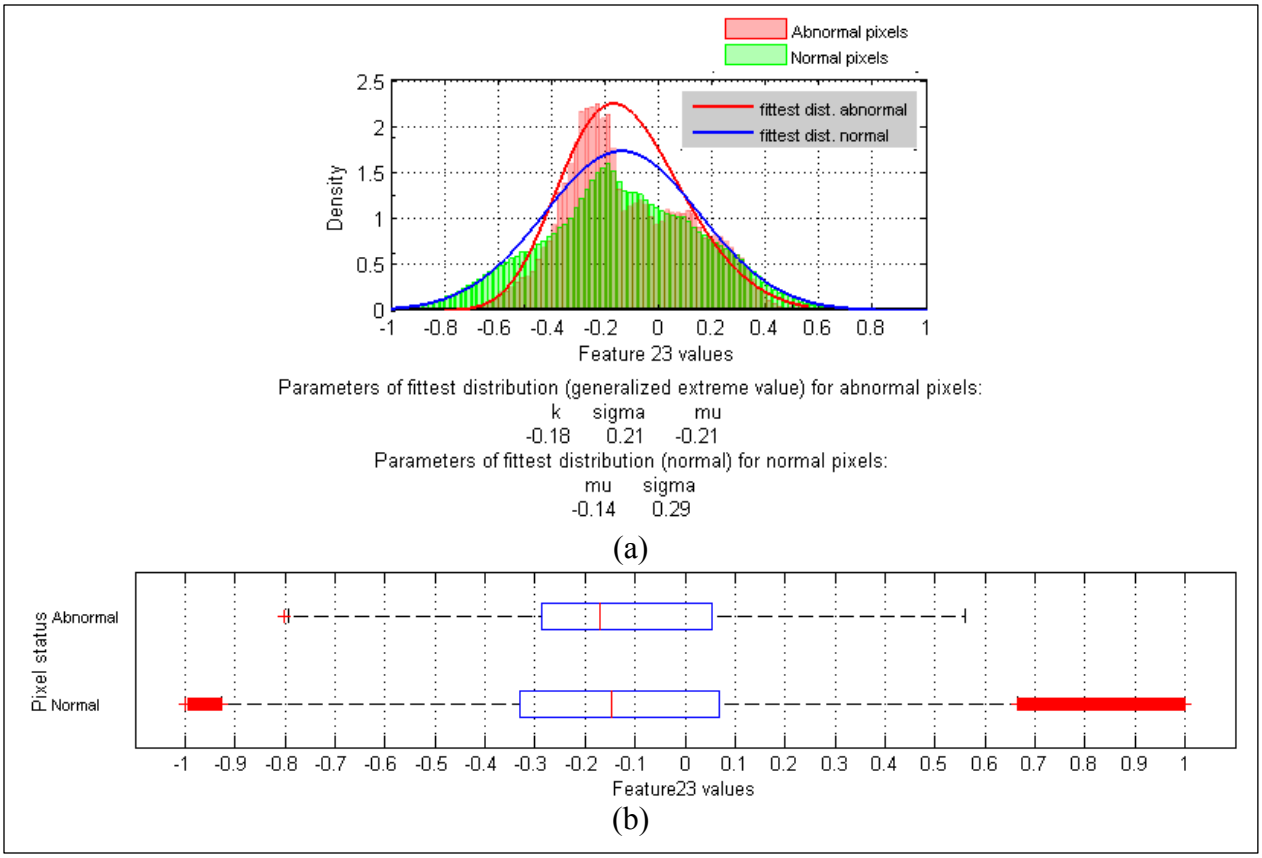


Fig. A.27: (a) bi-histogram of features 23; (b) box plot of features 23.

GLCM homogeneity

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.34) to calculate homogeneity. Homogeneity returns a value that measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal. Window w has a diagonal GLCM whenever each pixel within the window has its intensity value equal to the intensity value of its immediate neighbor (i.e., $d = 1$). The range of homogeneity feature is within $[0, 1]$. Window w has homogeneity of 1 whenever its GLCM is diagonal.

From the bi-histogram of feature 24 which is shown in Fig. A.28-a, we can see that the best distribution fit for both normal and abnormal pixels is type III generalized extreme value whose tails are finite ($k < 0$). Normal pixels are centered around 0.25 while abnormal pixels are peaked at 0.4.

From the boxplot of feature 24, that is shown in Fig. A.28-b, one can see that the median value for normal and abnormal pixels are around 0.21 and 0.22 respectively. With respect to the skewness, the boxplot of normal pixels is approximately symmetric. For abnormal pixels, the length of lower and upper whiskers are approximately equal but $|q2 - q1| < |q3 - q2|$. As a result, there exists some skewness to the right. Moreover, the spread of the normal pixels is more than the abnormal ones. From Fig. A.3 it can be seen that the mean value of feature 24 for normal and abnormal pixels is around 0.21 and 0.20 respectively which are very close to each other.

Regarding to feature 24, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between the normal and abnormal pixels with respect to mean and median values but they differ in their distributional shape, most frequent value as well as the variation.

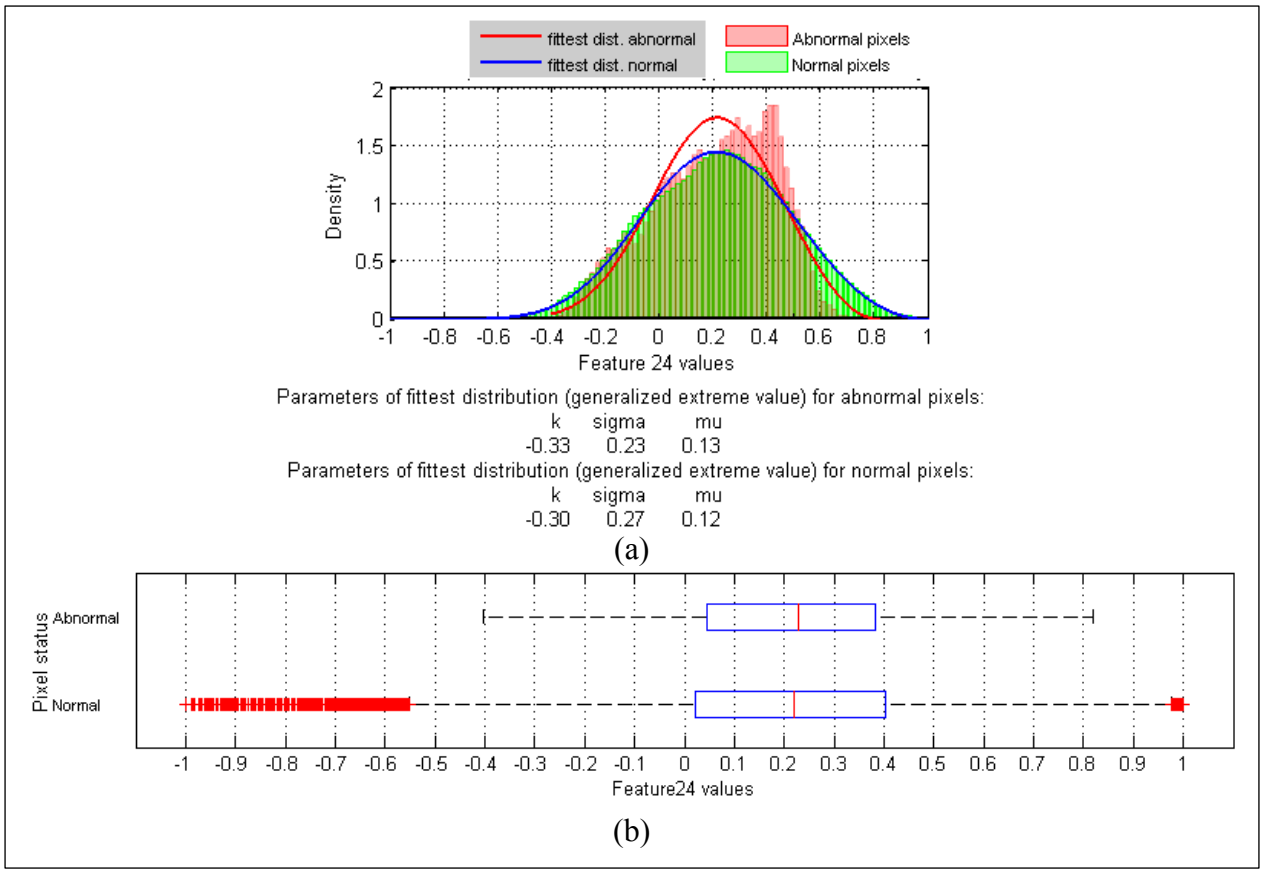


Fig. A.28: (a) bi-histogram of features 24; (b) box plot of features 24.

GLCM inverse difference moment

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.23) to calculate inverse difference moment feature. Comparing eq. (3.23) with eq. (3.34), one can see that inverse difference moment and homogeneity are quite similar to each other. The only difference between these two features is that inverse difference moment is inversely proportional to $(i-j)^2$ while homogeneity is inversely proportional to $|i-j|$. That is why that the bi-histogram and boxplot of feature 25, shown in Figs. A.29-a and A.29-b are also quite similar to the ones corresponding to feature 24.

Regarding to feature 25, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between normal and abnormal pixels with respect to mean and median values but their distributional shape, location and variation are different.

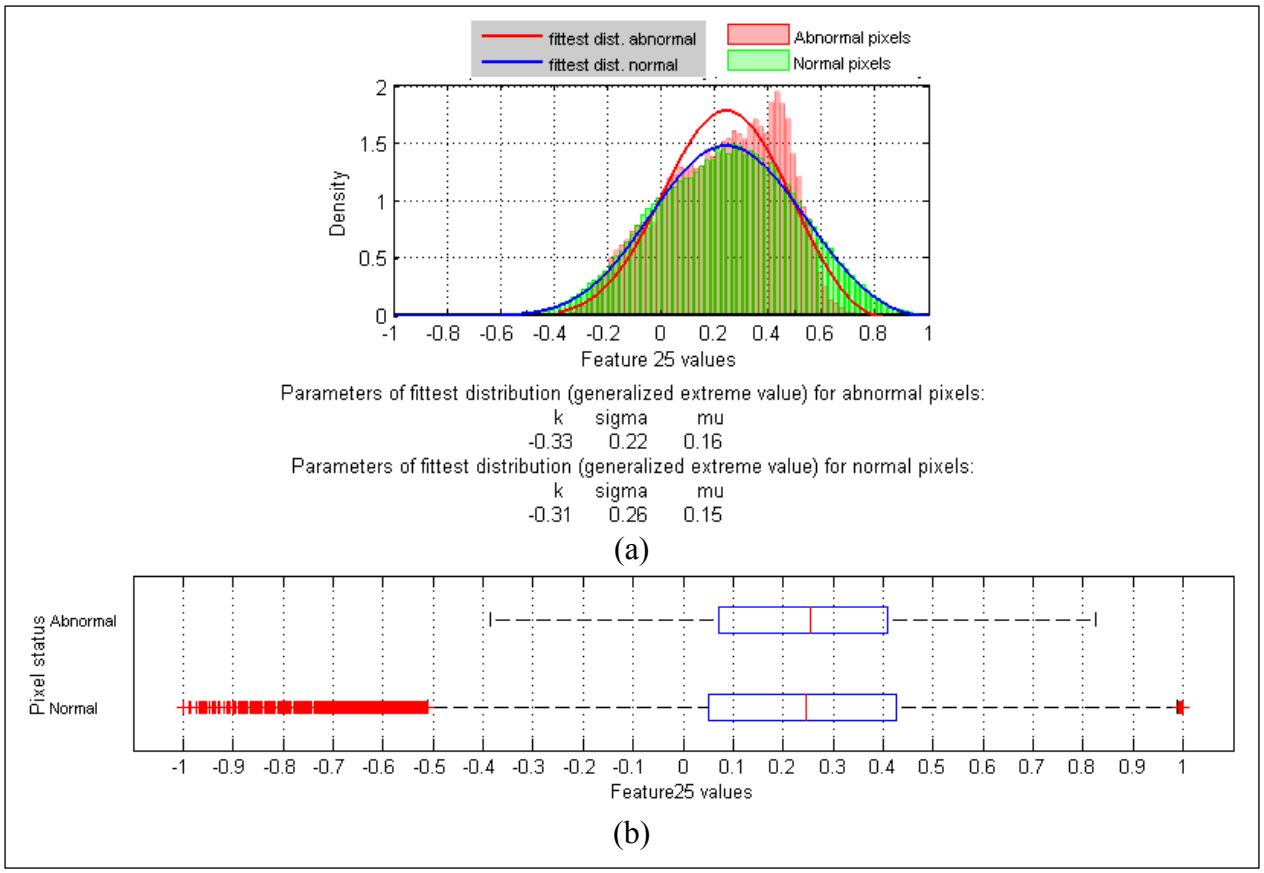


Fig. A.29: (a) bi-histogram of features 25; (b) box plot of features 25.

GLCM maximum probability

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.46) to calculate maximum probability feature. This feature extracts the probability value of the most frequent difference between gray levels of adjacent pixel pairs within window w. Maximum probability is expected to be high if the occurrence of the most predominant pixel pairs is high. As stated in [116] maximum probability plays a role similar to uniformity; the high values of this feature are usually associated with homogenous regions and the lower values with heterogeneous regions.

Looking into the bi-histogram of feature 26 which is shown in Fig. A.30-a, the best distribution fit for both normal and abnormal pixels is type III generalized extreme value whose tails are finite ($k < 0$). Moreover, both normal and abnormal pixels are centered around -0.3.

From the boxplot of feature 26 that is shown in Fig. A.30-b, one can see that the median value for normal and abnormal pixels are around -0.04 and -0.14 respectively. Moreover, for both normal and abnormal pixels, we have $|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) < length(upper\ whisker)$ which means both distributions are right skewed. With respect to variation, the spread of normal pixels is a bit more than the abnormal pixels. From Fig. A.3 it can be seen that the mean values of feature 26 for normal and abnormal pixels are around 0 and -0.08 respectively.

Regarding to Feature 26, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between the normal and abnormal pixels with respect to distributional shape and location but their variation, mean and median are to some extent different.

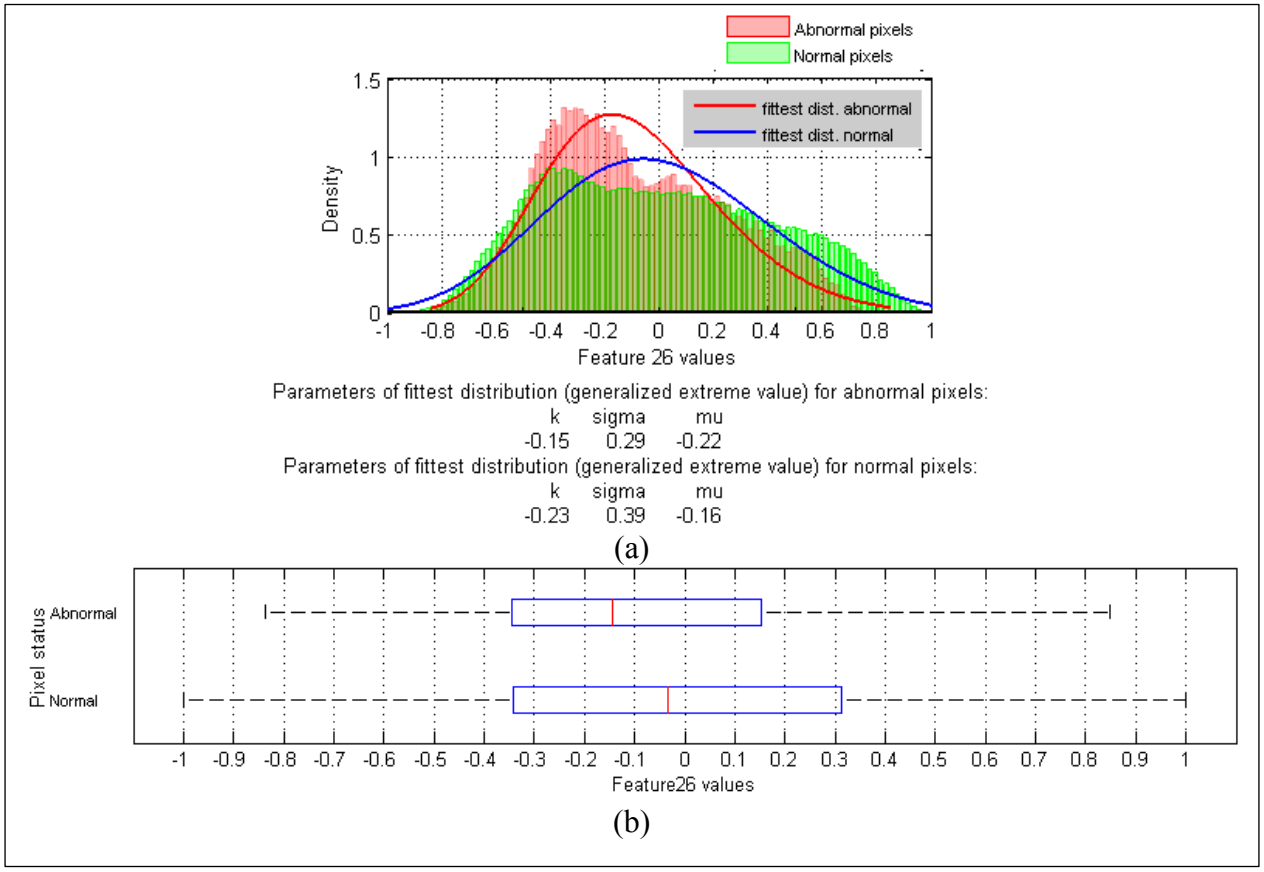


Fig. A.30: (a) bi-histogram of features 26; (b) box plot of features 26.

GLCM sum of squares

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.32) to calculate sum of squares feature.

From the bi-histogram of the 27^{th} feature which is shown in Fig. A.31-a, we can see that normal pixels are centered at a value of approximately -0.25 while abnormal pixels are centered at a value of approximately -0.55. That indicates that the two subgroups are displaced by about 0.3 units. Thus whether a pixel is normal or abnormal has an effect on the location for feature 27. Moreover, the best distribution fit for normal pixels is logistic which resembles the normal distribution in shape but has heavier tails. Abnormal pixels are modeled by type III generalized extreme value whose tails are finite ($k < 0$).

From the boxplot of feature 27 that is shown in Fig. A.31-b, one can see that the median value for normal and abnormal pixels is around -0.24 and -0.47 respectively. Furthermore, the boxplots of both normal and abnormal pixels shows some skewness to the right ($|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) < length(upper\ whisker)$). With respect to variation, the spread of normal pixels is more than the abnormal pixels. From Fig. A.3 it can be seen that the mean values of feature 27 for normal and abnormal pixels are around -0.23 and -0.44 respectively.

Regarding to Feature 27, the bi-histogram, boxplot and the mean plot reveal that there is a clear difference between normal and abnormal pixels with respect to median, mean, variation, location as well as their fittest distribution.

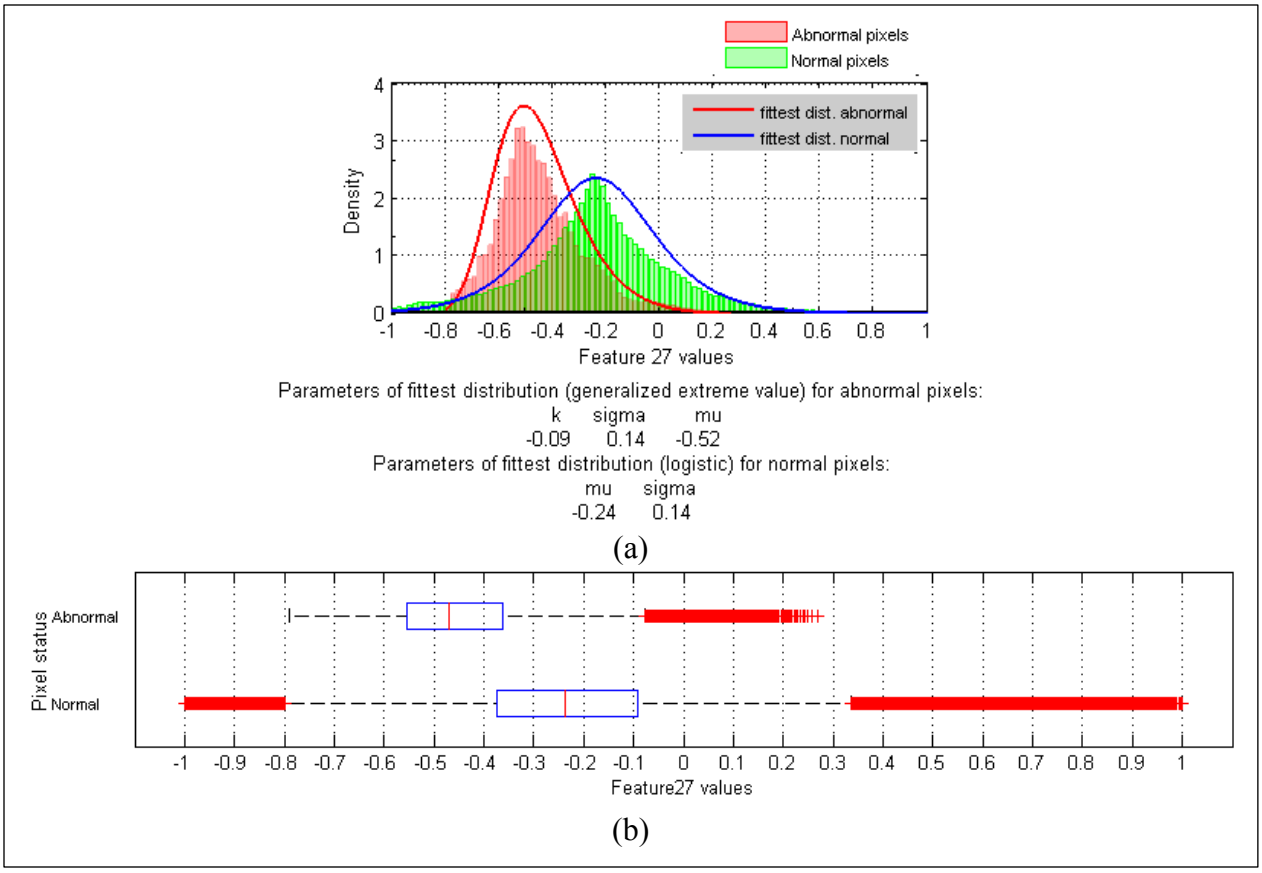


Fig. A.31: (a) bi-histogram of features 27; (b) box plot of features 27.

GLCM sum average

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.36) to calculate sum average feature.

From the bi-histogram of the 28th feature, shown in Fig. A.32-a, we can see that normal pixels are approximately centered at 0.1 while abnormal pixels are peaked at -0.25. That indicates that the two subgroups are displaced by about 0.35 units. The best distribution fit for normal pixels belongs to t location scale while abnormal pixels are modeled by type III generalized extreme value ($k < 0$).

From the boxplot of feature 28 that is shown in Fig. A.32-b, one can see that the median value for normal and abnormal pixels is around 0.05 and -0.22 respectively. Furthermore, for abnormal pixels, $|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) \cong length(upper\ whisker)$. As a result, the distribution of abnormal pixels is a bit right skewed. On the contrary, for normal pixels,

$|q2 - q1| > |q3 - q2|$ and $length(lower\ whisker) \cong length(upper\ whisker)$ which indicates that the corresponding distribution is a bit left skewed. With respect to variation, the spread of normal pixels is more than the abnormal pixels. From Fig. A.3 it can be seen that the mean values of feature 28 for normal and abnormal pixels are around 0 and -0.19 respectively.

Regarding to feature 28, the bi-histogram, boxplot and the mean plot reveal that there is a clear difference between normal and abnormal pixels with respect to median, mean, variation and location. Their fittest distributions also belong to different families.

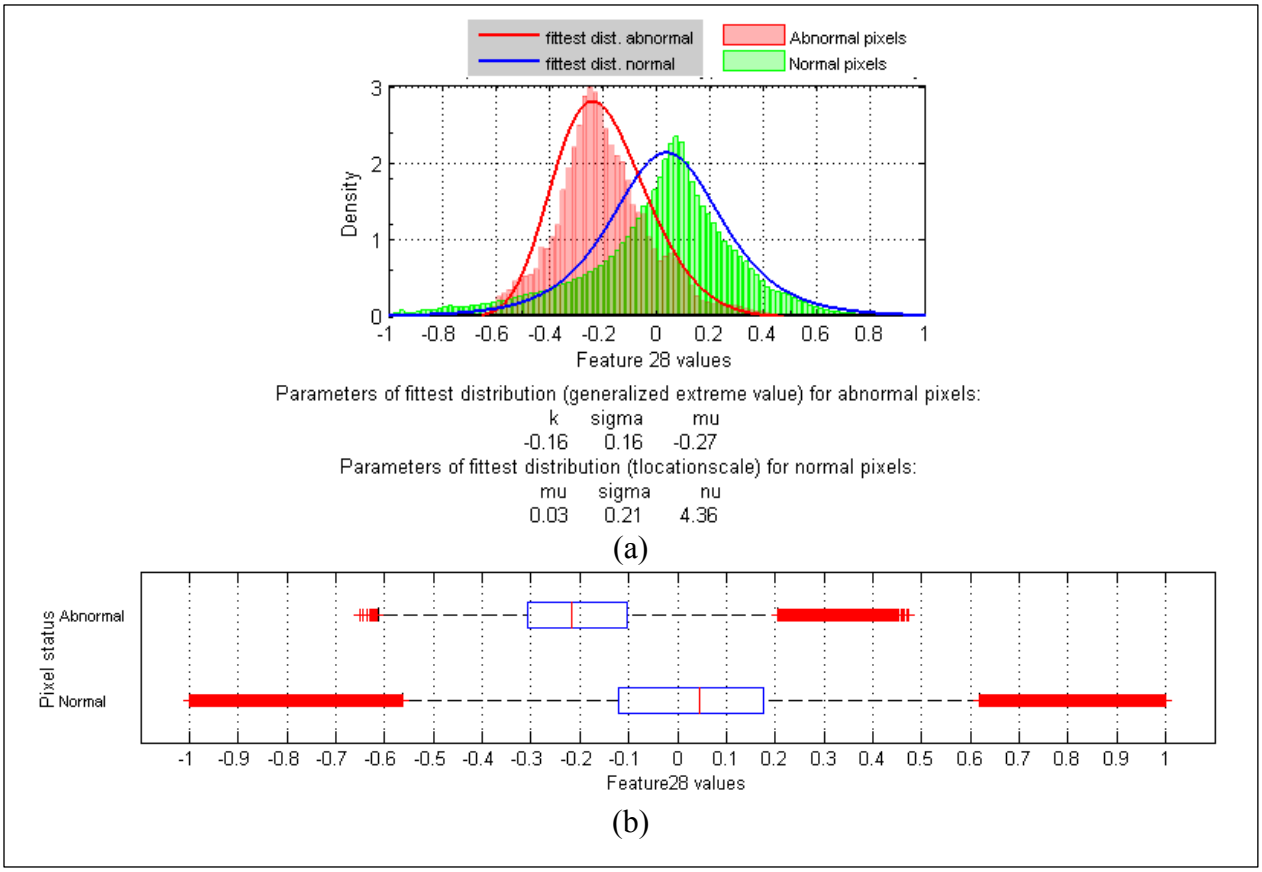


Fig. A.32: (a) bi-histogram of features 28; (b) box plot of features 28.

GLCM sum variance

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.38) to calculate sum variance feature.

From the bi-histogram of the 29th feature, shown in Fig. A.33-a, we can see that normal pixels are centered at 0 while abnormal pixels are peaked at -0.5. That indicates that the two subgroups are

displaced by about 0. 5 units. The best distribution fit for normal pixels belongs to t location scale family while abnormal pixels are modeled by type III generalized extreme value ($k < 0$).

From the boxplot of feature 29, shown in Fig. A.33-b, we can see that the median value for normal and abnormal pixels is around -0.18 and -0.49 respectively. Although for both normal and abnormal pixels the lengths of upper and lower whiskers are equal, we have $|q2 - q1| < |q3 - q2|$ for abnormal pixels and $|q2 - q1| > |q3 - q2|$ for normal pixels. As a result, the distribution of abnormal pixels is a bit right skewed while the distribution of normal ones is left skewed. With respect to variation, the spread of normal pixels is more than the abnormal ones. From Fig. A.3 it can be seen that the mean values of feature 29 for normal and abnormal pixels are around -0.21 and -0.45 respectively.

Regarding to feature 29, the bi-histogram, boxplot and the mean plot reveal that there is a clear difference between normal and abnormal pixels with respect to median, mean, variation, location as well as the best distribution fit.

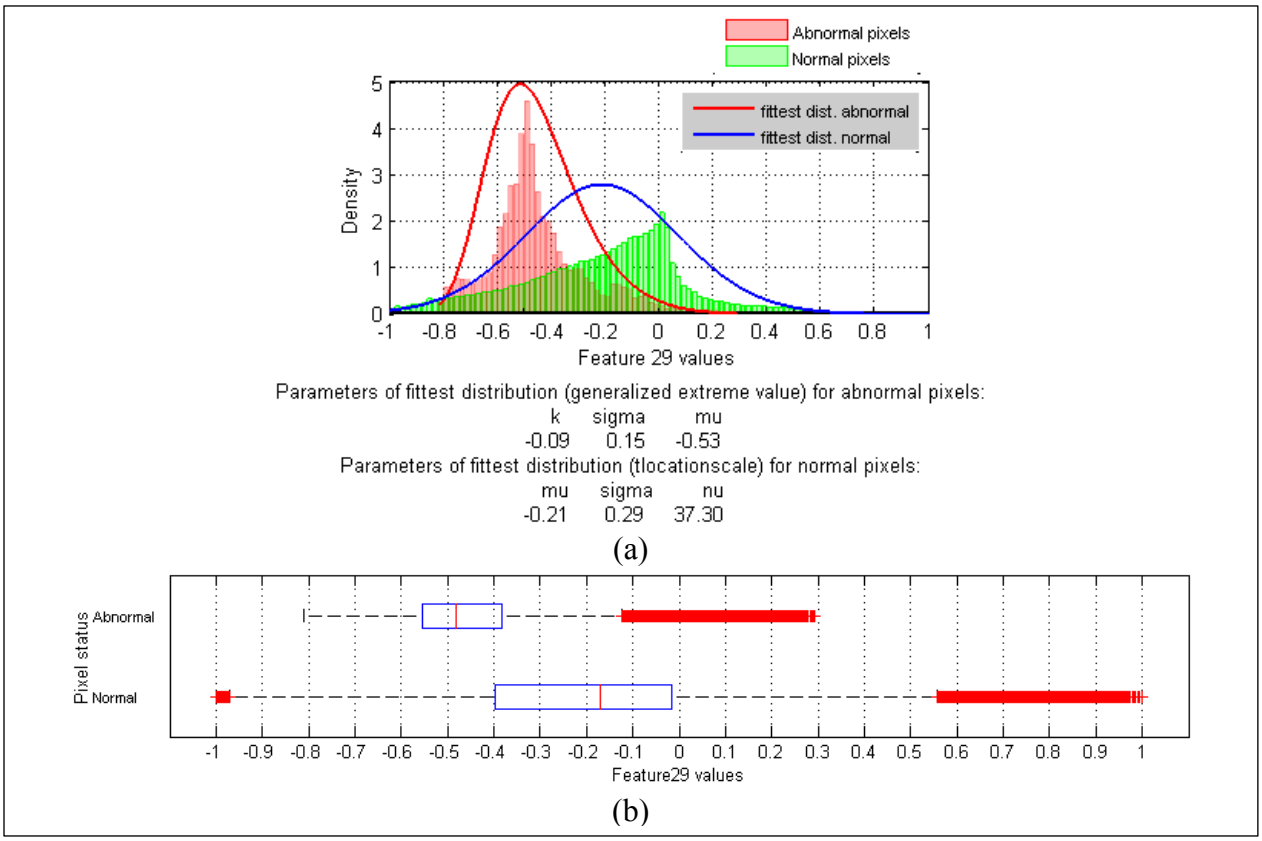


Fig. A.33: (a) bi-histogram of features 29; (b) box plot of features 29.

GLCM sum entropy

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.37) to calculate sum entropy feature. From the bi-histogram of the 30th feature which is shown in Fig. A.34-a, we can see that the best distribution fit for both normal and abnormal pixels belongs to type III generalized extreme value ($k < 0$). Moreover, both normal and abnormal pixels are approximately centered at -0.15.

From the boxplot of feature 30, shown in Fig. A.34-b, one can see that the median value for normal and abnormal pixels are around -0.08 and -0.1 respectively. Moreover, for normal pixels we have $|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) \cong length(upper\ whisker)$. As a result, there would be some skewness to the right. For abnormal pixels, $|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) > length(upper\ whisker)$. Hence we cannot say whether the distribution is left or right skewed from the corresponding boxplot. With respect to variation, the spread of normal pixels is more than the abnormal ones. From Fig. A.3 it can be seen that the average values of Feature 5 for normal and abnormal pixels are around -0.04 and -0.26 respectively.

Regarding to feature 30, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between normal and abnormal pixels with respect to median, location as well as the best distribution fit but their variation and mean values differ from each other.

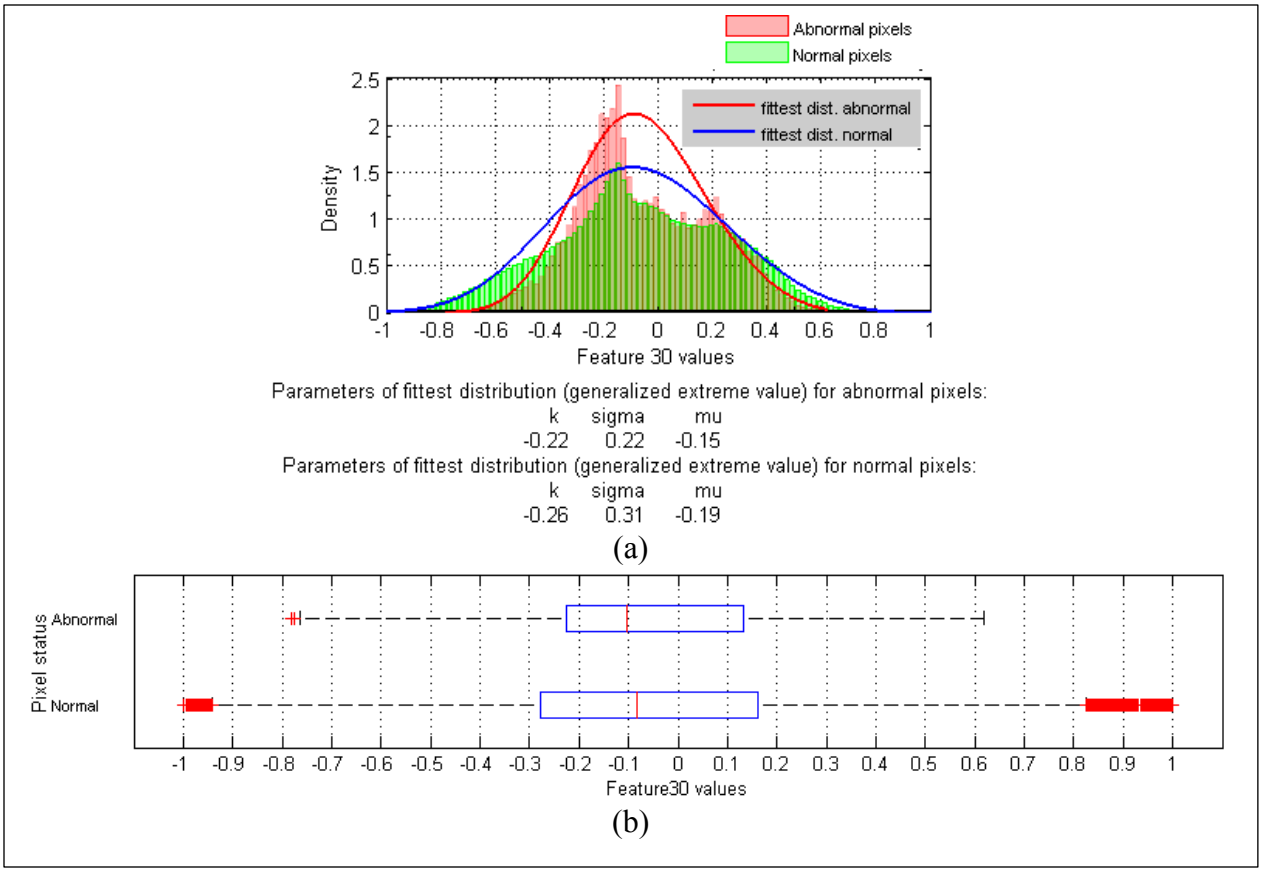


Fig. A.34: (a) bi-histogram of features 30; (b) box plot of features 30.

GLCM difference variance

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.39) to calculate difference variance feature.

From the bi-histogram of the 31th feature which is shown in Fig. A.35-a, we can see that the best distribution fit for both normal and abnormal pixels belongs to type II generalized extreme value whose tails decrease as a polynomial ($k > 0$). Moreover, the distribution of both normal and abnormal pixels seems to be bimodal and are centered at the same values -0.9 and -0.2 with a notice that the frequency at -0.9 is much higher than -0.2.

From the boxplot of feature31, shown in Fig. A.35-b, one can see that the median value for normal and abnormal pixels are around -0.87 and -0.86 respectively. Moreover, for both normal and abnormal pixels we have $|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) <$

$length(upper\ whisker)$. Hence, there would be some skewness to the right for both distributions. With respect to variation, the spread of normal pixels is more than abnormal pixels.

From Fig. A.3 it can be seen that the average values of feature 31 for normal and abnormal pixels are around -0.76 and -0.75 respectively.

Regarding to feature 31, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between normal and abnormal pixels with respect to median, mean location as well as the best distribution fit but their variation differ from each other.

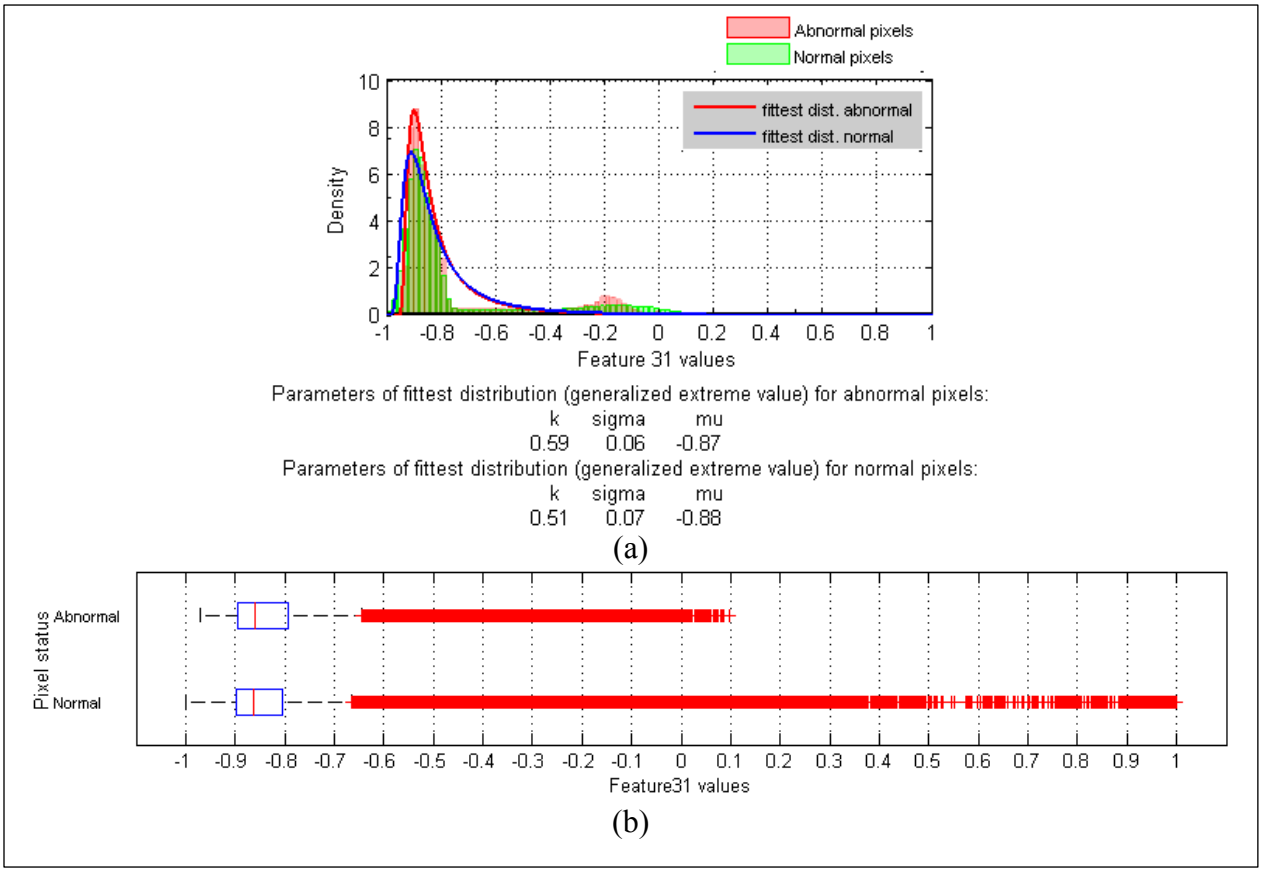


Fig. A.35: (a) bi-histogram of features 31; (b) box plot of features 31.

GLCM difference entropy

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.33) to calculate difference entropy feature.

From the bi-histogram of the 32nd feature which is shown in Fig. A.36-a, we can see that the best distribution fit for normal pixels is t location scale while abnormal pixels belong to type III

generalized extreme value ($k < 0$). Moreover, the distribution of normal pixels is centered around 0 while the most frequent value of abnormal pixels is around -0.3.

From the boxplot of feature 32, shown in Fig. A.36-b, one can see that the median value for both normal and abnormal pixels is around -0.08. Moreover, the boxplot of normal pixels seems to be symmetric meaning that there is no skewness in the corresponding distribution. For abnormal pixels we have $|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) \cong length(upper\ whisker)$. As a result, there would be some skewness to the right for the corresponding distribution. With respect to variation, the spread of normal pixels is more than the abnormal ones. From Fig. A.3 it can be seen that the average values of feature 32 for normal and abnormal pixels are around -0.08 and -0.06 respectively.

Regarding to feature 32, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between normal and abnormal pixels with respect to median and mean values but their variation, location and their best distribution fits differ from each other.

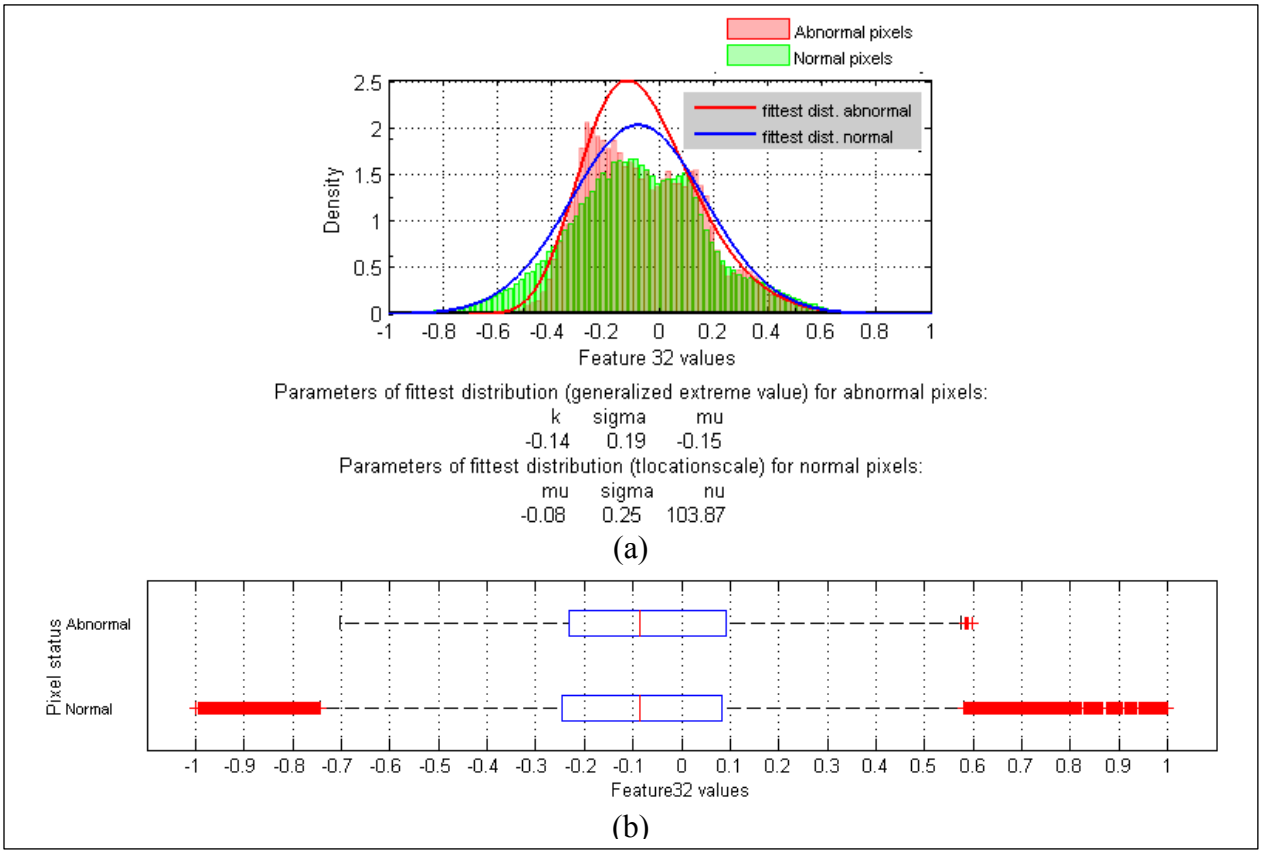


Fig. A.36: (a) bi-histogram of features 32; (b) box plot of features 32.

GLCM information measure of correlation1

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.40) to calculate information measure of correlation1 feature.

From the bi-histogram of the 33rd feature which is shown in Fig. A.37-a, we can see that the best distribution fit for both normal and abnormal pixels is type III generalized extreme value ($k < 0$). Moreover, the distribution of normal pixels is centered around 0.5 while the most frequent value of abnormal pixels is around 0.3.

From the boxplot of feature33, shown in Fig. A.37-b, one can see that the median values for normal and abnormal pixels are around 0.15 and 0.09 respectively. Moreover, for both normal and abnormal pixels we have $|q2 - q1| > |q3 - q2|$ and $length(lower\ whisker) > length(upper\ whisker)$. As a result, corresponding distributions are left skewed. With respect to variation, the spread of normal pixels is a bit more than the abnormal ones. From Fig. A.3 it can be seen that the average values of feature 33 for normal and abnormal pixels are around -0.09 and -0.05 respectively.

Regarding to feature 33, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between normal and abnormal pixels with respect to their variation and best distribution fits but their median, mean and location are different.

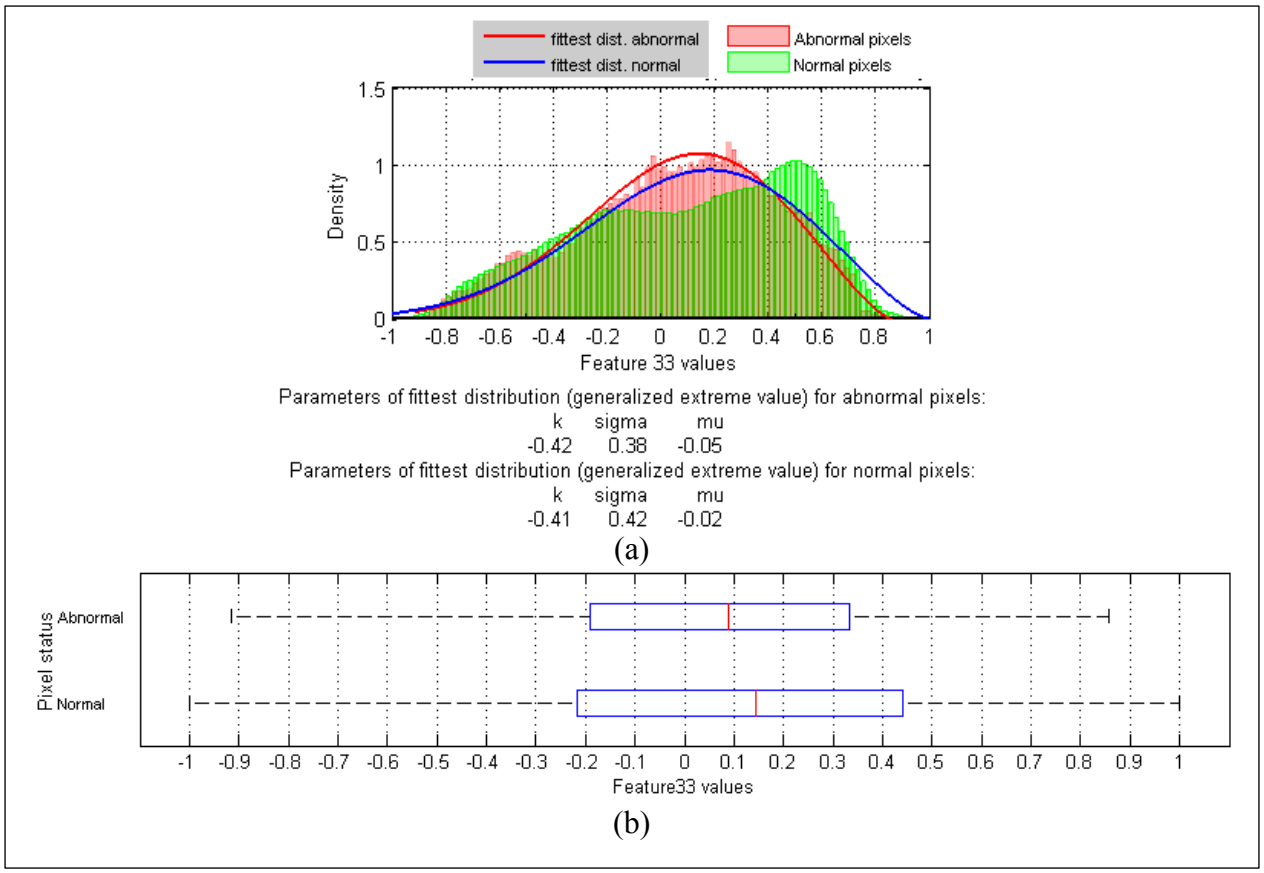


Fig. A.37: (a) bi-histogram of features 33; (b) box plot of features 33.

GLCM information measure of correlation2

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.41) to calculate information measure of correlation 2 feature.

From the bi-histogram of 34^{th} feature which is shown in Fig. A.38-a, we can see that the best distribution fit for both normal and abnormal pixels is type III generalized extreme value ($k < 0$). Moreover, the distribution of both normal and abnormal pixels seems to be bimodal. Normal pixels are centered around 0.1 and 0.7 while abnormal pixels are centered around 0.3 and 0.9.

From the boxplot of feature34, shown in Fig. A.38-b, one can see that the median values for normal and abnormal pixels are around 0.28 and 0.32 respectively. Moreover, for both normal and abnormal pixels we have $|q2 - q1| \cong |q3 - q2|$ and $length(lower\ whisker) \gg length(upper\ whisker)$. As a result, the corresponding distributions are left skewed. With respect to variation, the spread of normal pixels is more than the abnormal ones. From Fig. A.3 it

can be seen that the average values of feature 34 for normal and abnormal pixels are around 0.27 and 0.33 respectively.

Regarding to feature 34, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between normal and abnormal pixels with respect to their best distribution fits but their median, mean, spread and location are different.

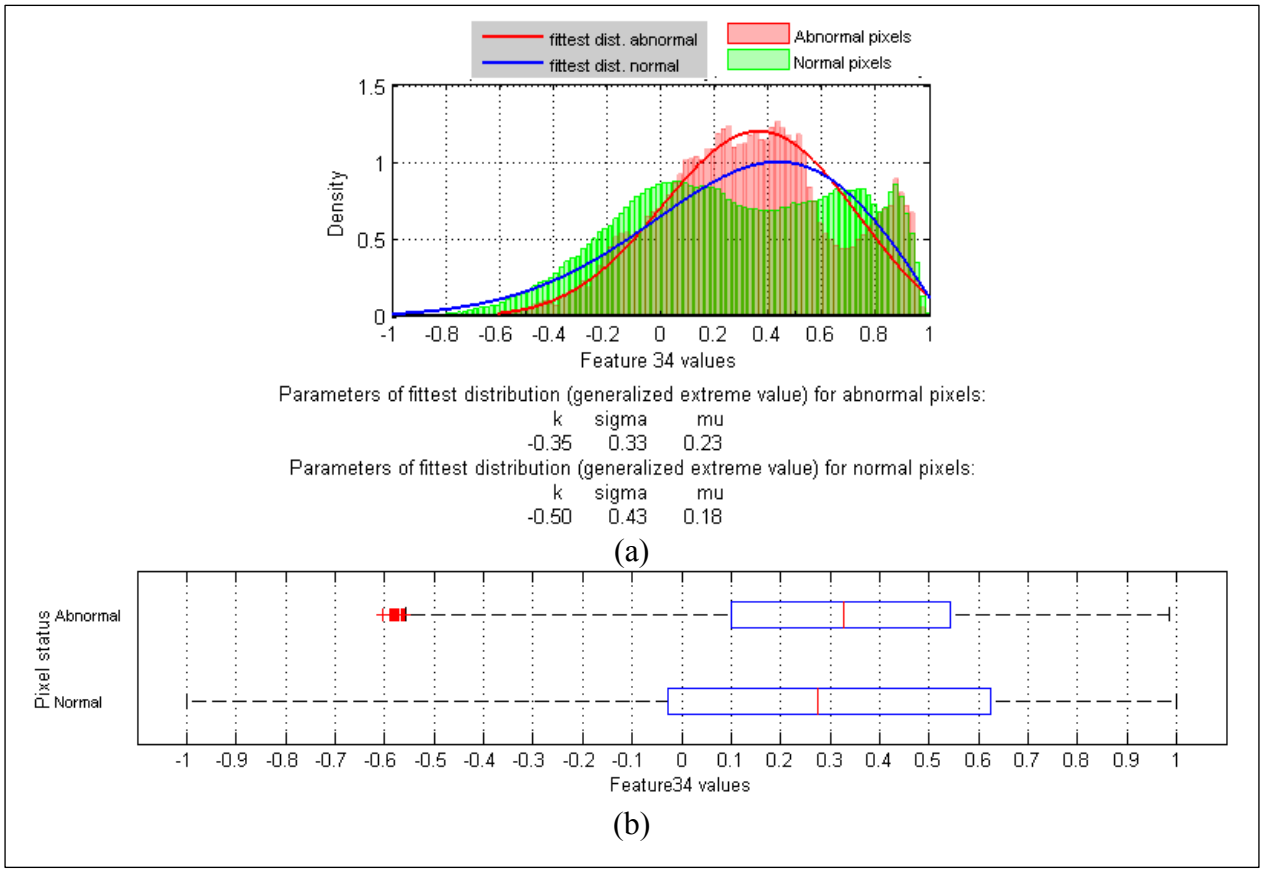


Fig. A.38: (a) bi-histogram of features 34; (b) box plot of features 34.

GLCM inverse difference normalized

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.44) to calculate inverse difference normalized feature.

From the bi-histogram of 35^{th} feature which is shown in Fig. A.39-a, we can see that the best distribution fit for both normal and abnormal pixels is type III generalized extreme value ($k < 0$). Moreover, normal pixels are centered around 0.4 while abnormal pixels are peaked at 0.5.

From the boxplot of feature 35, shown in Fig. A.39-b, one can see that the median value for both normal and abnormal pixels is around 0.35. Furthermore, for both normal and abnormal pixels we have $|q2 - q1| > |q3 - q2|$ and $length(lower\ whisker) > length(upper\ whisker)$. As a result, the corresponding distributions are left skewed. With respect to variation, the spread of normal pixels is more than the abnormal ones. From Fig. A.3 it can be seen that the average value of feature 35 for normal and abnormal pixels is identical and equal to 0.32.

Regarding to feature 35, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between normal and abnormal pixels with respect to their best distribution fits, mean and median but their spreads and locations are different.

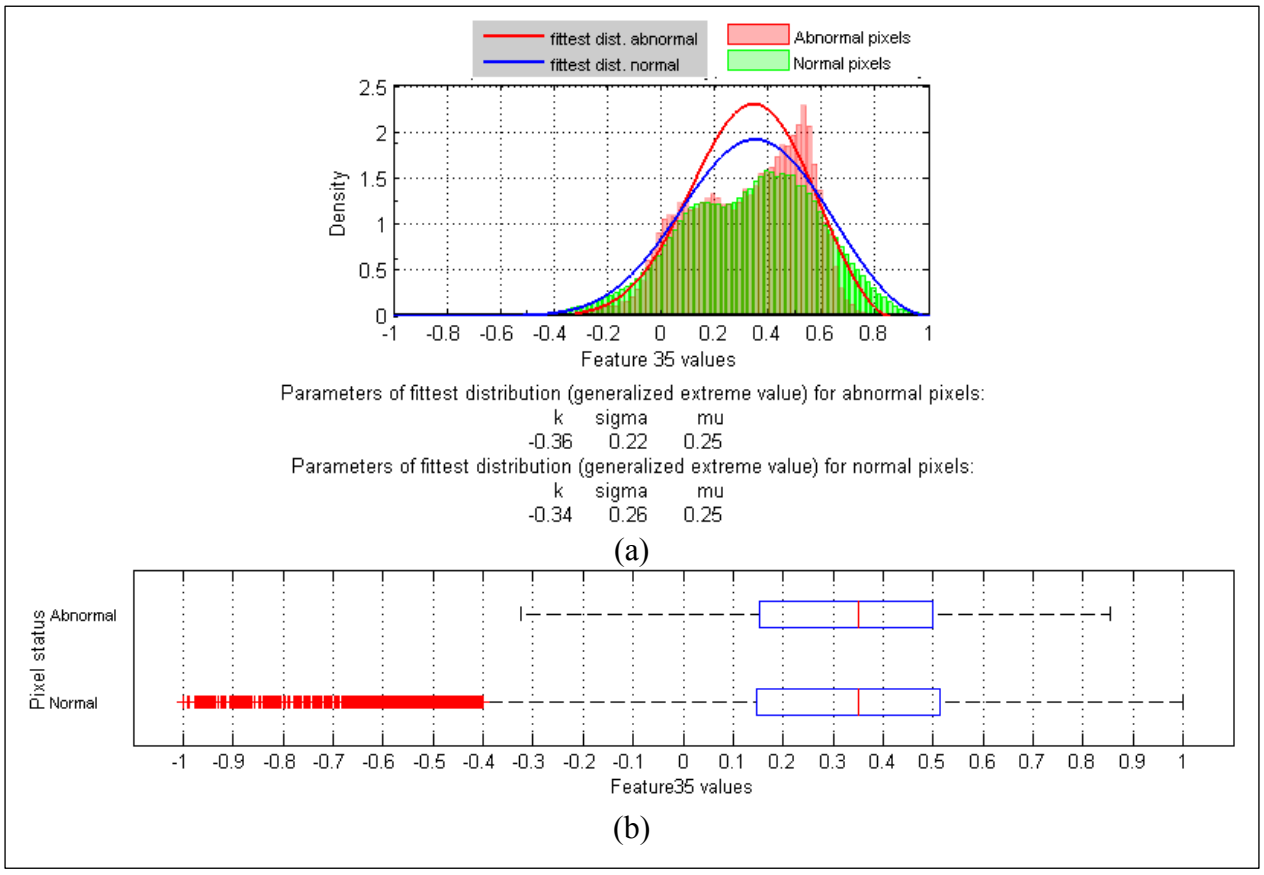


Fig. A.39: (a) bi-histogram of features 35; (b) box plot of features 35.

GLCM inverse difference moment normalized

Having the direction invariant GLCM matrix of 31×31 size window w centered at pixel $P(x, y)$ at hand, we used eq. (3.43) to calculate inverse difference moment normalized feature.

From the bi-histogram of 36th feature which is shown in Fig. A.40-a, we can see that the best distribution fit for both normal and abnormal pixels is t location scale. Moreover, both normal and abnormal histograms are bi-modal. Normal pixels are centered around 0.8 and 0 while abnormal pixels are peaked at 0.9 and 0.1.

From the boxplot of feature 36, shown in Fig. A.40-b, one can see that the median value for both normal and abnormal pixels is around 0.78. Furthermore, for both normal and abnormal pixels we have $|q2 - q1| > |q3 - q2|$ and $length(lower\ whisker) > length(upper\ whisker)$. As a result, the corresponding distributions are left skewed. With respect to variation, the spread of normal pixels is a bit more than the abnormal ones. From Fig. A.3 it can be seen that the average value of feature 36 for normal and abnormal pixels are 0.68 and 0.67 respectively.

Regarding to feature 36, the bi-histogram, boxplot and the mean plot reveal that there is no clear difference between normal and abnormal pixels with respect to their best distribution fits, mean and median but their spreads and locations have small differences.

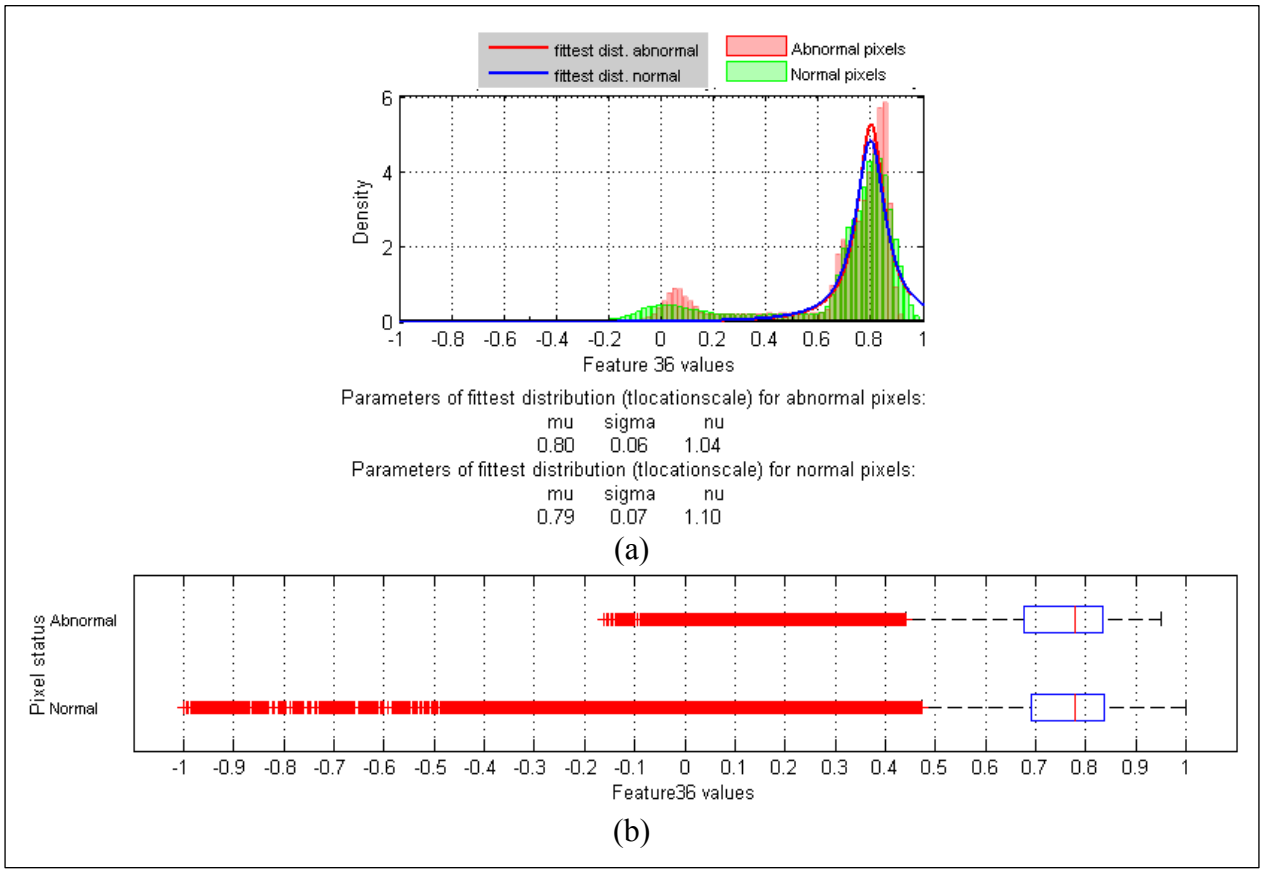


Fig. A.40: (a) bi-histogram of features 36; (b) box plot of features 36.

Variance

The 37th feature in our dataset is the variance of intensity values within window w of size 31×31 centered at normal or abnormal pixel $P(x, y)$. From the bi-histogram of the feature 37 which is shown in Fig. A.41-a, we can see that both normal and abnormal pixels are centered at a value of approximately -0.9. Thus whether a pixel is normal or abnormal does not have any effect on the most frequent value for feature 37. From the boxplot of feature 37 for normal and abnormal pixels, that is shown in Fig. A.41-b, one can see that the median value for normal and abnormal pixels is also around -0.93. From Fig. A.3 it can be seen that the mean values of Feature 37 for normal and abnormal pixels are around -0.83 and -0.86 respectively.

With respect to variation (please see Fig. A.41-b), the spread of normal pixels is more than the abnormal pixels. It is true; because in a normal CT series, we have the ventricles in the middle of the brain which appears very dark as well as the white matters which are quite lighter. This fact produces a high variation within the variance of intensity values around normal pixels. On the other hand, since most of the regions that is marked as abnormal in our dataset were ischemic stroke (i.e., which produces darker intensity values in CT images), the variation of the variance of intensity values around abnormal pixels is smaller than the normal group.

As we can see, the best distribution fit for both normal and abnormal pixels belongs to generalized extreme value family (i.e., Type II because of positive shape parameter). If we take a look to the boxplot of this feature, we can see that for noth normal and abnormal pixels, the length of upper whisker is greater than the lower one; also $|q2 - q1| < |q3 - q2|$. As a result, one can say that both histograms for normal nd abnormal pixels are to some extent right skewed which means that a great majority of windows around eighter normal or abnormal pixels does not have large variance values.

Regarding feature 37, the bi-histogram, mean plot and the boxplot reveal that there is no clear difference between normal and abnormal pixels with respect to their best distribution fits, mean, median and location but their spreads are different.

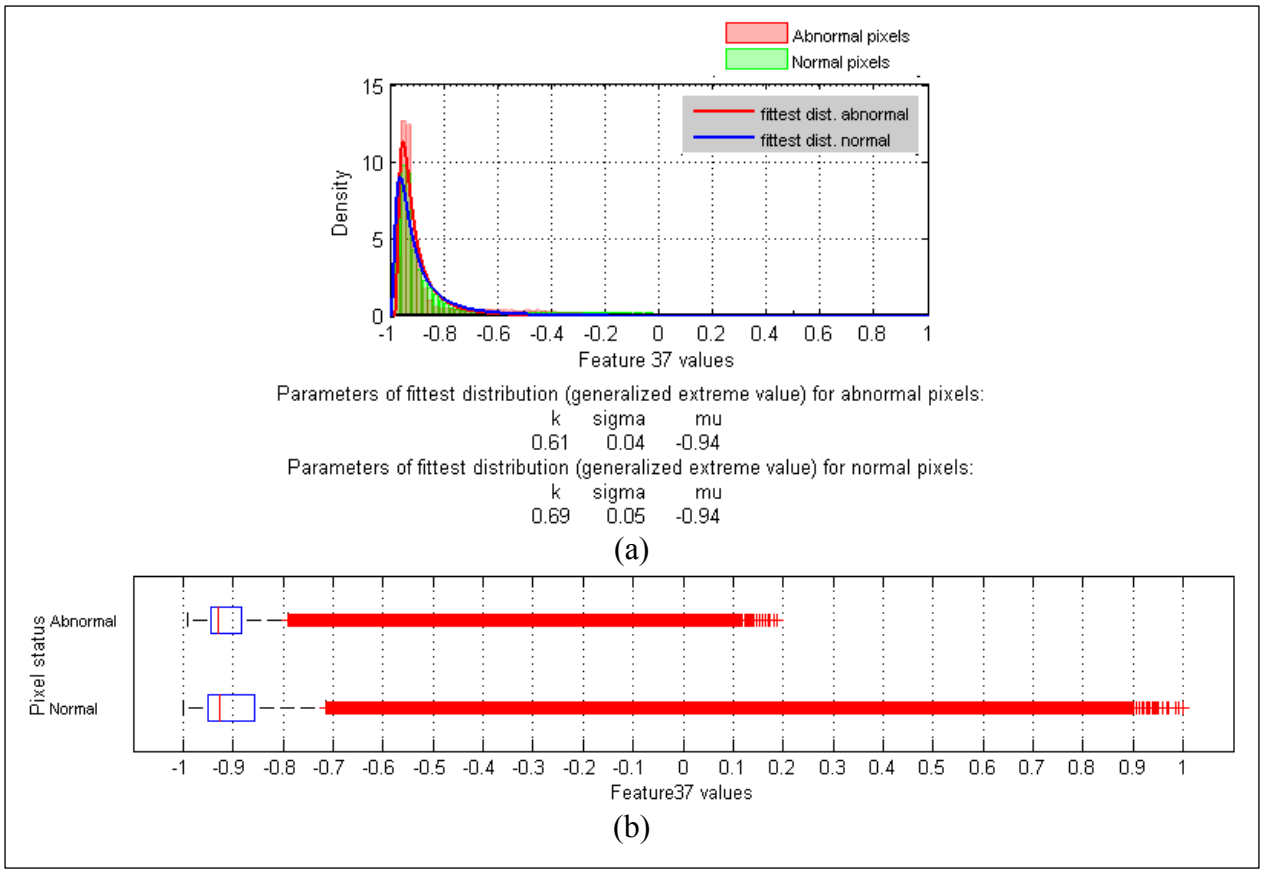


Fig. A.41: (a) bi-histogram of features 37; (b) box plot of features 37.

$\mathbf{F_1, F_2, F_3}$ and $\mathbf{F_4}$

Fig A.42 shows the bi-histograms and box plots of features 38-41. To obtain these features the 8 gray level of histogram of intensity values within window w of size 31×31 centered at normal or abnormal pixel $P(x, y)$ is calculated and eq. (3.9) is used to calculate these four features. As we can see in Fig A.42, the best distribution fit for both normal and abnormal pixels in all features is generalized pareto. Among them, features 38, 40 and 41 have a positive shape parameter ($k > 0$) while the shape parameter of feature 39 is negative.

Looking into Fig. A.3, for features 38 and 41, the mean values for normal and abnormal pixels are very close to each other while for features 39 and 40 the mean values for normal and abnormal pixels have a difference around 0.3.

From the boxplot of feature 38, one can see that the median value for normal and abnormal pixels is around -0.99 and -0.95 respectively. Moreover, for both normal and abnormal pixels,

$|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) < length(upper\ whisker)$. As a result, the distribution of both normal and abnormal pixels is right skewed.

From the boxplot of feature 39, one can see that the median value for normal and abnormal pixels is around 0.61 and 0.88 respectively. Moreover, for both normal and abnormal pixels, $|q2 - q1| > |q3 - q2|$ and $length(lower\ whisker) > length(upper\ whisker)$. As a result, the distribution of both normal and abnormal pixels is left skewed.

From the boxplot of feature 40, one can see that the median value for normal and abnormal pixels is around -0.84 and -0.99 respectively. Moreover, for both normal and abnormal pixels, $|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) < length(upper\ whisker)$. As a result, the distribution of both normal and abnormal pixels is right skewed.

Regarding the boxplot of feature 41, since 77.8% of normal and 75.7% of abnormal pixels (i.e., 1403323 out of 1802695 normal pixels and 49093 out of 64786 abnormal pixels) have a value of -1 for feature 41, $q1 = q2 = q3 = -1$ for both normal and abnormal pixels. As a result, the whiskers and the rectangular part of the boxplot are not visible in feature 41 (i.e., they are all placed in value -1).

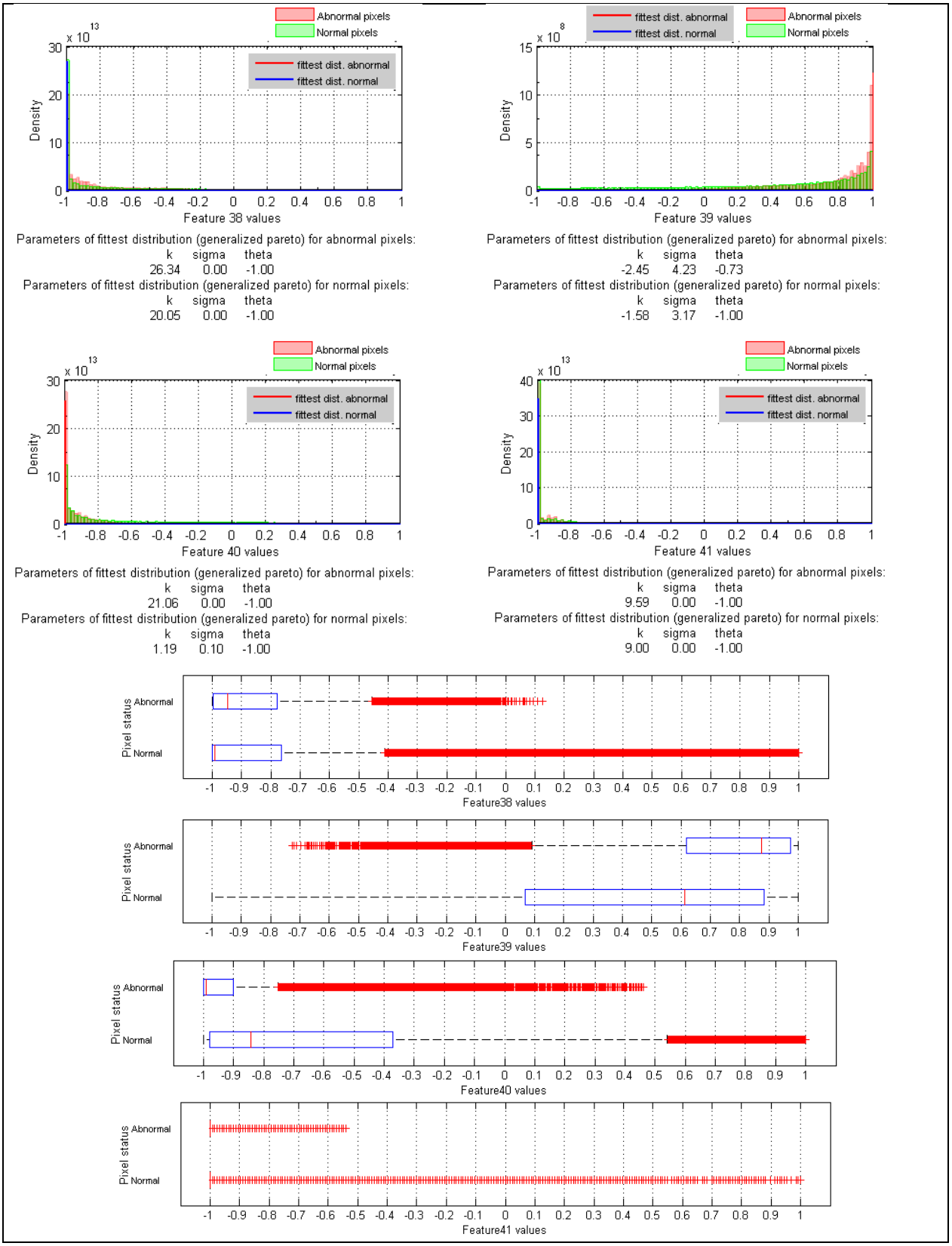


Fig. A.42: bi-histograms and box plots of features 38 (F_1), 39 (F_2), 40 (F_3) and 41 (F_4).

PCC

Fig A.43 shows the bi-histograms and box plots of features 42, 46 and 49 which are the PCC symmetry features calculated for 3 different window size 31×31 , 21×21 and 11×11. As we can see in Fig A.43, considering window size 31×31 (i.e., feature 42), the best distribution fit for normal and abnormal pixels are tlocation scale and normal accordingly. Regarding the window size 21×21 (i.e., feature 46), the best distribution fit for normal and abnormal pixels is logistic and normal respectively. Considering window size 11×11 (i.e., feature 49), the best distribution fit for normal and abnormal pixels is t location scale and generalized extreme value accordingly. Irrespective of the window size, normal pixels are peaked around 0 while abnormal pixels are centered around -1.

Looking into Fig. A.3, the difference between the mean values for normal and abnormal pixels for features 42, 46 and 49 are 0.09, 0.07 and 0.03 respectively which means that the window size 31×31 gives us a better discrimination power between normal and abnormal pixels in this feature.

From the box plots of features 42,46 and 49, shown in Fig. A.43, one can see that regardless of the window size, for both normal and abnormal pixels, $|q2 - q1| \cong |q3 - q2|$ and $length(lower\ whisker) \cong length(upper\ whisker)$. As a result, the distributions of both normal and abnormal pixels are not skewed.

Regarding features 42 and 46, the bi-histogram, mean plot and the boxplot reveal that there is no clear difference between normal and abnormal pixels with respect to their spreads but their best distribution fits, mean, median and location are different.

Regarding feature 49, the bi-histogram, mean plot and the boxplot reveal that there is no clear difference between normal and abnormal pixels with respect to their spreads, means and medians but their best distribution fits and location are to some extent different.

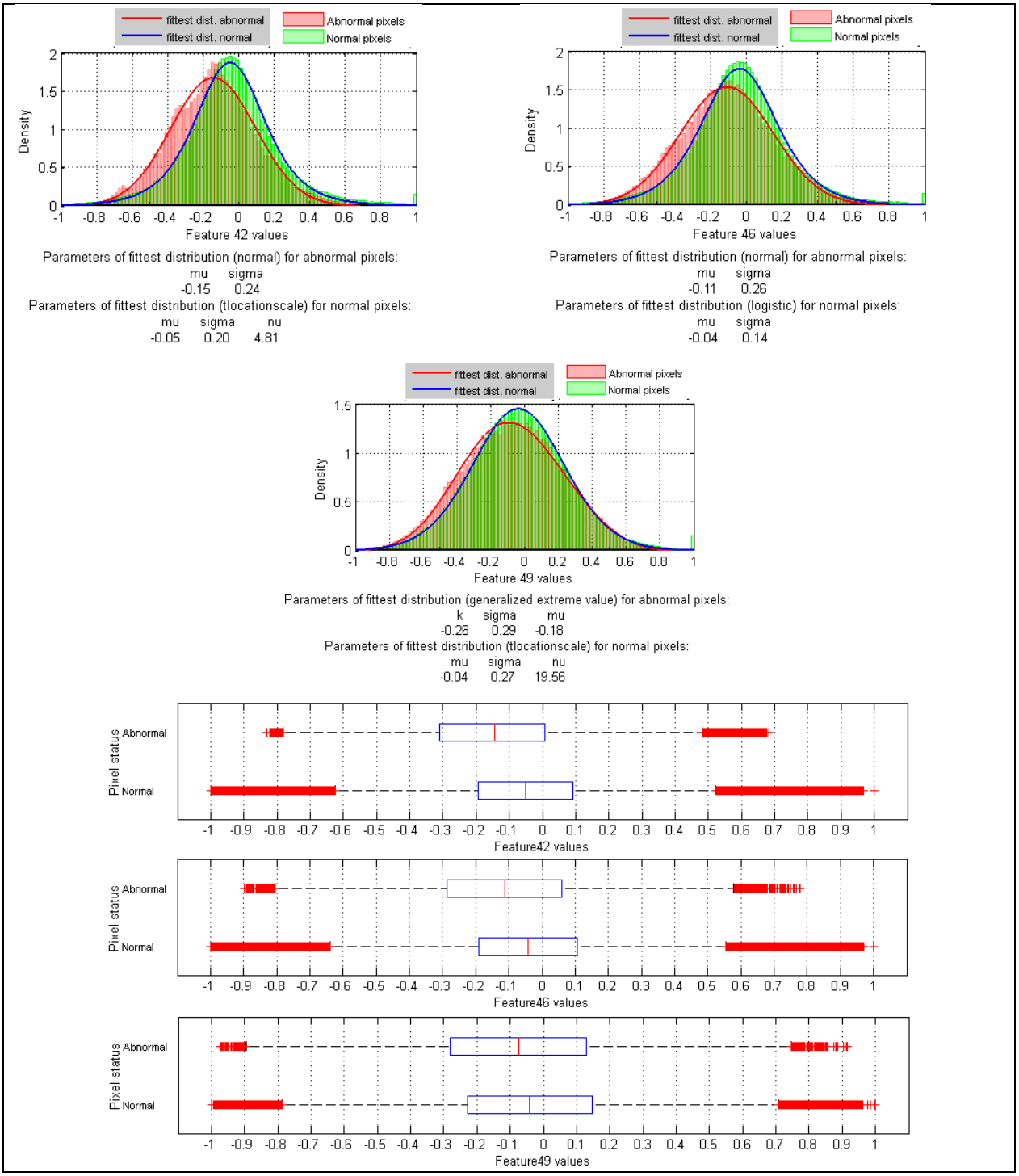


Fig. A.43: bi-histograms and box plots of PCC feature with different window sizes 31 × 31 (Feature 42), 21 × 21 (Feature 46) and 11 × 11 (Feature 49).

Diff

Fig A.44 shows the bi-histogram and box plot of feature 43. This feature compares the intensity value of the pixel that is marked by the expert and its corresponding pixel in the contralateral part of the brain. As we can see in Fig A.44-a, the best distribution fit for both normal and abnormal pixels are generalized pareto. Looking into Fig. A.3, the mean value for normal and abnormal pixels for features 43 is -0.85 and -0.77 respectively. From the box plots of features 43, shown in Fig. A.44-b, one can see that for both normal and abnormal pixels, $|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) < length(upper\ whisker)$. As a result, the distributions of both normal and abnormal pixels are right skewed.

Regarding feature 43, the bi-histogram, mean plot and the boxplot reveal that there is no clear difference between normal and abnormal pixels with respect to their spreads and their best distribution fits but their means, medians and locations are to some extent different.

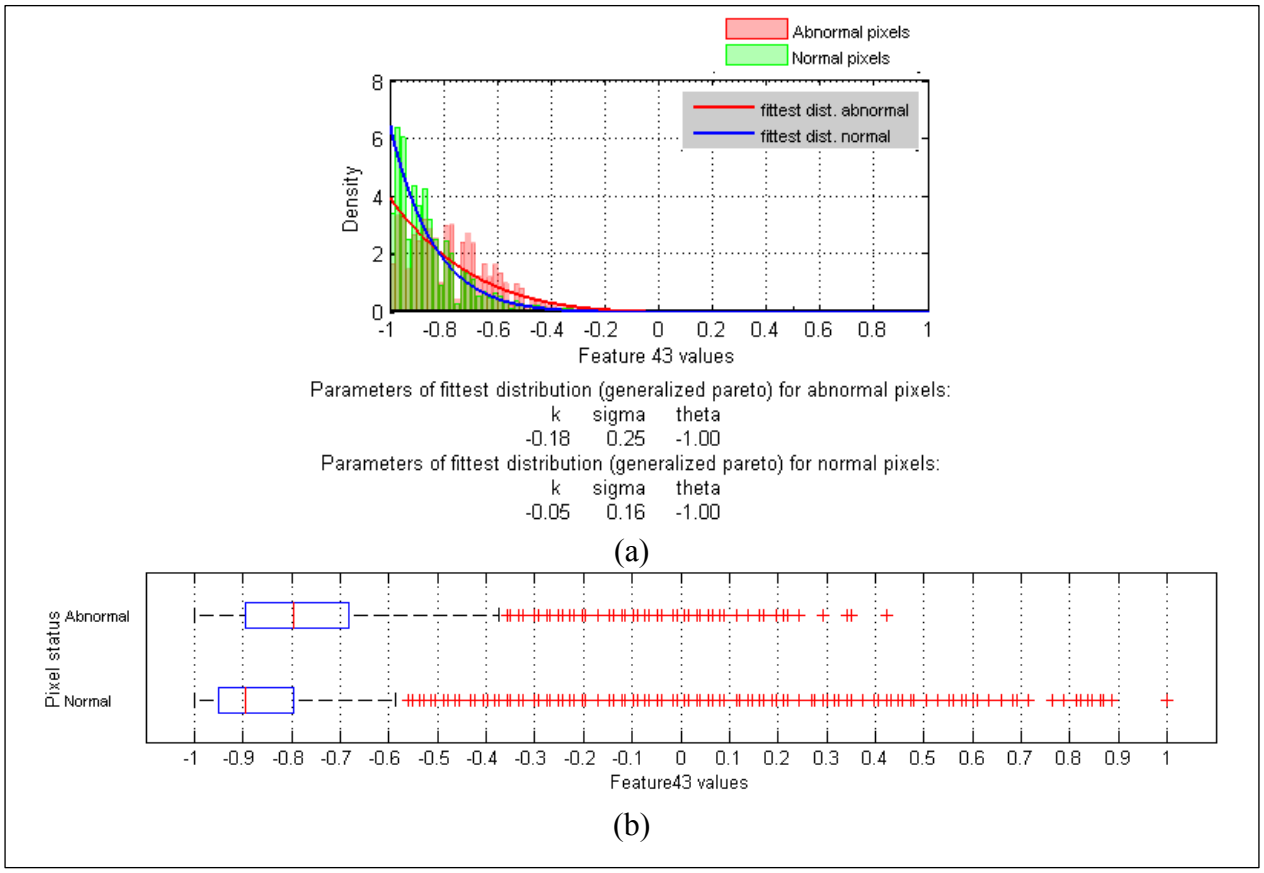


Fig. A.44: (a) bi-histogram of features 43; (b) box plot of features 43.

L_1

Fig A.45 shows the bi-histograms and box plots of features 44, 47 and 50 which are the symmetry features calculated for 3 different window size 31×31 , 21×21 and 11×11 using eq. (3.48). As we can see in Fig A.45, considering window size 31×31 (i.e., feature 44), the best distribution fit for normal and abnormal pixels are generalized extreme value and normal accordingly. Regarding the window size 21×21 and 11×11 (i.e., features 47 and 50), the best distribution fit for both normal and abnormal pixels is generalized extreme value.

Normal pixels in features 44, 47 and 50 are peaked around -0.6, -0.7 and -0.8 respectively. Abnormal pixels in features 44 and 47 have bimodal histograms. The two peaks for feature 44 are around -0.4 and -0.1. Abnormal pixels in feature 47 are centered around -0.5 and -0.3. The abnormal pixels in feature 50 are peaked around -0.6.

Looking into Fig. A.3, the difference between the mean values for normal and abnormal pixels for features 44, 47 and 50 are 0.17, 0.19 and 0.16 respectively.

From the box plots of features 44, 47 and 50, shown in Fig. A.45, one can see that for both normal and abnormal pixels, $|q2 - q1| < |q3 - q2|$ and $length(lower\ whisker) \cong length(upper\ whisker)$. As a result, irrespective of the window size, the distributions of both normal and abnormal pixels are a bit right skewed.

Regarding features 44, 47 and 50, the bi-histograms, mean plot and the boxplots reveal that there is a difference between normal and abnormal pixels with respect to their mean, median, location and spreads. The best distribution fit for normal and abnormal pixels in features 47 and 50 are the same.

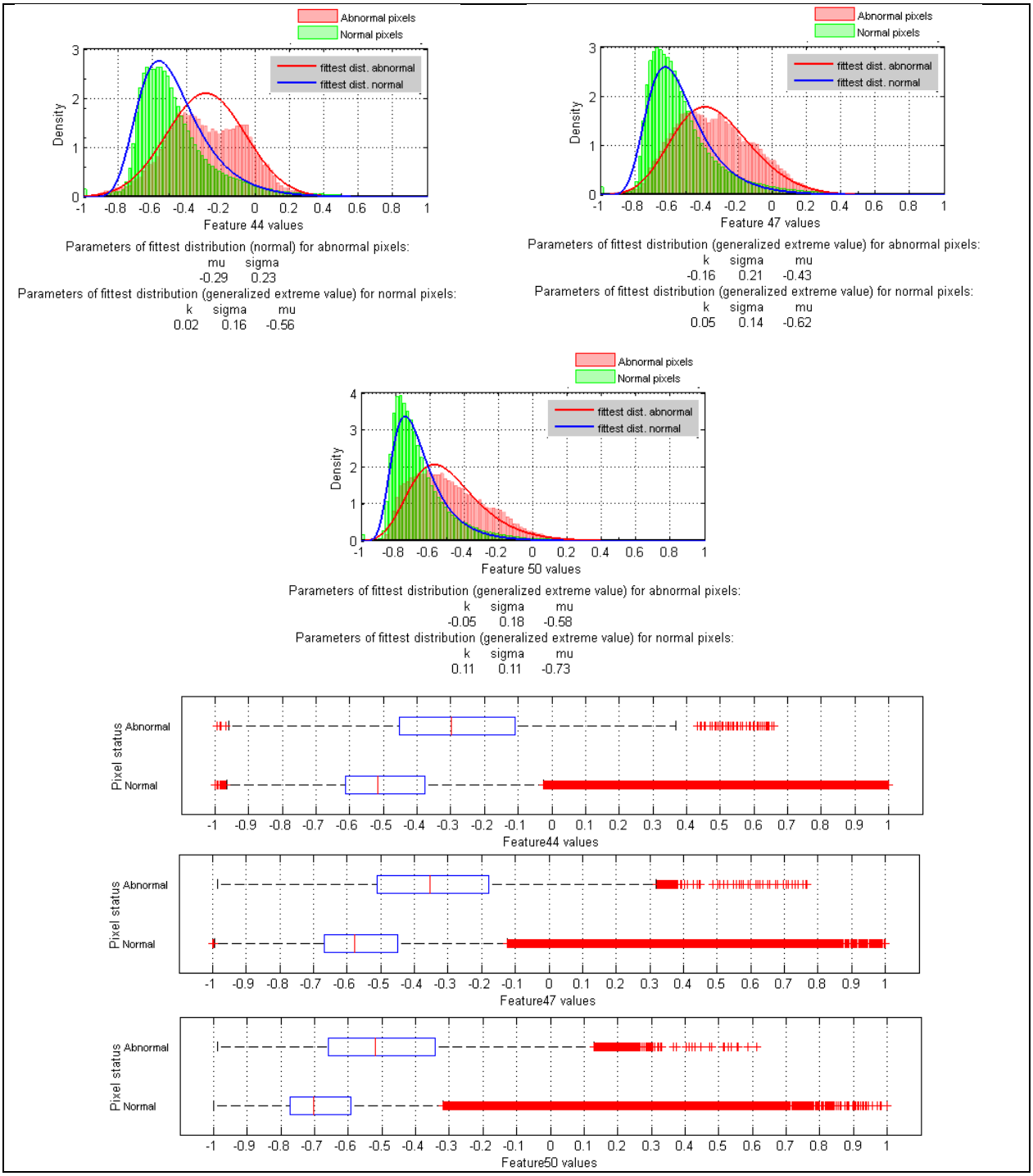


Fig. A.45: bi-histograms and box plots of L_1 feature with different window sizes 31 × 31 (Feature 44), 21 × 21 (Feature 47) and 11 × 11 (Feature 50).

$\mathbf{L_2^2}$

Fig A.46 shows the bi-histograms and box plots of features 45, 48 and 51 which are the symmetry features calculated for 3 different window size 31×31 , 21×21 and 11×11 using eq. (3.49). As we can see in Fig A.46, regardless of the window size, the best distribution fit for both normal and abnormal pixels is generalized extreme value.

Normal pixels in features 45, 48 and 51 are peaked around -0.9, -0.9 and -1 respectively. Abnormal pixels in feature 45 have bimodal histogram centered around -0.8 and -0.5. Abnormal pixels in feature 48 are centered around -0.8. The abnormal pixels in feature 51 are peaked around -0.9.

Looking into Fig. A.3, the difference between the mean values for normal and abnormal pixels for features 45, 48 and 51 are 0.14, 0.12 and 0.06 respectively which means that the window size 31×31 gives us a better discrimination power between normal and abnormal pixels in this feature.

From the box plots of features 45, 48 and 51, shown in Fig. A.45, one can see that for both normal and abnormal pixels, $|q2 - q1| \leq |q3 - q2|$ and $length(lower\ whisker) < length(upper\ whisker)$. As a result, irrespective of the window size, the distributions of both normal and abnormal pixels are right skewed.

Regarding features 45, 48 and 51, the bi-histograms, mean plot and the boxplots reveal that there is a difference between normal and abnormal pixels with respect to their mean, median, location and spreads but their best distribution fits are the same.

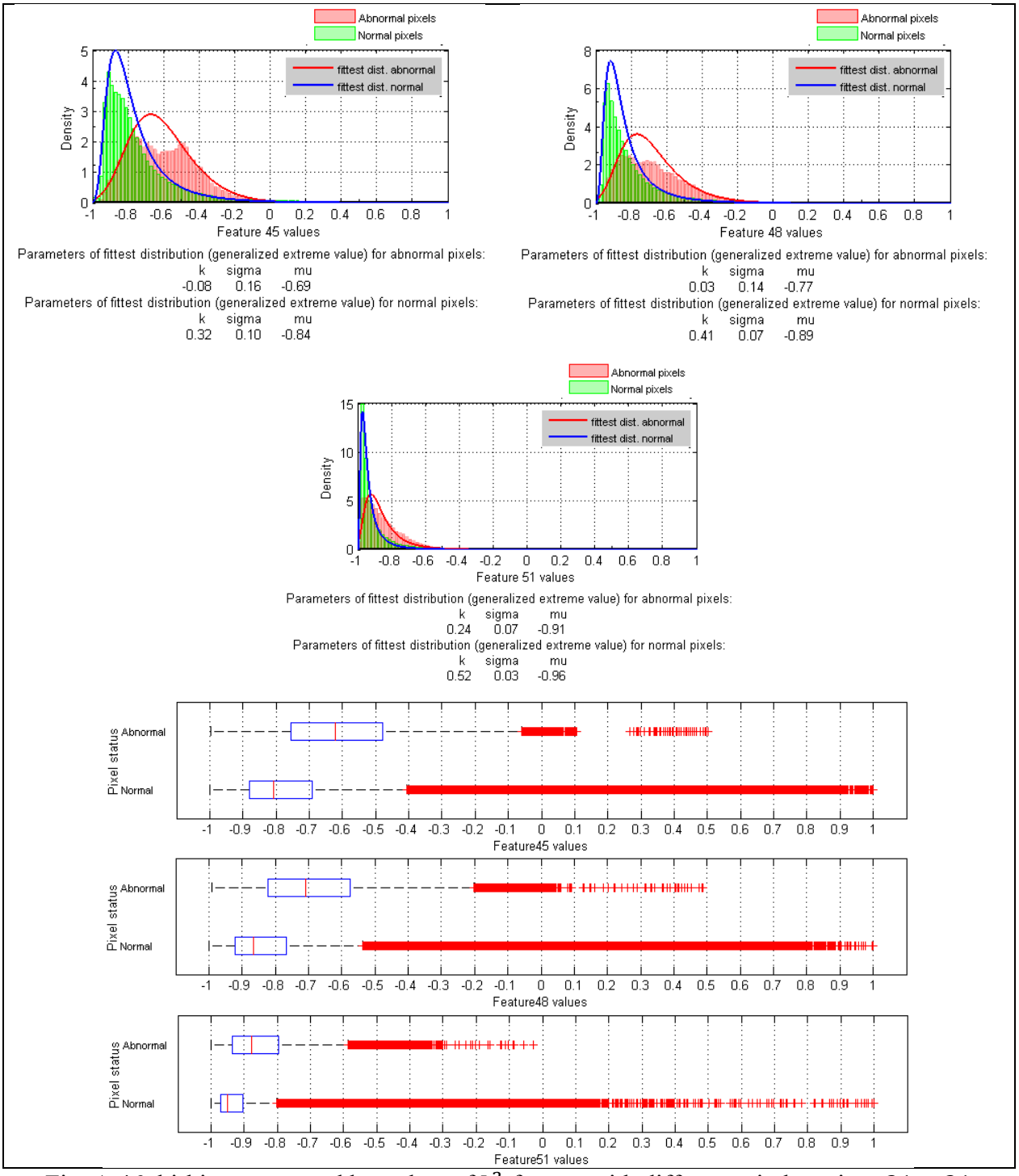


Fig. A.46: bi-histograms and box plots of L_2^2 feature with different window sizes 31×31 (Feature 45), 21×21 (Feature 48) and 11×11 (Feature 51).