UNIVERSIDADE DO ALGARVE

Object Detection and Recognition in Complex Scenes

Hussein Adnan Mohammed

Master Thesis in Computer Science Engineering

Work done under the supervision of:

Prof. Hans du Buf and Dr. Kasim Terzić

2014

Statement of Originality

Object Detection and Recognition in Complex Scenes

Statement of authorship: The work presented in this thesis is, to the best of my knowledge and belief, original, except as acknowledged in the text. The material has not been submitted, either in whole or in part, for a degree at this or any other university.

Candidate:

(Hussein Adnan Mohammed)

Abstract

Object Detection and Recognition in Complex Scenes

Contour-based object detection and recognition in complex scenes is one of the most difficult problems in computer vision. Object contours in complex scenes can be fragmented, occluded and deformed. Instances of the same class can have a wide range of variations. Clutter and background edges can provide more than 90% of all image edges. Nevertheless, our biological vision system is able to perform this task effortlessly. On the other hand, the performance of state-of-the-art computer vision algorithms is still limited in terms of both speed and accuracy.

The work in this thesis presents a simple, efficient and biologically motivated method for contour-based object detection and recognition in complex scenes. Edge segments are extracted from training and testing images using a simple contour-following algorithm at each pixel. Then a descriptor is calculated for each segment using Shape Context, including an offset distance relative to the centre of the object. A Bayesian criterion is used to determine the discriminative power of each segment in a query image by means of a nearest-neighbour lookup, and the most discriminative segments vote for potential bounding boxes. The generated hypotheses are validated using the k nearest-neighbour method in order to eliminate false object detections.

Furthermore, meaningful model segments are extracted by finding edge fragments that appear frequently in training images of the same class. Only 2% of the training segments are employed in the models. These models are used as a second approach to validate the hypotheses, using a distance-based measure based on nearest-neighbour lookups of each segment of the

hypotheses.

A review of shape coding in the visual cortex of primates is provided. The shape-related roles of each region in the ventral pathway of the visual cortex are described. A further step towards a fully biological model for contour-based object detection and recognition is performed by implementing a model for meaningful segment extraction and binding on the basis of two biological principles: proximity and alignment.

Evaluation on a challenging benchmark is performed for both k nearest-neighbour and model-segment validation methods. Recall rates of the proposed method are compared to the results of recent state-of-the-art algorithms at 0.3 and 0.4 false positive detections per image.

Keywords: object detection, edge fragments, Shape Context, computer vision, human vision

Resumo

Object Detection and Recognition in Complex Scenes

A deteção e o reconhecimento de objetos em cenas complexas, utilizando apenas o contorno ou a forma, é um dos problemas mais díficeis na visão por computador. Muitas vezes, os contornos de objetos são fragmentados, ocultos ou deformados, e objetos da mesma classe podem ter uma grande variação. A desordem e as arestas de fundo podem constituir mais do que 90% de todas as arestas de uma imagem. Contudo, o nosso sistema visual é capaz de efetuar essa tarefa sem esforço. No outro lado, o desempenho de algoritmos *state-of-the-art* na visão por computador ainda fica limitado em termos de velocidade e precisão. As razões principais, além da complexidade das arestas em imagens reais, são a aplicação de modelos, muitas vezes desenhados à mão, a aplicação de classificadores complexos, e um longo tempo de treino.

O trabalho nesta tese apresenta um algoritmo simples, eficiente e biologicamente motivado para a deteção e o reconhecimento de objetos complexos em cenas complexas, utilizando apenas segmentos de arestas dos objetos e dos contornos. Segmentos de arestas são extraidos das imagens de treino e teste, utilizando um algoritmo simples para seguir arestas ao nível pixel. Depois, um descritor de cada segmento é calculado com o algoritmo Shape Context, inclusive uma distância relativa ao centro do objeto. Basicamente, o algoritmo Shape Context distribui um número de pontos equidistantes, normalmente vinte, sobre um segmento, e um histograma é construido para cada ponto. O histograma de cada ponto é bidimensional e conta as relacões dos outros pontos em termos de orientações e distâncias. Depois, todos os histogramas de um segmento são concatenados para obter um vetor des-

critivo. Segmentos compridos são cortados em segmentos menores mas com sobreposições, para facilitar o processo de emparelhamento entre segmentos em imagens de treino e segmentos em imagens teste.

Um critério Bayesiano é utilizado para determinar o fator discriminativo de cada segmento numa imagem teste, aplicando uma pesquisa *nearest-neighbour*, e os segmentos mais discriminativos de cada classe de objeto votam para potentiais caixas envolventes (limites) de objetos. Depois de eliminar hipóteses com caixas envolventes sobrepostas, as hipóteses criadas são validadas utilizando o algoritmo *k nearest-neighbour*, com o objetivo de eliminar todas as deteções falsas.

Mais, os segmentos padrão (modelo) típicos são extraidos, procurando fragmentos de arestas que aconteçem frequentemente nas imagens de treino de cada classe. Apenas 2% dos segmentos de treino são aplicados nos modelos de objetos. Estes modelos são utilizados num segundo algoritmo para validar as hipóteses criadas, utilizando uma medida baseada na distância *nearest-neighbour lookup* de cada segmento das hipóteses.

Para atingir um dos objetivos do trabalho, nomeadamente a implementação de um modelo que é ainda mais biológicamente plausível, a tese fornece uma revisão da codificação de formas de objetos no córtex visual de primatas. As tarefas ligadas às formas nas regiões do caminho ventral no córtex são abordadas. Um passo adicional na direção de um modelo completamente biológico para a deteção e o reconhecimento de objetos, utilizando o contorno ou a forma, é feito pela implementação de um modelo de extração de segmentos expressivos e a ligação destes com dois princípios biológicos: a proximidade e o alinhamento.

A avaliação dos algoritmos desenvolvidos foi realizada utilizando uma *benchmark* que constitui um desafio, ETHZ, no caso dos métodos *k nearest-*

neighbour e *segmentos-modelo* de validação. As taxas *recall* dos dois algoritmis são comparadas com as taxas de algoritmos do estado da arte recente a dois níveis de fiabilidade: 0,3 e 0,4 deteções positivas falsas por imagem. Embora que os algoritmos desenvolvidos nesta tese são muito mais simples e rápidos, os resultados obtidos mostram que ainda não foi possível atingir o topo do estado da arte, mas pelo menos já conseguem entrar na competição.

Termos chave: deteção de objectos, fragmentos de arestas, Shape Context, visão por computador, visão humana

Acknowledgements

I would like to express my gratitude to my supervisors Prof. Hans du Buf and Dr. Kasim Terzić for the useful comments, remarks and engagement through the learning process of this master thesis. Furthermore, I would like to thank my sister, Sarah, who has supported me throughout the entire process, and my family for their encouragement. Also, I would like to thank the mobility office staff at the University of the Algarve for their constant support.

This work was supported partially by the EU under the FP-7 grant ICT-2009.2.1-270247 NeuralDynamics and the Erasmus Mundus action 2, Lot IIY 2011 scholarship program.

To my fiancé Duaa

Contents

List of Figures

Chapter 1

Introduction

1.1 General Background

A distinction is made in the literature between detection and recognition of an object. Detection usually refers to the process of detecting the existence of an object in a given image, together with a rough estimation of the location and size of that object. Recognition is assumed to be the process of identifying and validating detected objects. Recognition can also be considered as a categorisation problem, where the goal is to classify detected objects into certain classes.

Since shape is a natural property of many complex structures, it can be used as a clue to detect and recognise objects, especially when we consider the fact that most objects are best recognised by their global shape, for example bottles and swans.

It has long been known through psychological experiments that shape plays a primary role in real-time recognition of intact objects [5]. In fact, human vision is capable of locating and recognising objects based on their shape easily, even when they are occluded or seen in a heavily cluttered

scene. In other words, human vision does not need fully connected contours to recognise objects with distinctive shapes in images; a few non-connected contour fragments are sufficient without any texture or colour information. In Fig. 1.1 we can see an example of the powerful processing capability of the human visual system. We can easily recognise the object given its edge map, despite the missing parts of the contour and the complex background.

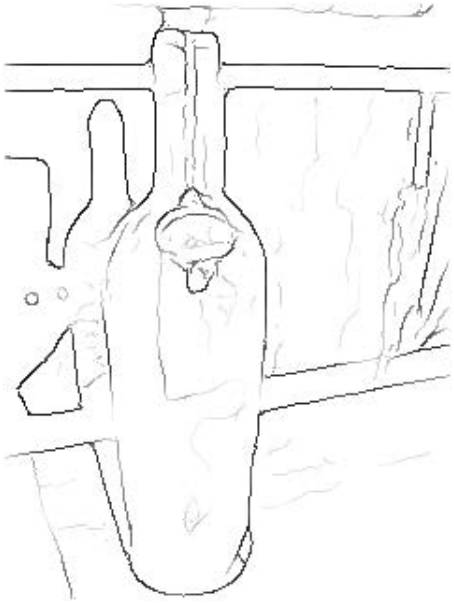


Figure 1.1: Example of the ability of human vision to detect and recognise objects in a complex scene given only few non-connected edge fragments.

Shape is perhaps the most important feature in human vision. Consequently, it has also been an important field of study in computer vision, artificial intelligence, psychology and cognitive neuroscience. While powerful computational algorithms exist for recognising complete and clutter-free contours, shape detection in noisy natural images remains a challenging problem. The principal issues are the lack of complete and reliable contours due to the difficulty of bottom-up segmentation, and the presence of occlusions and background clutter.

(a) Bottle in a simple scene. (b) Giraffe in a simple scene. (c) Bottle in a complex scene. (d) Giraffe in a complex scene.

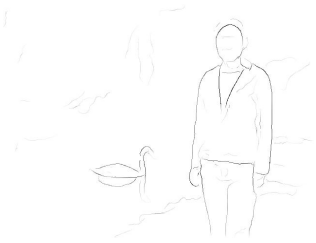
Figure 1.2: Difference between simple and complex scenes. Detecting objects in complex scenes is no longer a problem of simple shape matching.

Simple edge fragment descriptors like Shape Context [3] followed by direct distance measurement like the sum of matching errors between corresponding points was enough to recognise and classify clutter-free shapes with complete contours. On the other hand, objects in complex scenes cannot be detected directly by employing their holistic shapes; it is thus no longer a simple shape matching problem. Dealing with fragmented contours where object parts are occluded or connected to clutter is a difficult task indeed, especially if we take into consideration that objects might occupy as little as 10% of the overall image. This means that we have to deal with a very low signal-to-noise ratio; see Fig. 1.2.

There are two principal approaches to accomplish contour-based object detection and recognition in complex scenes [65]: either by segmenting regions, which generally cannot be assumed to correspond to an entire object, or by trying to string together edge segments and corners into complete objects; see Fig 1.3. A comprehensive review of state-of-the-art contour-based object detection approaches can be found in [47]. The edge detection approach is more biologically plausible since experimental evidence and research

(a) The original image.

(b) Result of image segmentation.



(c) Result of edge extraction.

Figure 1.3: Two different approaches for shape-based object recognition: segmentation and edge detection. Images are from the ETHZ dataset [18], see Section 5.1.

suggest that extraction of outermost contours by coding bars and edges in the scene occurs in the visual cortex [48, 58, 5, 11].

The visual cortex in primates is able to accomplish object detection and recognition with an amazing accuracy and in real-time (about 150-200 ms [1]) with almost no effort. Such performance and efficiency has been the goal of computer vision for decades. However, it has never been possible to match human vision's performance, except on very specific constrained tasks. Nevertheless, much research effort resulted in a rapid increase of the classification rate. In the course of three years, the classification rate on the Caltech-101 database rose from under 20% in 2004 to almost 90% in 2007 [6].

1.2 Deep Hierarchy of the Primate Visual Cortex

Although information about the detailed wiring and functionality is not yet available for most of the regions in the primate visual cortex, the general layout indicates the existence of a deep hierarchy of the different regions with feedback and feedforward connections [30]. This hierarchical system

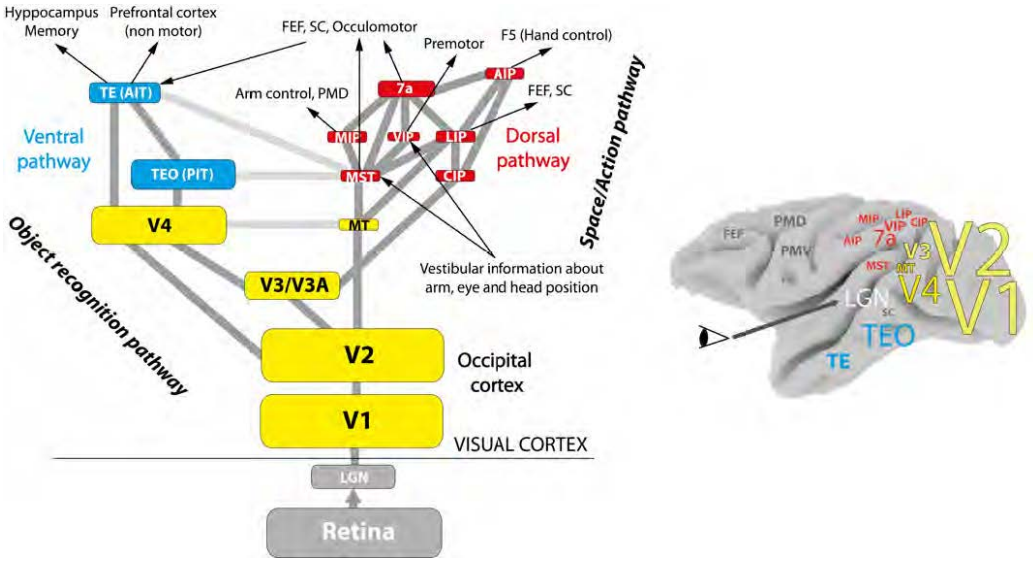


Figure 1.4: Simplified hierarchical structure of the primate's visual cortex and approximate area locations. Box and font sizes are proportional to the area size. Reproduced from [30].

consists of two interconnected streams: dorsal, which is referred to as the *where pathway*, and ventral, which is referred to as the *what pathway*. As the name suggests, the ventral stream (what) is believed to be responsible for object recognition, along with other tasks. A simplified structure of the primate's visual cortex is illustrated in Fig. 1.4.

In the primary visual cortex, also called area V1, there are simple, complex and end-stopped cells. These cells serve to code, at many scales, lines, edges and keypoints. This information is very useful for object recognition. In the hierarchical visual system, simple low-level features are detected and extracted (e.g. orientation, motion, disparity, etc.) over small local regions of visual space in area V1 [53]. These are then combined in higher areas V2 and V4 that have larger visual fields in order to detect more complex and bigger structures like contour fragments, but with a loss of localisation: from local to global representations [11]. These structures are further combined in

the inferior-temporal (IT) cortex to represent even more complex features, which may cover as much as half of the visual field and therefore represent entire objects [58].

It is not yet clear how exactly human vision performs contour-based object recognition. Accumulated evidence suggests that shape representation in primates starts in V4 by recognising meaningful parts of contours, using some kind of curvature calculation on lines and edges extracted in V1 and V2. Then more sophisticated relations are constructed in IT cortex between parts of the same object in accordance with the shape centre to provide the final description of the perceived object in terms of its outermost contours [30].

Due to the complexity of the visual system, most of the current computer vision algorithms avoid implementing such a hierarchy to achieve active real-time vision and focus instead on specific tasks. This approach resulted in flat processing schemes rather than deep hierarchies (see Fig. 1.5), and thus widened the gap between biological and computer vision. Nevertheless, there are some biologically plausible models like HMAX [52], which was extended later in [60], and deep convolutional neural networks (e.g. [29, 8]) which are the result of cumulative efforts since the proposal of the Neocognitron in [20]. These models mimic the hierarchical process in human vision, but they are very slow and resource-demanding, especially during training. The work in this thesis is therefore an attempt to implement a fast and efficient algorithm that is biologically motivated. Although this work is not yet fully biological, it is a first step in this direction.

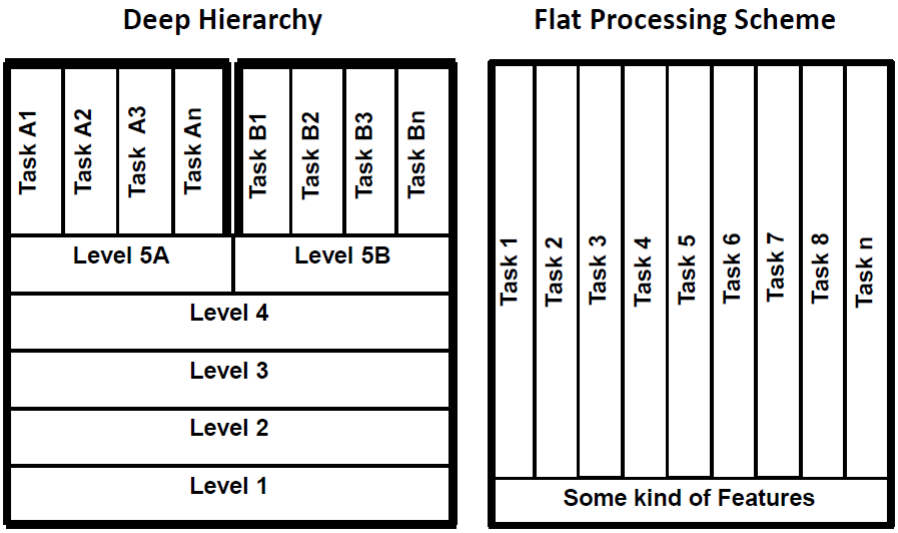


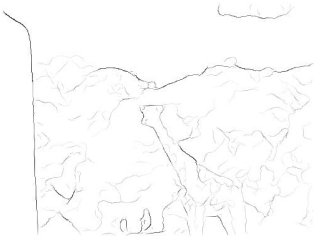
Figure 1.5: A deep hierarchy versus flat processing scheme. Reproduced from [30].

1.3 Challenges

While we are able to accomplish visual tasks like object detection and recognition in complex scenes without any noticeable effort, this is not the case for artificial vision systems. Implementing a fast, efficient and reliable algorithm to do so is a challenging task, knowing that contour-based object detection and recognition in complex scenes has been proven to be one of the most difficult problems in computer vision.

Many problems complicate the ability to detect and recognise an object by its shape within a complex scene (see Fig. 1.6a). Perhaps the worst problem is getting a clean edge map of the scene by low-level image processing. Even with a state-of-the-art edge-detection algorithm, like the Berkeley edge detector [41], we will miss parts of the contours in the presence of clutter; see Fig. 1.6b. In fact, contour extraction has an important top-down component, and our expectations play a large role to do that. Because of this,

(a) An image of a giraffe in a complex scene.



(b) Edge map of the giraffe using the Berkeley edge detector.

(c) The real contours of the giraffe marked manually.

Figure 1.6: Limitation of edge extraction in computer vision.

(a) Slim mug

(b) Wide mug

Figure 1.7: Variation between instances of the same class.

no purely data-driven, bottom-up algorithm can be expected to extract complete object contours; see Fig. 1.6. In addition, objects from the same class can greatly differ, which is referred to as intra-class variation; see Fig. 1.7. Therefore, recognising an object by its shape in a complex scene is not an obvious task at all. In real-world images one can expect all types of complications: clutter, occlusions and intra-class variations, not to mention changes in scale and viewpoint of the objects and scenes.

Furthermore, extracting features locally to obtain global shapes is not possible because shape is an emerging property that becomes only apparent

after all object boundary contours have been grouped, unlike colour or texture which can be captured by small image regions (patches). This leads us to one of two approaches: either connecting fragmented contours depending on some rules like Gestalt principles (proximity, co-linearity, etc.), or trying to find some geometrical or spatial relations between fragments. Further details will be provided in Chapter 2.

Despite all challenges that face contour-based object recognition, there is an increasing number of proposed solutions to overcome the obstacles. Nevertheless, most state-of-the-art methods are complex and slow (the reasons will be discussed later in Chapter 2), which means that they cannot be used in a real-time scenario. This issue adds an additional challenge: to develop a simple, efficient and fast algorithm.

Developing a biological model for contour-based object recognition can contribute to the unified architecture of active vision as proposed in [66]; see Fig. 1.8. In that work, a number of biologically plausible algorithms are applied in an attempt to build a general-purpose, real-time active vision system. Contour-based object recognition can be integrated as part of the ventral pathway where object recognition takes place.

Recognising an object in early extrastriate cortex (namely area V4) by its shape or even by some of its contour parts is very useful for rapid object categorization. This can trigger attention using fast feedforward (bottom-up) information, such that higher areas like inferior temporal cortex can perform further processing, like biasing likely object categories in memory [59]. This strategy can be helpful in robotics where real-time processing has to be guaranteed: the estimated location of an object in the scene can be passed down from higher areas and it can trigger attention to direct a more precise visual search. Furthermore, contour-based object recognition can be

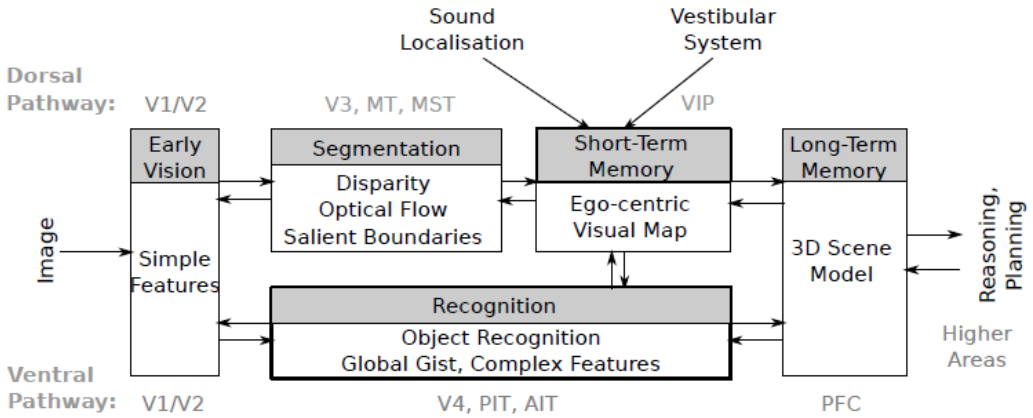


Figure 1.8: Overview of a biologically-inspired active vision system by [66]. The top path models the dorsal pathway (localisation, motion and attention), the bottom path the ventral pathway (recognition). Grey text indicates corresponding cortical areas.

used as a standalone algorithm in many cases where objects to be recognised are best represented by their shape, such as a traffic environment; see [4].

1.4 Structure of This Thesis

The rest of this thesis is structured as follows: Chapter 2 presents related work in the field of object detection and recognition in general, and in particular the field of contour-based object detection and recognition, the main topic of this thesis. Chapter 3 introduces the developed algorithm to solve the problem of contour-based object detection and recognition in complex scenes. Chapter 4 provides an overview about shape coding and contour-related processing in the visual cortex of primates and describes a biologically inspired segment extraction approach. Chapter 5 is dedicated to experimental evaluation of the methods on a standard dataset. Chapter 6 provides a discussion of presented work and possible future work.

Chapter 2

Related Work

2.1 Object Detection and Recognition

The most common method for object classification in computer vision is Bag-of-Keypoints (BoK) [9]. This method is motivated by the Bag-of-Words (BoW) approach for text categorisation [25, 67, 35]. This technique basically involves the extraction of local features like image patches using any invariant descriptor like SIFT [36], then creating a codebook from these features under the independence assumption, e.g., by using k-means clustering and considering each cluster as a word; see Fig. 2.1. Resulting feature vectors are usually classified using powerful parametric classifiers like Support Vector Machines (SVM). Relations between words are not considered in this technique. Since the features are local, spatial information of the patches is not employed. Also, fine differences between features are lost due to the clustering process. Several attempts have been carried out to overcome these weaknesses. To consider also the spatial layout of extracted features, Lazebnik et al. [31] proposed the famous state-of-the-art Spatial Pyramids technique, which greatly improves the performance of object classification using BoK.

This improvement clearly indicates the importance of considering the neighbourhood of extracted features and spatial layout as additional information in object classification.

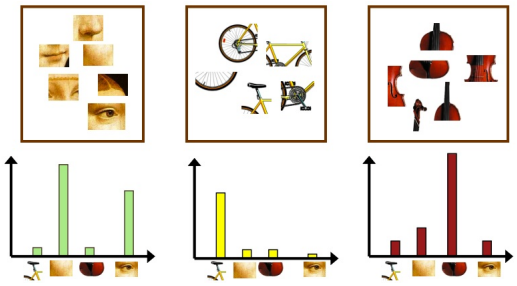


Figure 2.1: Bag-of-Keypoints illustration. Creating histograms from image patches.

Classification methods can be divided into two categories: parametric and non-parametric. Parametric classifiers construct a model from the data and try to estimate the parameters for that model, while non-parametric classifiers attempt to classify by comparing testing data directly with the training data. Each method has its advantages and disadvantages. Obviously, parametric classifiers require a training phase to determine the parameters of the underlying model. Also, learning a new class typically requires re-training the entire classifier. Furthermore, parametric classifiers are usually resource-hungry and slow during training. On the other hand, the main problem with non-parametric classifiers is poor performance, which has been addressed and fixed by Boiman et al. [6].

Since many non-parametric classifiers are based on nearest-neighbour distance estimation, they inherited the bad reputation regarding low classification rate that they can offer. This proved to be a wrong assumption according to the discussion in [6]. The authors proposed a state-of-the-art classifier which they called Naive Bayes Nearest-Neighbour (NBNN). They argued that two practices actually lead to significant degradation in the performance of

nearest-neighbour distance estimation (and by extension of non-parametric classifiers). The two practices that should be avoided are:

1. **Descriptor quantisation:** this practice is necessary if we want to construct a codebook for the BoK technique, but it can cause a large loss of information for non-parametric classifiers because they do not have a training phase to compensate this loss. Furthermore, it is not necessary in terms of time efficiency in case of the NN-based technique.

2. **Image-to-image distance:** measuring the distance between descriptors of a query image and descriptors of the closest class (image-to-class) directly using nearest-neighbour estimation can provide a much better performance than measuring the distance between descriptors of the query image and descriptors of the closest image (image-to-image), because the search will be generalised to class matching instead of image matching and will cope better with intra-class variations.

An NBNN classifier can be summarised as follows:

1. Compute n descriptors $d_1, ..., d_n$ of the query image q.

2. $\forall d_i \, \forall c \in C$ compute the NN of d_i in c: $NN_c(d_i)$. NN_c is the Nearest-Neighbour in class c.

3. $\hat{C} = \underset{C}{\operatorname{argmin}} \sum_{i=1}^{n} \parallel d_i - NN_c(d_i) \parallel^2$.

An improvement of NBNN has been proposed in [43], which further increased the classification accuracy and improved the ability to scale to a large number of classes (run time of improved NBNN grows with the log of the number of classes rather than linearly); see Fig. 2.2. The main modification is eliminating the need to search for a nearest-neighbour match in

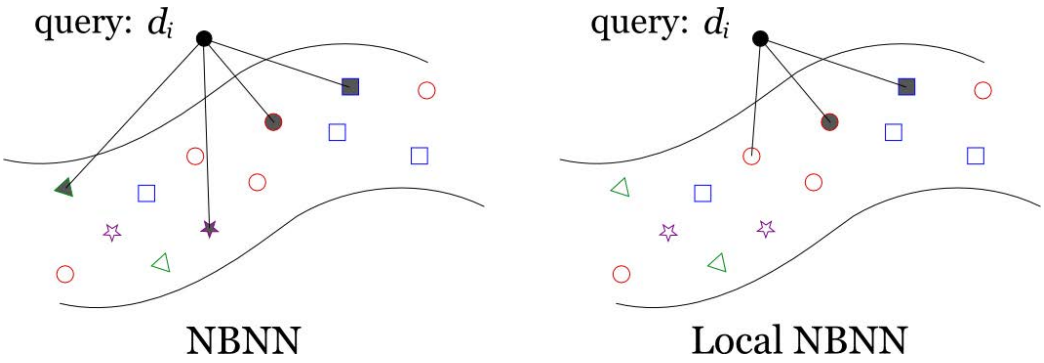


Figure 2.2: The difference between NBNN and Local NBNN. NBNN forces a query descriptor d_i to search for its closest neighbour in all classes. Local NBNN requires the query descriptor to search only the closest classes. Reproduced from [43].

all classes; instead, only the classes within a certain neighbourhood of the query descriptor in feature space are considered. This simple modification results in a significant speed-up over the original NBNN and yields a better performance.

NBNN is typically applied to image patches using descriptors like SIFT in order to classify objects. It has never been applied to contours. The work in this thesis proposes to apply NBNN to contour fragments, as will be shown in detail in Chapter 3.

2.2 Contour-Based Object Detection and Recognition

2.2.1 Computational Methods

In recent years, focus has shifted from classifying entire binarised shapes, where the complete contour of a shape is available, to contour-based object detection in noisy, natural images, such as in the ETHZ Shapes Dataset [18];

see Fig. 1.2. Contours detected in such images are inherently unreliable: objects are typically broken into numerous contour segments, large parts of the object contour may be missing, segments have varying length, and there is a considerable amount of clutter.

Methods for contour-based object detection approach the problem in different ways: by using powerful kernel-based classifiers [45, 17], optimised Chamfer matching [34, 61], learning ensembles of short contour segments [50, 19], or by using Gestalt principles for grouping adjacent segments into larger ones [70, 27, 49, 71, 42]. Much of the complexity of these methods is devoted to compensating for incomplete and noisy features.

Many recent state-of-the-art approaches use machine learning algorithms and novel shape models in an attempt to learn consistent configurations and groupings of segments in order to extract reliable features for detection [19, 17, 50, 45]. The steady improvement in detection rates shows that such methods are increasingly successful at extracting information from data, but most of these methods are complex and therefore slow, and knowledge about which segments are useful is often hidden deep inside model parameters.

Several algorithms have been proposed in recent years to perform contour-based object detection and recognition in complex scenes. Although they vary in terms of methodology and structure, they all need a way to describe the features of extracted contour fragments in the edge map of a given image. There exist a number of proposed shape descriptors, but the most common one by far is the Shape Context descriptor [3], which will be described briefly in Chapter 3. When Shape Context was introduced, focus was on simple closed contour shapes with no scale variations. Attention has shifted since then to recognising objects in complex scenes with intra-class variation in both scale and viewpoint. Since closed contours cannot be ob-

tained in complex scenes, Shape Context can be used to describe individual contour fragments instead of the overall shape; see [75]. Nevertheless, some authors , like Shu and Wu [62], are still investigating global shape descriptors inspired by Shape Context to solve simple shape matching with fully connected contours.

After extracting shape features, there is a matching phase between training and query images. A classical method for shape matching is Chamfer matching, in which the distance is defined as the average distance from points on the training shape to the nearest points on the query shape [2]. However, it has been repeatedly noted that Chamfer matching does not cope well with clutter and shape deformations. Even if a hierarchy of many templates is used to cover deformations, the rate of false positives is rather high (typically more than one false positive per image or FPPI [57]). Nevertheless, Chamfer matching has not been ignored; [34] improved the accuracy of Chamfer matching while the computational time was reduced from linear to sub-linear. Also, edge orientation information was included in the matching algorithm, which resulted in a piecewise smooth cost function. Such improvements allow to use Chamfer matching in the hypothesis validation phase [61].

Fragmented contours in complex scenes can have any length and curvature, and they are often caused by clutter in the image. Therefore, some researchers focused on high-curvature points in contours to cope with deformations of the object, assuming that deformations typically happen at high-curvature points. An example is the work done in [50] which uses short line fragments, favouring curved segments over straight ones, and allowing for certain joints by splitting edges at high curvature points. In this thesis we will avoid such "manual" interventions by using a discrimination criterion,

where discriminative segments by definition should have distinctive properties which are specific to the class that the object belongs to.

For category-level object detection, there are two commonly used techniques: sliding windows, where windows of different sizes (typically 6-8 sizes) scan the image by moving them a few pixels in each iteration (e.g. [19]), and voting methods which are dominated by the Generalised Hough Transform, where the problem of finding the model's position in a query image is cast into the problem of finding the transformation parameters which map the model into the image (e.g. [28]). As an alternative to these two leading approaches, [45] proposed a weighted, pairwise clustering of voting lines to obtain globally consistent hypotheses. It is basically a hierarchical approach which is based on a sparse representation of object boundary shape. Then, a verification stage is used to re-rank the hypotheses.

Assuming that contour-based object detection can be formulated as a matching problem between model contour parts and image edge fragments, [72] proposed to solve the problem by finding dominant sets in weighted graphs; the weights are based on shape similarity. Since this approach is based on shape similarity, it can determine an optimal scale of the detected object without the common evaluation of all possible scales. Other work [73] proposed clustering based on the occurrence of patterns of edges instead of visual similarity to create a dictionary of meaningful contours; in other words, contours are matched if they occur similarly in a number of training images.

In an attempt to implement holistic matching of shapes with a large spatial extent, [75] has formulated object detection as a many-to-many fragment matching problem by using a contour grouping method to obtain long, salient matching candidates, which are then compared using standard Shape

Context descriptors. The large number of possible matches is handled by encoding the shape descriptors algebraically in a linear form, where optimisation is done by linear programming. To evaluate the distance to image segments, they used hand-drawn models.

The idea of explicitly extracting fragments which appear frequently in positive training images of a class, but seldom in negative ones, was explored by Shotton et al. [61]; they learned class-specific boundary fragments and their spatial arrangement as parts of a star-shaped configuration. In addition to their own local shape, such fragments include a pointer to the object centre, enabling object localisation in novel images by using a voting scheme. Also, they employed boosting to select fragments from a large pool of candidates. These candidates were constructed by using random rectangles sampled from training segmentation masks. The classifier was explicitly trained against clutter to improve the performance. Finally, Chamfer matching was used to validate the hypotheses. The work presented in this thesis shares the idea of selecting fragments that occur frequently in positive training images, but employs a completely different implementation.

Other work has shown that objects can be detected accurately in images using simple model sketches and by building a contour segment network, finding paths that resemble the model chains [18]. Ferrari et al. [18] start by partitioning image edges of the object model into groups of adjacent contour segments. To match these models, they find paths through the segments in the Berkeley edge maps which resemble the outline of the modelled categories. In later work [16], they learned a codebook for Pairs of Adjacent Contours (PAS) directly from cropped training images without any segmented examples and constructed shape models automatically. In order to localise the model in cluttered images, they combined Hough-based centre voting and

non-rigid thin-plate matching techniques. The thin-plate spline model allows for a global affine transformation of the shape while also allowing some local deviations from the affine model for each straight contour segment.

To expand their previous research, they presented in [19] a family of scale-invariant local shape features formed by short chains of connected straight contour segments which they call **kAS**, capable of cleanly encoding fragments of an object boundary. These are then matched to the Berkeley edge map of a query image to detect objects using the sliding window approach. The method offered an attractive compromise between information content and repeatability, and encompassed a wide variety of local shape structures. Finally, they integrated their impressive effort in [17] by learning shape models directly from training images, using only bounding boxes without segmented examples. Then objects are detected and localised at the boundary level, integrating Hough-style voting with a non-rigid point matching (thin-plate spline).

Many recent techniques attempt to segment shapes into visually meaningful parts. Although this approach generated impressive results, these techniques have only focused on relatively simple shapes, such as those composed of a single object either without holes or with a few simple holes. In many applications, shapes created from images can contain many overlapping objects and holes. Liu et al. [33] proposed a new decomposition method, called Dual-space Decomposition, that handles complex 2D shapes by recognising the importance of holes and classifying them as either topological noise or structurally important.

2.2.2 Biological Models

Inspired by the unmatched level of performance and speed of primate visual systems, researchers have developed computational models to replicate and employ hierarchical process of object recognition in the visual system. Since shape is a primary clue in object recognition [5], several biological models explored the possible representation of shapes, starting from the extraction of simple lines and edges at V1 in early vision to the description of the object itself using complex relations between contour parts in inferior temporal cortex, and employing intermediate representations of contour parts at V4.

To model line and edge detection in cortical area V1, [53] developed multi-scale models with no free parameters based on responses of simple and complex cells. In [54], the authors emphasised the importance of intermediate 2D shape representation by providing a biologically plausible model which incorporates intermediate layers of visual representation. They proposed that end-stopped and curvature cells are important for shape selectivity.

In a study of the role of area V4 in processing shape information, [48] investigated cell responses to contour features like angles and curves, which are proposed to be intermediate shape primitives by many theorists. Such investigations are important for understanding the transformation from low-level orientation signals to complex object representations. A quantitative model developed in [7] provided a plausible mechanism for shape representation in V4. They considered in their proposed model the selectivity of neurons in area V4 to complex boundary shapes and invariance to spatial translation.

The roles of end-stopping and curvature in intermediate layers of visual representation were modelled in [55]. Shape selectivity in that model is achieved by integrating end-stopped and curvature computations in a hierarchical representation of 2D shape; see Fig. 2.3. This model shows the

critical role that end-stopped neurons play in achieving shape selectivity.

The work presented in this thesis is based on nearest-neighbour lookups, which can be done efficiently using the K-D tree implementation provided by the FLANN library [44]. In order to avoid generating a huge number of hypotheses by using the sliding window technique as used in many algorithms (e.g. [19]), here we will use a Bayesian criterion to determine the discriminative power of each segment in a query image and let the most discriminative segments vote for potential bounding boxes. Since closed contours cannot be obtained in complex scenes, Shape Context [3] will be used to describe the individual contour fragments instead of the overall shape. We will avoid manual selection of high-curvature points in contours, which is applied by some methods like [50], by using a discrimination criterion, where discriminative segments by definition should have distinctive properties which are specific to the class that the object belongs to. Also, a fast method will be proposed to extract meaningful model segments, by finding edge fragments that appear frequently in positive images of the same class. The same idea was employed by Shotton et al. [61] but with a completely different implementation.

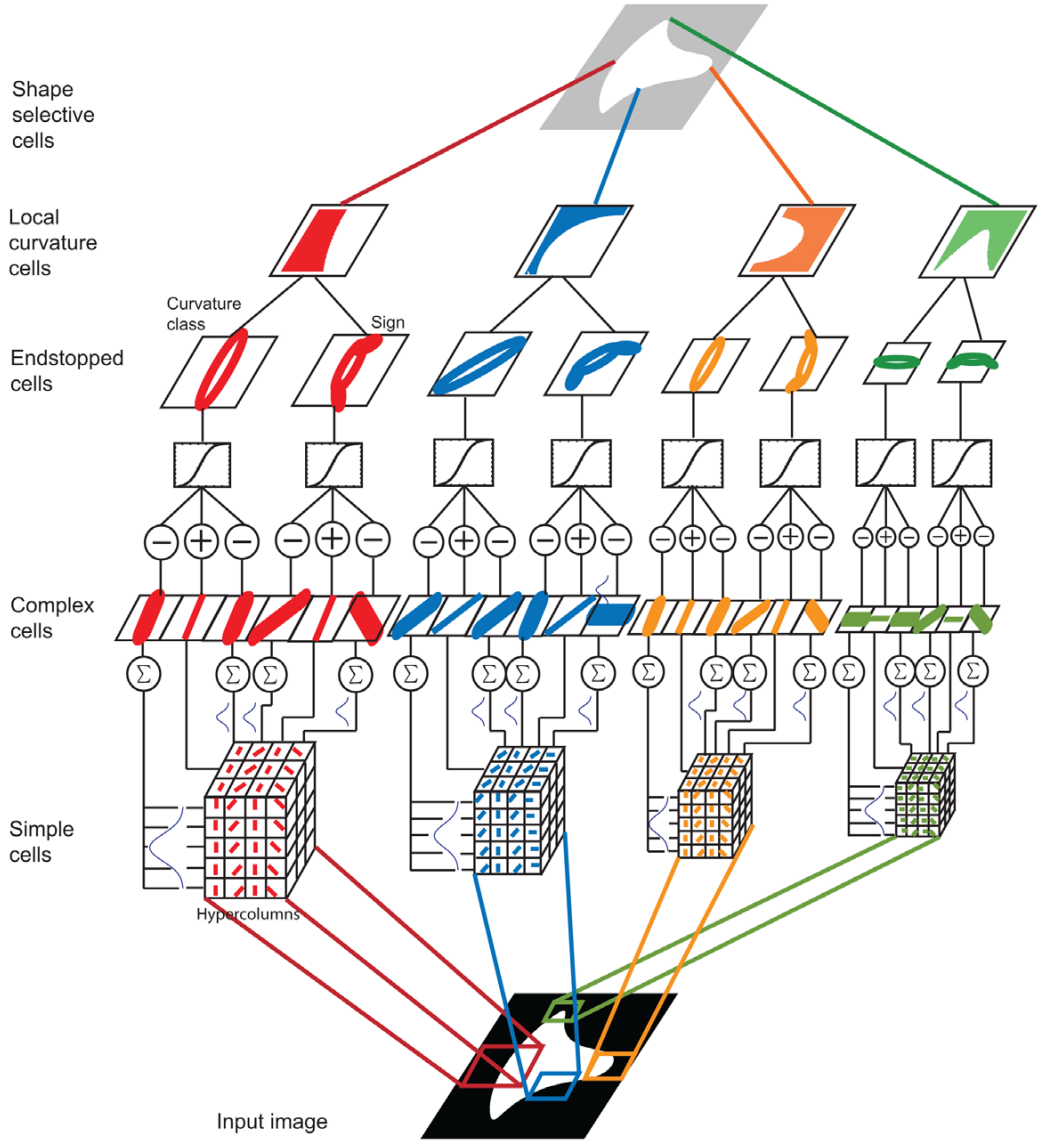


Figure 2.3: This is the system proposed by [55]. They proposed an architecture of the representational and computational system for the detection of 2D shapes.

Chapter 3

Contour-Based Object Detection and Recognition

The work in this chapter is mostly inspired by biological vision in terms of (1) using contour parts with certain curvatures as triggers to drive the attention for a more precise search, (2) using a coding scheme similar to log-polar mapping in V4 to describe shapes of contour fragments, and (3) using the absolute distance from contour fragment to shape centre as important information that helps relating fragments of the same contour together. In the next chapter, a further step will be taken towards a fully biological implementation.

Two important metrics in the evaluation of object detection and classification algorithms are precision, which is the percentage of the correct detections among all generated hypotheses by the algorithm and recall, which is the percentage of correctly detected objects. Most proposed algorithms for object detection and recognition consist of two basic steps: (i) detection, which identifies potential object locations and sets the upper bound on achievable recall; see Eqn. 5.2, and (ii) verification, which eliminates false

positive detections and sets the upper bound on precision; see Eqn. 5.1. Many modern algorithms implement the detection stage by using a sliding window at varying scales, and rely on a powerful binary classifier to eliminate false detections. This approach works well if the binary classifier is accurate, but it also results in slower recognition. In order to avoid that delay, here we will use a Bayesian criterion to determine the discriminative power of each segment in a query image in order to let the most discriminative segments vote for potential bounding boxes. Also, we try to locate meaningful segments which most likely correspond to object contours.

The work presented in this thesis is based on nearest-neighbour lookups, which can be done efficiently using K-D trees implementation provided by the FLANN library for nearest-neighbour lookups [44]. Recent work in image classification has exploited the fact that conditional class probabilities can be well approximated by the Euclidean distance to the nearest feature belonging to the correct class [6], as shown below.

Given a query image q represented by a set of local features d and a set of classes C, it can be classified as belonging to class $c \in C$ according to the conditional probability

$$c = \underset{C}{\operatorname{argmax}}\ p(C|q). \tag{3.1}$$

By applying Bayes' theorem and assuming a uniform prior probability over classes we obtain

$$c = \underset{C}{\operatorname{argmax}}\ p(q|C). \tag{3.2}$$

If the n descriptors d_i, extracted from image q, are assumed to be independent, the equation can be re-written as

$$c = \underset{C}{\operatorname{argmax}}\ \left[\log(\prod_{i=1}^{n} p(d_i|C))\right] \tag{3.3}$$

$$= \underset{C}{\operatorname{argmax}} \left[\sum_{i=1}^{n} \log p(d_i|C) \right]. \tag{3.4}$$

The probability $p(d_i|C)$ in Eqn. 3.4 can be approximated using a Parzen window estimator, with kernel K, i.e.,

$$\hat{p}(d_i|C) = \frac{1}{L} \sum_{j=1}^{L} K(d_i - d_j^c), \tag{3.5}$$

where L is the number of descriptors that belong to class c in the training set, and d_j^c is the j-th nearest descriptor to d_i in class c. A further approximation can be done by using only the r nearest-neighbours,

$$\hat{p}_r(d_i|C) = \frac{1}{L} \sum_{j=1}^{r} K(d_i - d_j^c). \tag{3.6}$$

It can be approximated further by considering only the single nearest-neighbour ($\mathrm{NN}_c(d_i)$) by setting r to 1:

$$\hat{p}_1(d_i|C) = \frac{1}{L} K(d_i - \mathrm{NN}_c(d_i)). \tag{3.7}$$

Substituting Eqn. 3.7 into Eqn. 3.4 and using a Gaussian kernel for K gives

$$c = \underset{C}{\operatorname{argmax}} \left[\sum_{i=1}^{n} \log \frac{1}{L} e^{-\frac{1}{2\sigma^2} \|d_i - \mathrm{NN}_c(d_i)\|^2} \right] \tag{3.8}$$

$$= \underset{C}{\operatorname{argmin}} \left[\sum_{i=1}^{n} \| \, d_i - \mathrm{NN}_c(d_i) \, \|^2 \right], \tag{3.9}$$

where (log) is the natural logarithm.

The last Eqn. 3.9 shows that conditional class probabilities can be approximated by the Euclidean distance to the nearest feature belonging to the

correct class. In other words, it suffices to find the class with the minimum Euclidean distance of its features to those of the query image.

This approximation performs well as long as there is a large number of training features. Instead of trying to discover meaningful rules for merging and splitting segments, our approach will rely on statistics present in training images to discover meaningful segments.

In a nutshell, the method performs the following steps: (i) extraction of many segments from training and testing images; (ii) for each segment in a test image, find the k nearest-neighbours among training segments; (iii) keeping a small percentage of segments after applying a Bayesian discrimination criterion; (iv) creating bounding-box hypotheses from each of the selected segments; and (v) classifying the hypotheses as object or background based on k nearest-neighbours.

3.1 Feature Extraction

Contour extraction in natural images is a difficult task. We start with the results of the excellent Berkeley edge detector [40], which are provided as part of the ETHZ dataset [18] and are also used as a starting point by most other algorithms. This is a bottom-up approach, so the extracted edges do not always correspond to object boundaries.

In order to provide sufficient training samples for reliable nearest-neighbour lookups, a large number of overlapping segments is extracted to ensure that each segment in a test image can have a close match; see Fig. 3.1. The approach is simple: contour-following at each edge pixel is applied, in order to extract many possible segments of many possible lengths, the pseudo-code for segment extraction is shown in Algorithm 1. Fig. 3.2 demonstrates the

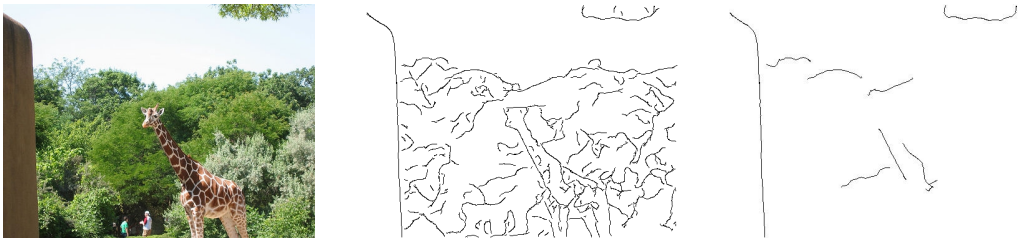
(a) Original segment.

(b) First overlapping sub-segment.

(c) Second overlapping sub-segment.

(d) Third overlapping sub-segment.

Figure 3.1: Segment overlapping is necessary to enable many possible matches.



(a) An image of a giraffe in a complex scene.

(b) Edge segments longer than 3 pixels.

(c) Edge segments longer than 100 pixels.

Figure 3.2: Long contours of objects in complex scenes can be fragmented into many very short segments.

importance of short segments in complex scenes. Therefore, after counting the number of pixels of each edge fragment, only fragments with a length of at least 20 pixels are kept because of Shape Context (see below). Larger fragments are split into smaller but partially overlapping fragments, but always with at least 20 pixels. For example, if a fragment counts 100 pixels, three fragments of 50 pixels are created, at the two ends and in the middle. In case of bifurcations, all possible paths are considered as long as the resulting fragments have at least 20 pixels. During training, only annotated regions containing objects are used.

The extraction of overlapping segments of different lengths is important

Algorithm 1 Extraction of all segments starting at a pixel location.

input : starting point $pt_0 = [x_0, y_0]$
output: set of all segments starting with pt_0
AllSegments $\leftarrow$ []
ActiveSegments $\leftarrow$ []
$S \leftarrow [pt_0]$
ActiveSegments $\leftarrow$ [ActiveSegments; S]
repeat
  NextActiveSegments $\leftarrow$ []
  for $S \in$ ActiveSegments **do**
    for $pt \in$ neighbours($S[last]$) **do**
      if pt is an edge pixel **and** $pt \notin S$ **then**
        $S' \leftarrow [S, pt]$
        AllSegments $\leftarrow$ [AllSegments; S']
        NextActiveSegments $\leftarrow$ [NextActiveSegments; S']
      end if
    end for
  end for
  ActiveSegments $\leftarrow$ NextActiveSegments
until ActiveSegments $= \emptyset$
return AllSegments

because it ensures that parts of long contours can be matched to short contour fragments. Although this method generates a large number of segments, this is not a problem because a discriminative power criterion will be introduced later for measuring how meaningful each segment is. In practice, the algorithm works well even with a small subset of segments.

The algorithm presented in this thesis does not attempt to split segments into sections between junctions like in [50], because there are many junctions between object contours and background segments. Also, Gestalt grouping principles are not applied. Instead, the available partial segments are considered as they are and we will try to find the best possible explanation for each of them. We intentionally kept feature extraction simple in order to find out how much information is actually present in the data, before introducing additional knowledge.

For each extracted segment, a feature vector is calculated. This starts by computing the Shape Context parameters [3] based on 20 equidistant points on the segment, and by concatenating them into one long descriptor. After careful experimentation, the following parameters are used: 12 orientation intervals of 30 degrees each, 3 distances that increase logarithmically starting from the point, and a maximum distance that covers 80% of segment length; see Fig 3.3. These parameters give descriptors of 720 bins in total, many bins being empty (zero) of course. Since each segment is sampled using a fixed number of equidistant points, the descriptors are size invariant. Rotation-invariance can be achieved by rotating each segment so that the ending points are aligned to the x-axis before calculating the Shape Context descriptor. We note that rotation-invariance has not been included when validating the method on the ETHZ dataset. Below is the procedure of calculating Shape Context:

1. Select N equally spaced points along the segment ($N = 20$), including the end points.

2. Calculate the vectors from each selected point p_i to the other points $p_{j \neq i}$.

3. Create a histogram h_i of vectors for each point

$$h_i = \#\{p_j \neq p_i : (p_j - p_i) \in \text{bin}(k)\},$$

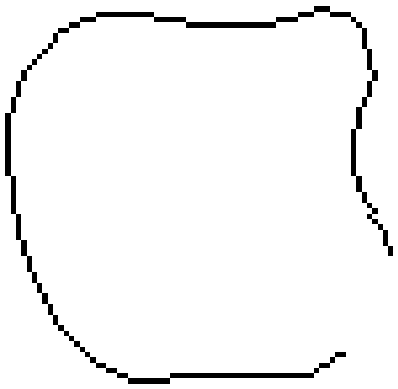
   where k is the number of bins in the histogram.

4. Concatenate the histograms of all points into one vector for each segment.

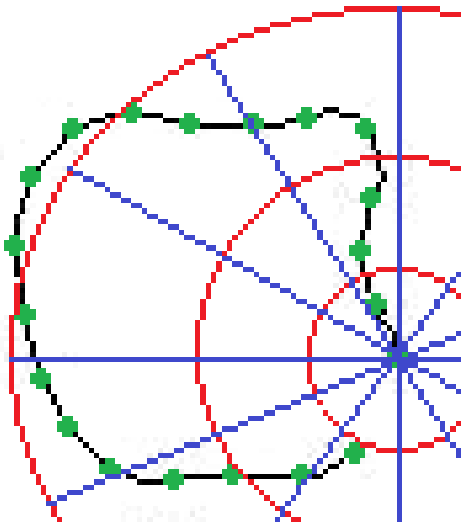
The offset from the centre of each segment to the centre of the bounding box in the training image is also added to the descriptor. This offset is used during hypothesis validation because each hypothesis has an estimated centre. During object detection, only Shape Context descriptors are used.

During training, the class of shape (object) which the segment belongs to is recorded. Its centre offsets from the top-left and bottom-right corners of the annotated object bounding box are also recorded. A background class is learned by extracting many segments from outside the bounding boxes of objects in all training images of all object classes. The descriptors of all collected segments are stored in an efficient K-D tree implementation provided by the FLANN library for nearest-neighbour lookups [44].

(a) Edge map of an Apple logo.



(b) Extracted segment from the edge map.



(c) Shape Context illustration. 20 equidistant points are distributed over the segment. Each point has a histogram which consists of 12 orientations and 3 logarithmic distances.

Figure 3.3: Segment extraction and feature calculation. This is only an illustration in case of a long contour segment. In reality such long segments are split into overlapping short segments.

3.2 Hypothesis Generation

Hypothesis generation also starts by extracting all segments from a test image and by calculating their feature vectors, as described in the previous section. Most of the extracted segments typically belong to the background, or do not help much in distinguishing between shapes (such as straight lines). Therefore, a Bayesian discrimination criterion is applied in order to select the most discriminative segments for a class:

$$D_c(d_i) := \frac{P(d_i \mid c)}{P(d_i \mid \bar{c})}, \tag{3.10}$$

where d_i is a feature descriptor associated with segment S_i of class c, and $P(d_i \mid \bar{c})$ is the probability that d_i was generated by any class except c.

The probabilities are estimated using Eqn. 3.9 and an approximate distance measure is obtained:

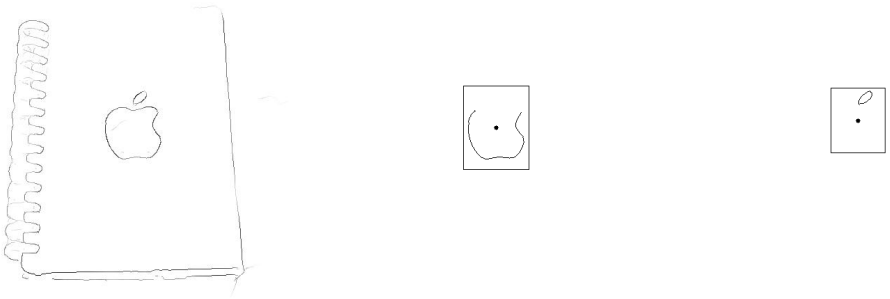
$$\log D_c(d_i) \approx \mathrm{dist}_c(d_i) - \mathrm{dist}_{\bar{c}}(d_i), \tag{3.11}$$

where

$$\mathrm{dist}_c = \| \, d_i - NN_c(d_i) \, \|^2; \qquad \mathrm{dist}_{\bar{c}} = \| \, d_i - \min_{\bar{c} \neq c} NN_{\bar{c}}(d_i) \, \|^2 \, .$$

For each segment extracted from a test image, the nearest neighbours of the training segments are determined in feature space using the FLANN library [44]. Then the class label of the nearest neighbour is assigned to the query segment. The discriminative power of each segment in the test image is calculated using Eqn. 3.11. The segments are then sorted by their discriminative power. Finally the most discriminative segments are kept.

Each training segment is associated with a class and a bounding box.



(a) Edge map of an Apple logo image.

(b) First discriminative segment voting for the Apple logo.

(c) Second discriminative segment voting for the Apple logo.

Figure 3.4: Different parts of the contour should vote for the same object, such hypotheses overlaps can be removed easily.

During training, the offset of the central point of each segment from the top-left and bottom-right corners of the bounding box containing the object is stored. During detection, this information is used to create an object hypothesis consisting of a bounding box and a class label for each discriminative segment; see Algorithm 2.

A hypothesis which overlaps with another hypothesis of the same class by more than 90% (the area of the intersection of the bounding boxes over the area of the union) is removed to reduce the number of hypotheses without impacting the recall. This can happen because many parts of the contour should vote for the same object, as illustrated in Fig. 3.4. Nevertheless, overlapping segments are needed to guarantee the existence of a segment's match despite the variation of contour fragments' lengths in each object instance.

Since each single segment can generate a hypothesis, two restrictions are imposed to eliminate meaningless hypotheses: hypotheses that have less than 50% overlap with the test image are eliminated, which can occur at the image border, and hypotheses with very few segments are assumed to be empty and

ignored. These two restrictions have absolutely no effect on the detection result and significantly reduce the number of generated hypotheses.

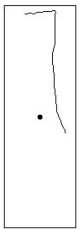
The process described above guarantees that segments triggering hypotheses are discriminative and meaningful, but they can still be false positives because the existence of one part of the contour does not necessarily mean that there is an object; see Fig. 3.5. The detection step results in about 16 hypotheses per class per image on average on the ETHZ dataset, without missing any object, but precision is still bad. The number of generated hypotheses depends mostly on the number of segments in the image and how many discriminative segments are kept.

Algorithm 2 Hypothesis generation for all classes.

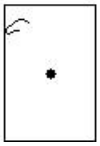
input : all image segments ImgSegments, all training segments TrSegments, all training bounding boxes TrBoxes
output: set of all hypotheses
AllHypotheses $\leftarrow$ []
for $S \in$ ImgSegments **do**
  Neighbours $\leftarrow$ kNearestNeighbours(S, TrSegments)
  NearestSegment $\leftarrow$ Neighbours[0]
  Class $\leftarrow$ class(NearestSegment)
  NearestOther $\leftarrow$ Neighbours[i], where class(Neighbours[i] $\neq$ Class)
  Discriminative power $\leftarrow$ $dist$(NearestSegment)^2 $-$ $dist$(NearestOther)^2
  BoundingBox $\leftarrow$ TrBoxes(NearestSegment) scaled to match S
  CurrentHypothesis $\leftarrow$ S, Class, BoundingBox, Discriminative power
  AllHypotheses $\leftarrow$ [AllHypotheses; CurrentHypothesis]
end for
GoodHypotheses $\leftarrow$ percentage of most discriminative hypotheses
return GoodHypotheses

3.3 Validating Hypotheses

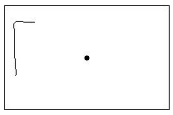
The final step of the algorithm is to process all object hypotheses and to try to discard any false positives. For each hypothesis generated in a test image



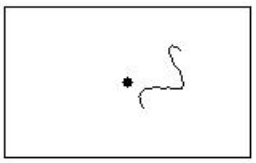
(a) False positive due to a bottle-like segment corresponding to the right part of a bottle.



(b) False positive due to a giraffe-like segment corresponding to the head of a giraffe.



(c) False positive due to a mug-like segment corresponding to the upper-left part of a mug.



(d) False positive due to a swan-like segment corresponding to the upper part of a swan.

Figure 3.5: False positives can occur during the detection process because each single segment can generate a hypothesis. None of the hypothesised objects shown here exist in the query image.

by the previous step, all segments within the estimated bounding box are selected and the hypothesis is classified as object or background, depending on a simple strength measure.

For each segment selected from the hypothesis, the number of neighbours of training segments in feature space of the same class label given to the hypothesis is counted, as is the number of neighbours of any other class label. The ratio of these two numbers represents the strength value of this hypothesis:

$$\text{Strength-kNN}_h^c = \frac{\sum_{i=0}^{k} N_i^c}{\sum_{i=0}^{k} N_i^{\bar{c}\neq c}}. \tag{3.12}$$

Strength-kNN_h^c is the strength measure of hypothesis h of class c, and k is the number of neighbours of each segment. N_i^c is one if the neighbour is of the same class label of the hypothesis and is zero otherwise. $N_j^{\bar{c}}$ is zero if the neighbour is of the same class label of the hypothesis and is one otherwise.

All hypotheses which overlap with a stronger hypothesis of the same class by more than 50% (PASCAL criterion) are eliminated, such that only one object of a particular class is allowed to exist at a given position in the image. Finally, all the hypotheses are sorted by decreasing strength. A threshold is applied to the remaining hypotheses, and this threshold is the free parameter of this method which can be used for balancing precision and recall.

3.4 Learning Model Segments

Many researchers use hand-drawn models to estimate kernel parameters for classification (e.g. [45]) or to match them directly to the query images using template matching techniques like Chamfer Matching [34, 61]. Such models do not cope well with shape deformations and intra-class variations

because one model cannot represent a whole class in most cases; see Fig. 3.6. Therefore, some authors prefer to learn models directly from training images (e.g. [17]).

Learning shape models from training images is a very complicated and slow part in most state-of-the-art algorithms. Also, these models often do not provide information about important contour parts in each class. A simple, fast and efficient method is proposed here to extract meaningful model segments by finding edge fragments which appear frequently in positive images of the same class.

Meaningful model segments are extracted from training images using nearest-neighbour lookups. An identification number is assigned to each object in the training images. This identification number is attached to the training segments during the extraction process. After extracting segments from training images as described earlier, the most similar segments are determined for each training segment using the FLANN library [44]. Then the number of nearest neighbours in descriptor space with the same class label but from different objects is incremented, until a neighbour with a different class label is found. Finally, training segments are sorted according to their number of neighbours and only the top 2% are used to create the models. This very low percentage of segments indicates the huge amount of clutter inside the bounding boxes of training images and the powerful selectivity of this simple method; see Fig. 3.7.

Since models extracted in this work consist of many individual segments, important parts of object contours can be recognised and learned easily with many possible deformations. These models can be used directly by matching model segments to test segments. Each class model consists of the most frequently occurring and most similar segments in the bounding boxes of

training images. Therefore, the sum of distances of test segments of a hypothesis to the model segments of the same class can be used as a measure of strength for that hypothesis. These distances are normalised by the number of segments of the hypothesis:

$$\text{Strength-Models}_h^c = \frac{1}{\text{Average-Distance}_h^c}, \tag{3.13}$$

$$\text{Average-Distance}_h^c = \frac{\sum_{i=0}^{n} \text{dist}_i^c}{n}, \tag{3.14}$$

where n is the number of segments of the hypothesis, dist_i^c is the distance of the ith segment in the hypothesis of class c to the nearest model segment of the same class and $\text{Average-Distance}_h^c$ is the average distance of hypothesis h of class c to the model segments of the same class.

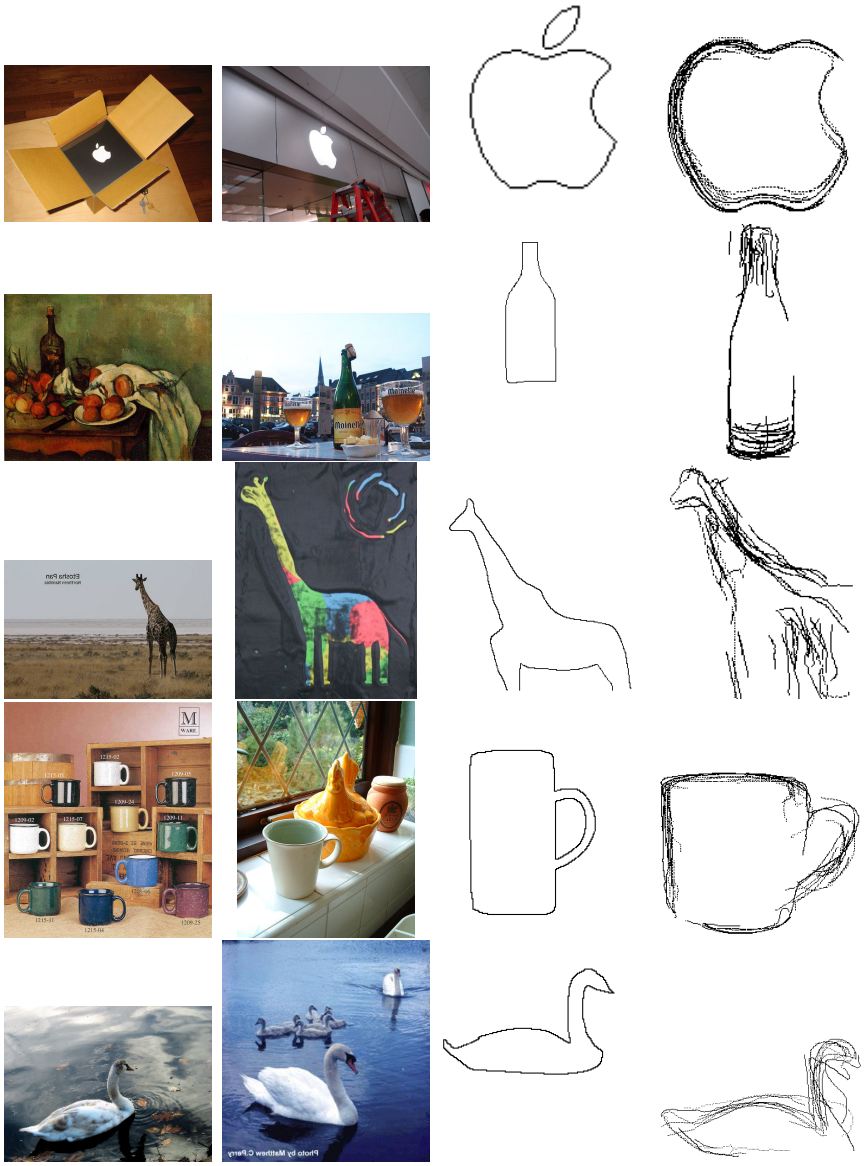


Figure 3.6: A single hand-drawn model cannot cope with shape deformations and intra-class variations. The first and second column show two examples of each class. The third column shows the hand-drawn model of each class as provided in the ETHZ dataset [18]. The fourth column shows model segments for each class as generated by the method proposed in this thesis. The model segments are scaled for better visualisation.

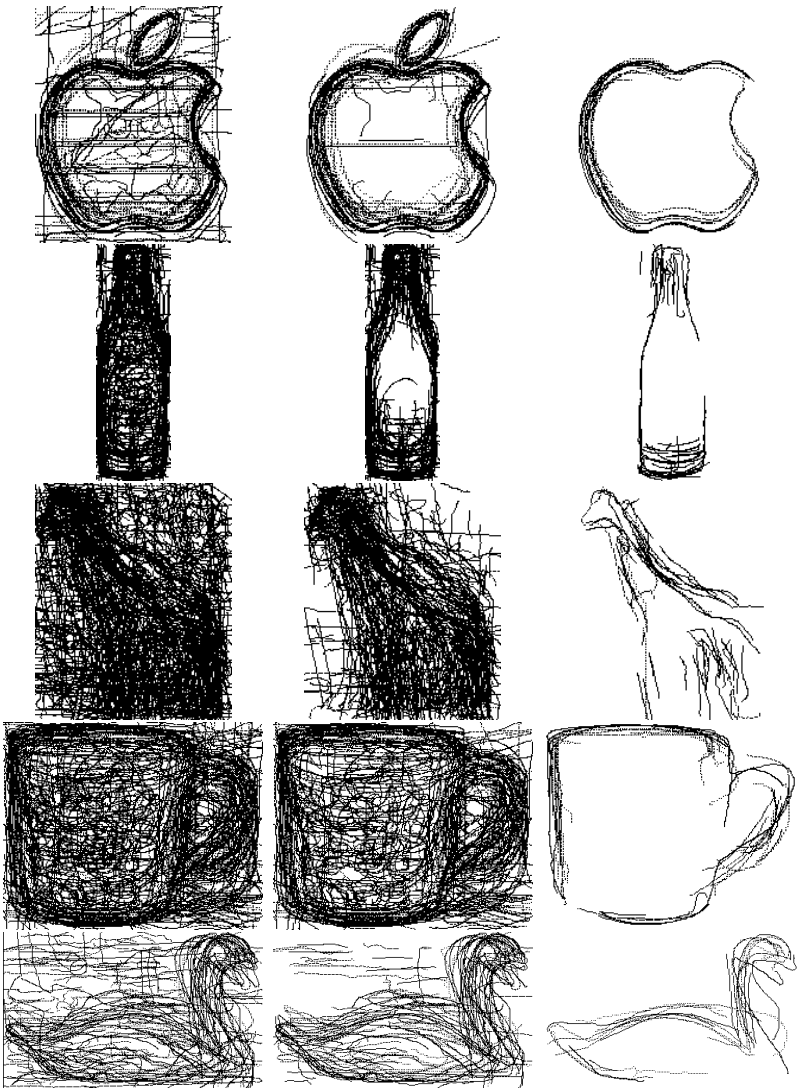


Figure 3.7: Most of the segments inside the bounding boxes of the annotated objects belong to clutter or background segments. The first column shows all the extracted segments from the annotated objects of the same class. The second column shows the top 50 % of the most frequent segments. The last column shows the top 2% of the segments. Segments are sorted and selected using the method described in the text. The segments are scaled for better visualisation.

Chapter 4

Towards a Fully Biologically Plausible Implementation

In the previous chapter, a method mostly motivated by biological vision has been presented to detect and recognise objects by their contour fragments. This chapter presents a further step towards a fully biological model of contour-based object detection and recognition in complex scenes.

A general and brief description is provided first about the visual system in primates, followed by an overview about shape coding of fragmented contours. Finally, a biologically inspired segment extraction method is proposed.

4.1 Contour Coding in Biological Vision

The work on the cat's visual cortex by Hubel and Wiesel [24] was an important motivation for Marr [39] and others to build visual hierarchies analogous to the primate visual system. Such computational modelling of biological vision can serve two goals: first, it provides a better understanding of the detailed wiring and functionality of the visual cortex; second, it gives in-

sight and inspiration for computer vision to build an active vision system that can be used in real-time applications, with the ability of general scene understanding.

4.1.1 How Does Biological Vision Process Visual Information?

Processing visual information in primates starts in the retinae of both eyes, then the signals are gathered in the Lateral Geniculate Nucleus (LGN) for further processing before they enter the visual cortex. This stage is called *pre-cortical processing*. In the visual cortex, *early vision* processing starts in the occipital part of the cortex which consists of areas V1-V4 and the middle temporal cortex (MT). Two interconnected streams emerge at this stage of processing: dorsal, which is referred to as the *where pathway*, and ventral, which is referred to as the *what pathway*. The ventral stream from V1 to IT is responsible for invariant object recognition. The dorsal stream deals with disparity, motion and attention.

Receptive fields of neurons in visual cortex increase in size gradually from very small spatial regions in V1 and V2 to half of the visual field in areas like TE. These receptive fields are retinotopically organised in early vision areas; in other words, adjacent neurons cover adjacent visual locations.

Such a hierarchical system is able to share information and low-level features in different visual tasks at the same time. A generic description of visual properties is extracted in areas V1-V4 and MT, which cover about 60% of the size of visual cortex [13]. This fact indicates the importance of generality and the concept of using basic perceptual entities as building blocks to perform different visual tasks. Furthermore, the concept of a deep hierarchy is linked to object recognition as neurophysiological evidence suggests [64].

In order to provide learning ability and adaptation to the system, visual representations increase gradually along the way up in the hierarchical structure, starting from the retina where no evidence of learning is observed [21] up to IT cortex where measurable changes even at a single-cell level have been observed due to learning and adaptation [32]. Such plasticity in later areas of the visual cortex allow primates to learn new objects and new representations of the same objects.

In order to pave the path for computer vision, authors in [30] provided a sufficient, and yet simple, description of the visual cortex in primates, so artificial vision's engineers can learn from and be inspired by the powerful hierarchy in biological vision (see Fig 1.4). The provided guidelines can be summarised as follows:

1. **Hierarchical processing** can be useful for:
    - Computational efficiency: in case of the availability of multiple processing units and GPUs.
    - Learning efficiency: because
        - Objects and scenes are hierarchical by nature so it will be easier to structure them in terms of their parts.
        - Appropriate features at a relatively high level will already be available in case of hierarchical processing.
2. **Separation of information channels** can be useful for:
    - Cases where some information channels are not available at all times.
    - Efficiency of representation: because

- Separate information channels result in a higher level of compactness.
- Integrated channels do not scale well to new objects.

3. **Feedback** can provide very useful capabilities to the system such as expectation, top-down reasoning, attention, imagination and filling in missing information.

4.1.2 How Does Biological Vision Process Shape Information?

When edges are available, they play the primary role in real-time object recognition even when other cues like colour and texture are present [5]. This shows the extreme importance of shape information in real-time systems. Shapes in complex scenes consist of fragmented contours, and the processing of these fragments starts in the extrastriate cortex (beyond V1) as individual contour parts with specific curvatures. In later areas, these fragments are grouped according to the to the centre of the objects they share. Rapid serial presentation experiments suggest that high accuracy in rapid object categorisation can be plausibly explained by a feedforward architecture, given the multi-stage processing scheme involved and average neural latencies [59]. Below are the most recently known details of shape coding in each area of the visual system:

1. **Retinal ganglion cell level:**

   A distinction is made between magnocellular (M) and parvocellular (P) streams [23]. Since P ganglion cells have small receptive fields and are responsible for high visual acuity, they are believed to be responsible for carrying shape information [26]. In order to perceive spatial changes,

centre-surround receptive fields are employed and these can be modelled by difference-of-Gaussian functions and used as edge detectors [22].

2. **V1:**

   Edges are the most meaningful features in natural scenes [12]; therefore, V1 is dominated by linear detectors of features like edges, lines and bars.

3. **V2:**

   To offer invariance of shape perception, a greater variety of cues is provided by cells sensitive to texture-defined contours [46] and to illusory contours [26]. On the other hand, cells sensitive to border ownership contribute to reducing missing and ambiguous visual information [27]. It has been found that 50% of the cells in V2 respond to the direction of the "owner" in shared boundaries [74], i.e., of simple geometric foreground shapes like rectangles.

4. **V4:**

   An object-centred representation of shape is constructed by cells sensitive to contour fragments with a certain curvature. The position of contour fragment relative to the centre of the shape is also coded in the cell responses [48].

5. **TEO (temporo-occipital cortex):**

   Curvature-tuned cells in TEO are invariant to position and size of contour fragments [46], which helps in integrating the information about shapes and relative positions of multiple contour fragments of a same object.

6. **TE (temporal cortex):**

   The primary stimulation of TE neurons is 2D shape [30], which is an extremely important clue for recognising objects. Therefore, it is essential for features calculated in this area to achieve both selectivity and invariance. Fulfilling such conflicting requirements can be achieved by separating variant and invariant information so that it can be used efficiently [10].

In summary, shape features in the visual system of primates start with simple spots in the retina, then bars and edges in V1. These bars and edges are combined to form contour fragments with certain curvatures which stimulate cells in V4. Finally, complex patterns and object parts are coded in the IT cortex. Such a gradual increase in feature complexity is gained by combining simpler stimuli like spots into more complex features like bars.

Despite the fact that receptive fields of cells in IT are very large in general, the size of their fields can be adapted to clutter in the scene to achieve better recognition [56]. Objects in a cluttered background or among other objects can result in relatively smaller receptive fields compared to the case where objects are on a blank background.

To avoid mixing features from different objects, edges that belong to the same 2D shape must be integrated. This mechanism of integrating elementary features rather than dealing with them as a collection of independent edges is known as the binding problem [69].

Since contour fragments in natural images cannot be analysed locally, a network of interacting neurons is needed to extract such extended contours. This results in *association fields* which differ from the classical fields in their rules of combining the outputs of orientation-sensitive neurons into a network [11].

4.2 Biological Segment Extraction

Inspired by biological vision and its hierarchical structure, a method of extracting meaningful segments from the fragmented contours in natural images is proposed here. It is based on the assumption that responses from neighbouring neurons are merged together if they satisfy two conditions: proximity and alignment.

At each layer of the extraction process, contour fragments grow longer and merge with adjacent fragments if they are within a certain distance and the difference of orientation between their endings is less than a certain threshold; see Fig. 4.1. The modelled receptive fields start by covering limited areas of the image, then grow gradually from small adjacent patches to an entire object. Information about contour parts within the field is passed from each layer to the next layer in a hierarchical scheme.

The proposed hierarchical process of segment extraction and binding involves many layers. The receptive field size of modelled association cells is doubled in each layer to cover increasingly larger spatial areas of the image. Segments extracted in each layer are passed to the following layer with coded information about the orientation of the ending points in each segment. Each layer then decides whether to merge segments with adjacent endings or not, depending on the orientation information of the segments.

The first layers in this model are similar to line and edge extraction in V1 and V2. Then these lines and edges are merged in the following layers to form contour fragments with specific curvatures, similar to the features extracted in V4. If any of these fragments satisfy the merging criteria mentioned above, they are merged to form discriminative object parts similar to the parts recognised in TEO; see Fig. 4.2. A complete shape structure then emerges from integrating these object parts in a process similar to what happens in

TE; see Fig. 4.3.

A coding scheme similar to log-polar mapping in V4 can be used to describe shapes of extracted contour fragments. Then the absolute distance from contour fragments to the shape centre can be used as important information that helps relating fragments of the same contour together to construct a complete shape structure.

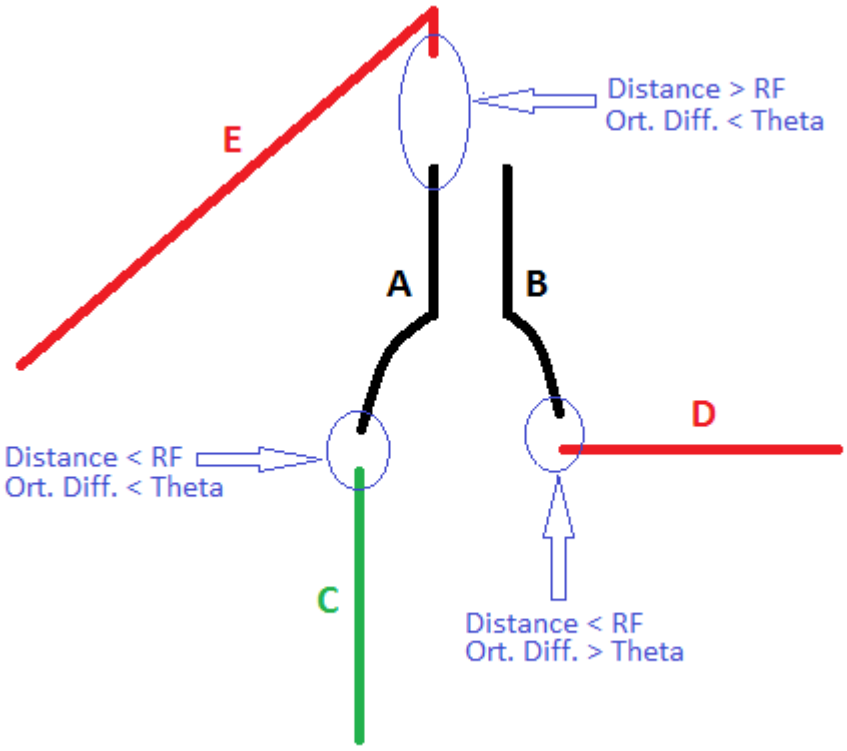
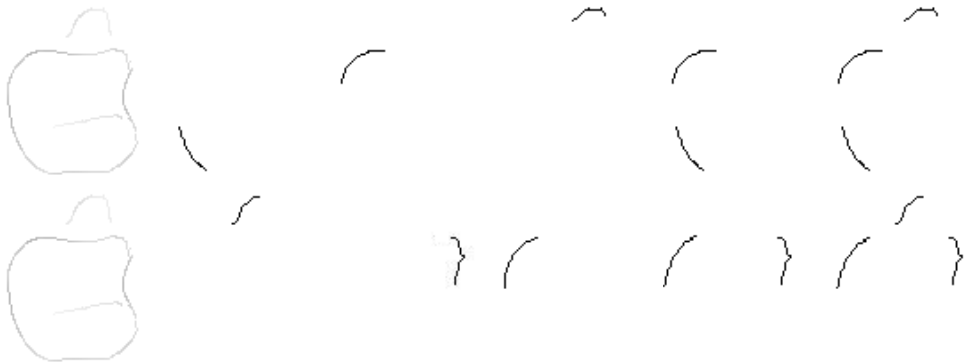


Figure 4.1: Illustration of segment merging criteria inspired by biological vision. Two constraints are applied: a distance threshold which represents the size of the association receptive field, and orientation difference between nearby segment endings. The green segment (C) should be merged with (A) while the red segments (D and E) are assumed to be not connected.



(a) Edge map. (b) Short edges. (c) Short edges. (d) Short edges. (e) Contour fragments. (f) Parts of object.

Figure 4.2: Different modelled cells merge different parts of the contour. This is useful to provide a description of all possible contour parts. The first column shows an edge map of the object. The second, third and fourth columns show illustration of short edges that can be extracted in the first layers. The fifth column shows contour fragments with specific curvatures. The sixth column shows object parts which result by merging contour fragments from previous layers.

(a) Edge map. (b) Object parts. (c) Shape structure.

Figure 4.3: A complete shape structure emerges from integrating object parts. The first column shows edge maps of the objects. The second column shows object parts that can be extracted by merging compatible contour fragments similar to the contour fragments recognised in TEO. The third column shows shape structures of entire objects similar to the structures recognised in TE.

Chapter 5

Evaluation

5.1 Validation on a Standard Dataset

To evaluate and measure the efficiency and performance of object detection and recognition algorithms, various datasets can be used, such as ETHZ, Caltech, PASCAL VOC, etc. Some of these datasets are especially suitable for the problem of contour-based object detection and recognition, because all the objects they contain are best defined by their shape.

The leading example is the ETHZ dataset [18] which contains 255 test images and features five distinct classes (Apple logos, bottles, giraffes, mugs and swans) in many kinds of scenes. As some images contain multiple instances, the objects appear 289 times in total. These images have been taken under varying, uncontrolled conditions, as they were collected from *Google* and *Flickr* search engines. Some of the images are paintings, drawings or computer renderings, but most of them are photographs. Objects in these images can appear at a wide range of scales; the scale difference can reach a factor of 6 for some classes, but are always seen with a consistent viewpoint from the side. The dataset also contains one hand-drawn model for each

class, which can be very suitable for some images, but quite different from the real shapes in other images. These models are not used in this thesis. In order to provide a common starting point for research, the dataset also contains an edge map (using the Berkeley edge detector [41]) and segmented regions for each of the 255 images. Fig. 5.1 shows some examples from this dataset.

Two important metrics in the evaluation of object detection and classification algorithms are precision and recall. Given the correct object label, scale and position (size and position of the bounding box) in the image, which is called ground truth (GT), and the resulting object label, scale and position from the applied recognition method (RM), four scenarios are possible, see Fig. 5.2: true positive (TP), true negative (TN), false positive (FP) and false negative (FN). Precision and recall factors can then be calculated by:

$$\text{Precision} = \frac{GT \cap RM}{RM} = \frac{TP}{RM}, \tag{5.1}$$

$$\text{Recall} = \frac{GT \cap RM}{GT} = \frac{TP}{GT}. \tag{5.2}$$

The standard benchmark procedure for the ETHZ dataset is to consider half of the images from each class as training instances and the remaining half as testing images. Although the creators of the ETHZ dataset did not mention anything about image selection criteria, often a random image selection for the training and testing sets is applied. Detection is defined using the PASCAL 50% size overlap criterion (area of the intersection of detected bounding boxes and ground truth annotations divided by the area of their union). The recall is normally calculated at 1.0 False Positive Per Image (FPPI) for the voting (detection) stage and 0.3/0.4 FPPI for the classifica-

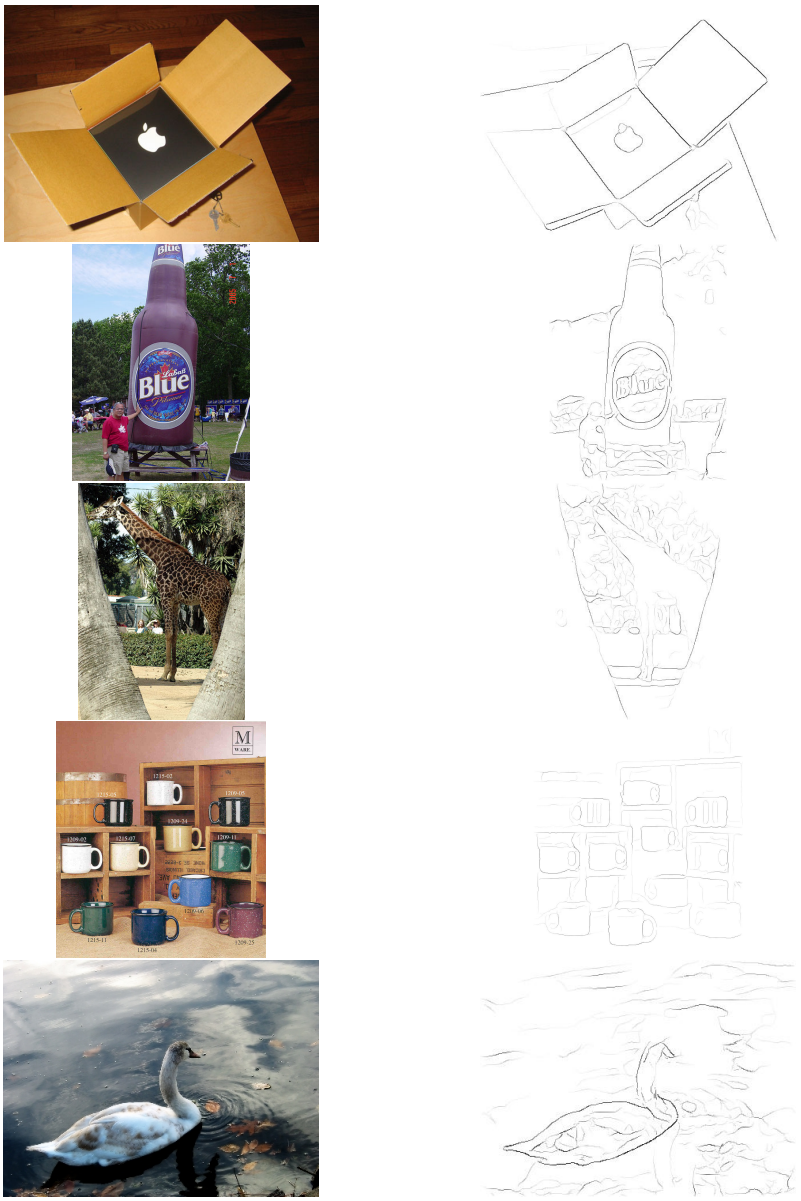


Figure 5.1: Examples from the ETHZ dataset [18] which contains five classes of objects in complex scenes. The first column shows images. The second column shows edge maps of the images.

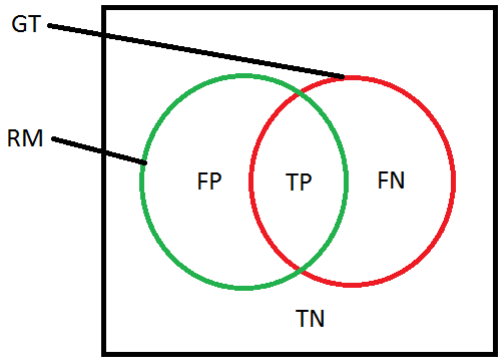


Figure 5.2: Illustration of evaluation metrics for object classification algorithms. A larger intersection area between the green circle (RM or recognition method) and the red circle (GT or ground truth) provides more true positive detections, which means a better performance.

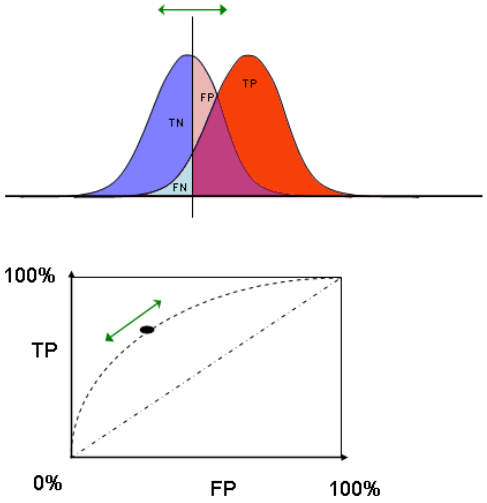
tion (hypotheses validation) stage.

To provide an idea about current classification rates, a comparison between the proposed method and other state-of-the-art algorithms on the ETHZ dataset is shown in Table 5.3.

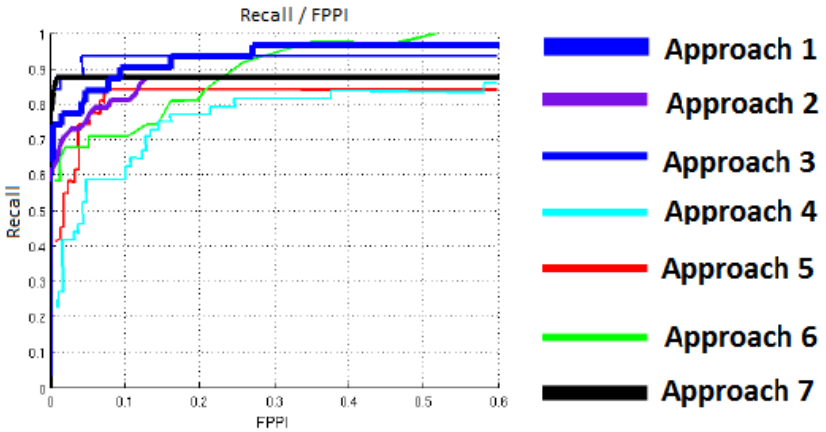
Curves can also be used to visually evaluate and compare the detection results and visualise the effect of changing parameters. The most used curves are recall/FPPI for detection and validation and Receiver Operating Characteristics (ROC) for binary classification. The ROC curve is simply a plot of TP versus FP rates; see Fig. 5.3.

5.2 Results

Although software optimisation was not taken into consideration, the implementation of the proposed method does not require a large amount of memory to store all the extracted segments and their feature vectors. Due to the simplicity of the algorithm, the processing time is a few seconds per



(a) Illustration of a Receiver Operating Characteristic (ROC) curve. The ROC curve is a plot of true positive versus false positive rates.



(b) Performance comparison in terms of recall/FPPI curves on one of ETHZ shape classes. reproduced from [73].

Figure 5.3: Different kinds of curves can be used to visualise the performance of object detection and recognition: ROC and recall/FPPI curves.

Figure 5.4: Examples of detected objects and their bounding boxes by using the Bayesian discrimination criterion. The black dots are the centres of the boxes. The first and second column show false positive detections. The third and fourth column show true positive detections.

image for the detection and validation phases. By using a multi-core system and an optimised implementation, the algorithm should be suitable for real-time applications, for example in robotics.

Examples of detected objects in the ETHZ dataset, using the Bayesian discrimination criterion, are shown in Fig. 5.4. Detections are represented by their bounding boxes at objects positions. Each detection is also shown in the original image.

Table 5.1 shows results of the proposed method at 0.3, 0.4 and 1.0 FPPI on the ETHZ dataset. These results were obtained by validating the hypotheses using the k nearest-neighbour method, i.e., final recall rates.

Table 5.1: Recall rates for the k nearest-neighbour method at 0.3, 0.4, and 1.0 FPPI on the ETHZ dataset.

Class	0.3 FPPI	0.4 FPPI	1.0 FPPI
Apple	75%	95%	100%
Bottle	25%	43%	64%
Giraffe	37.5%	43.7%	60.4%
Mug	48.3%	48.3%	67.7%
Swan	35.2%	35.2%	70%
Mean	44.2%	53%	72.4%

Table 5.2 shows recall rates for the proposed method at 0.3, 0.4 and 1.0 FPPI on the ETHZ dataset. These results were obtained by validating the hypotheses using model-segments method.

Table 5.2: Recall rates obtained by validating the hypotheses using the model-segments method at 0.3, 0.4, and 1.0 FPPI on the ETHZ dataset.

Class	0.3 FPPI	0.4 FPPI	1.0 FPPI
Apple	85%	90%	95%
Bottle	85.7%	85.7%	89.2%
Giraffe	68.7%	68.7%	77%
Mug	74.2%	74.2%	77.4%
Swan	76.4%	82.3%	82.3%
Mean	78%	80.2%	84.2%

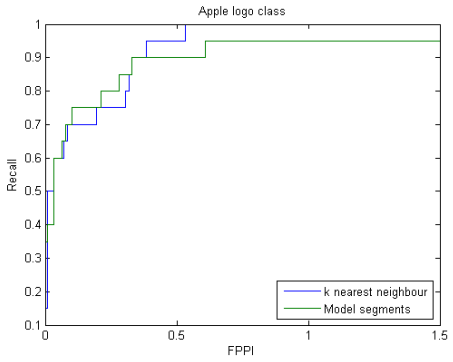
A comparison between the two proposed approaches, i.e., using k nearest-

neighbour and using model segments, is shown in Fig. 5.5 on the basis of Recall/FPPI curves. Validating hypotheses using model segments provides much better results for all classes, except for Apple logos but this is only for large FPPI rates.

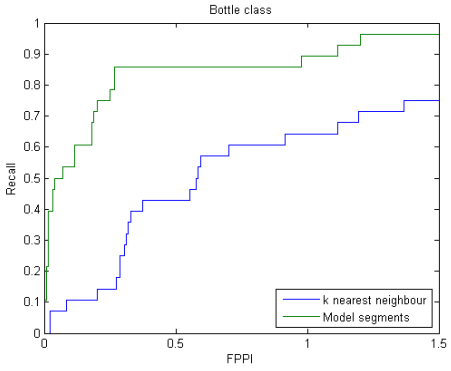
A comparison between the results of this work and the recent state of the art is provided in Table 5.3. Since the method proposed in this thesis is very simple and intuitive, it can be seen that our results are not yet comparable to the best results. However, the proposed method is very efficient in terms of resource consumption and very fast. In contrast to state-of-the-art methods, the proposed work only needs a very simple and fast training to provide very useful models which contain the most important segments in any given class.

Table 5.3: This table summarises recall rates of the proposed work (model segments method) and recent state-of-the-art methods at 0.3/0.4 FPPI on the ETHZ shape classes.

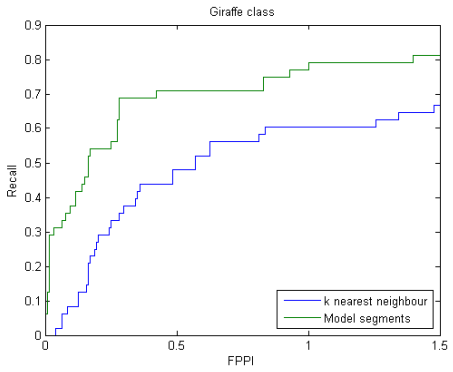
Method	Applelogos	Bottles	Giraffes	Mugs	Swans	**Mean**
[73] 2012	95/95	100/100	91.3/91.3	96.7/96.7	100/100	96.5/96.5
[68] 2010	100/100	96/97	86/91	90/91	98/100	94/96
[63] 2010	95/95	100/100	87.2/89.6	93.6/93.6	100/100	95.2/95.6
[15] 2010	95/95	96.3/100	84.7/84.7	96.7/96.7	94.1/94.1	93.3/94.1
[37] 2011	92/92	97.9/97.9	85.4/85.4	87.5/87.5	100/100	92.6/92.6
[38] 2009	95/95	92.9/96.4	89.6/89.6	93.6/96.7	88.2/88.2	91.9/93.2
[51] 2010	93.3/93.3	97/97	79.2/81.9	84.6/86.3	92.6/92.6	89.3/90.5
[45] 2009	95/95	89.3/89.3	70.5/75.4	87.3/90.3	94.1/94.1	87.2/88.8
[14] 2008	95/95	100/100	72.9/72.9	83.9/83.9	58.8/64.7	82.1/83.3
This work	85/90	85.7/85.7	68.7/68.7	74.2/74.2	76.4/82.3	78/80.2
[17] 2009	77.7/83.2	79.8/81.6	39.9/44.5	75.1/80	63.2/70.5	67.1/72



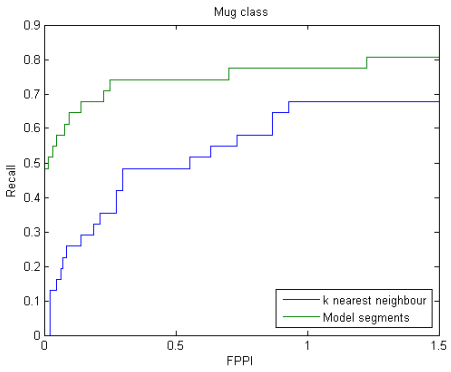
(a) Recall/FPPI for the Apple logo class



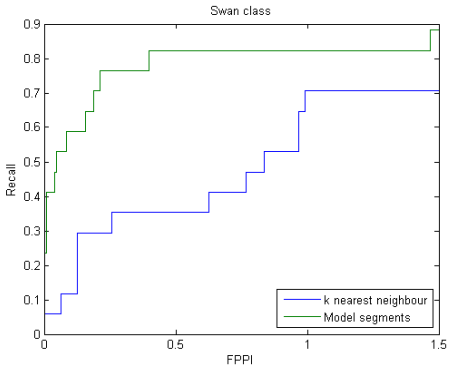
(b) Recall/FPPI for the Bottle class



(c) Recall/FPPI for the Giraffe class



(d) Recall/FPPI for the Mug class



(e) Recall/FPPI for the Swan class

Figure 5.5: A comparison between the two proposed approaches: using k nearest-neighbour and using model segments.

Until here we have applied only *one* random selection of the training and test sets for showing results in Tables 5.2 , 5.3 and in Fig. 5.5. The question is what happens when we repeat the random selection several times. The mean and standard deviation of recall rates are provided in Table 5.4 for all classes in the ETHZ dataset at 0.3 and 0.4 FPPI. The algorithm was executed 10 times. The results used in this table were obtained using model segments.

Table 5.4: The mean and standard deviation of recall rates for all the classes of the ETHZ dataset at 0.3 and 0.4 FPPI.

Class	0.3 FPPI	0.4 FPPI
Apple	80.5 ± 13.6	84.2 ± 7.5
Bottle	73.5 ± 7.4	74.7 ± 8.8
Giraffe	61.4 ± 7.7	64.2 ± 5.6
Mug	67.6 ± 8.0	71.8 ± 6.7
Swan	75.2 ± 9.2	80.3 ± 9.2

From Table 5.4 we can see that the actual random selection can have a huge impact on the results. The standard deviation does not even include the worst and the best result of each class (in fact, the distributions are likely not Gaussian). Unfortunately, the papers from the state-of-the-art methods listed in Table 5.3 do not even mention variability due to random selections, and it is possible that they only mention the best results. If we do this and take the mean plus one standard deviation, we could get 91.7/94.1 (the Apple logo class), 80.9/83.5 (the Bottle class), 68.1/69.8 (the Giraffe class), 75.6/78.5 (the Mug class) and 84.4/89.5 (the Swan class), with means of 80.14/83.08.

Most of the state-of-the-art methods in Table 5.3 focus on implementing descriptors of contour fragments and extracting models from training data.

In order to validate the generated hypotheses by the sliding window technique, most of them use either Fast Directional Chamfer Matching or an SVM.

Only one of the listed methods provided information about the system that has used to run the algorithm and the processing times of the training and testing stages. The system used in [15] was a desktop computer. All experiments were done on a 2.8 GHz 8-core Intel Xeon Mac Pro computer running Mac OS X 10.5. The system makes use of the multiple-core architecture for computing filter responses in parallel. It took about 20 hours to train all models and an average runtime per test image of around 2 seconds.

The system used to run the proposed algorithm in this work was a laptop computer, a 2.5 GHz 4-core Intel Core i5 Sony VAIO running the Windows 8 operating system. The algorithm used only about 3 GB of memory. It took only 3 minutes to extract training segments, calculate all feature vectors and extract all model segments. The average runtime per test image for the detection stage was 0.9 seconds. In the validation stage using model segments, it took about 28 seconds on average per test image to process all the generated hypotheses.

Chapter 6

Discussion and Future Work

6.1 Discussion

Although the proposed algorithm is simple in principle, there are several tunable parameters which can impact the performance. In general, decisions and parameters which make the algorithm simpler were always favoured. Such decisions also tended to work better in practice.

Concerning the Shape Context descriptors, standard parameters used in the original publication also provided the best results here. There was no need to compensate for any shape rotation, because object rotations in the ETHZ dataset are limited to small angles. The final descriptor contains two components: the Shape Context descriptor and the relative position to object centre, i.e., the offset value from the centre of the segment to the centre of the object. An intuitive balance between these two components was applied, each component contributing about 50% to the final descriptor. However, changing this factor has a small effect on results; a wide range of values can be used.

In order to reduce the number of generated hypotheses during the detec-

tion phase, a few intuitive rules have been applied. Any hypothesis with less than 50% overlap with the test image is eliminated, simply because this can only happen at the image border where no objects are expected. Very short segments are not allowed to trigger any hypothesis, because such segments can often be confused with background clutter. Finally, hypotheses should have a minimum number of edge pixels inside the detected bounding box in order to be considered; the minimum number of pixels can be easily learned from the training images. Nevertheless, a reasonable number of hypotheses can still be obtained using only the discrimination criterion.

The number of segments to be considered in the class models can vary from 1% to almost 5% of all the training segments of any class without having a noticeable impact on the final results. Since the models of the proposed method consist of many individual segments, they can cope with shape deformation and intra-class variation.

In summary, this thesis presented a simple and intuitive method for detecting objects in natural images. The proposed algorithm is extremely simple and easy to implement, yet it can already provide results which approach results of the state-of-the-art in terms of detection accuracy, and its learning-free nature in detection phase makes it considerably faster than most competing methods.

6.2 Future Work

The results seem to suggest that there is more to edge maps than meets the eye. Since the trivial algorithm performs quite well but is still at the bottom end of the recent state-of-the-art methods, and it is quite different from most published approaches, there should be significant room for improvement.

Unlike most of the state-of-the-art methods, models created in this work are not edge maps of a single-pixel width. If a proper filtering is applied to these models, the resulting masks can be used as a cost function to reward segments within an allowed range of spatial positions, and penalise other segments.

Getting a completely plausible biological model is a major goal for future work. The segment merging process must be improved to get as meaningful segments as possible. The Shape Context descriptors must be modified to mimic the coding of contour fragments in V4 using orientation, relative position and curvature information. The binding mechanism must be further developed to ensure a correct integration of shape parts in a similar way to the process as carried out in the IT cortex.

Bibliography

[1] M. Bar. "Visual objects in context". In: *Nature Reviews Neuroscience* 5.8 (2004), pp. 617–629.

[2] H. G. Barrow et al. *Parametric correspondence and chamfer matching: Two new techniques for image matching*. Tech. rep. DTIC Document, 1977.

[3] S. Belongie, J. Malik, and J. Puzicha. "Shape matching and object recognition using shape contexts". In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 24.4 (Apr. 2002), pp. 509–522. ISSN: 01628828. DOI: 10.1109/34.993558.

[4] J. Bengtsson. "Shape Based Recognition Cognitive Vision Systems in Trafic Safety Applications". In: *UPPAAL, A Tool for Automatic Verification of Real-Time Applications* 1395 (1996).

[5] I. Biederman and G. Ju. "Surface versus edge-based determinants of visual recognition". In: *Cognitive Psychology* 20.1 (1988), pp. 38–64.

[6] O. Boiman, E. Shechtman, and M. Irani. "In defense of nearest-neighbor based image classification". In: *2008 IEEE Conf. on Computer Vision and Pattern Recognition* (2008), pp. 1–8.

[7] C. Cadieu et al. "A model of V4 shape selectivity and invariance". In: *Journal of Neurophysiology* 98.3 (2007), pp. 1733–1750.

[8] D. Ciresan, U. Meier, and J. Schmidhuber. "Multi-column deep neural networks for image classification". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conf. on*. IEEE. 2012, pp. 3642–3649.

[9] G. Csurka et al. "Visual categorization with bags of keypoints". In: *Workshop on statistical learning in computer vision, ECCV,* vol. 1. 2004, p. 22.

[10] J. J. DiCarlo and D. D. Cox. "Untangling invariant object recognition". In: *Trends in Cognitive Sciences* 11.8 (2007), pp. 333–341.

[11] S. O. Dumoulin et al. “Contour extracting networks in early extrastriate cortex”. In: *Journal of Vision* 14.5 (2014), p. 18.

[12] J. H. Elder. “Are edges incomplete?” In: *Int. Journal of Computer Vision* 34.2-3 (1999), pp. 97–122.

[13] D. J. Felleman and D. C. Van Essen. “Distributed hierarchical processing in the primate cerebral cortex”. In: *Cerebral Cortex* 1.1 (1991), pp. 1–47.

[14] P. Felzenszwalb, D. McAllester, and D. Ramanan. “A discriminatively trained, multiscale, deformable part model”. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conf. on*. IEEE. 2008, pp. 1–8.

[15] P. F. Felzenszwalb et al. “Object detection with discriminatively trained part-based models”. In: *Pattern Analysis and Machine Intelligence, IEEE Trans. on* 32.9 (2010), pp. 1627–1645.

[16] V. Ferrari, F. Jurie, and C. Schmid. “Accurate object detection with deformable shape models learnt from images”. In: *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conf. on*. IEEE. 2007, pp. 1–8.

[17] V. Ferrari, F. Jurie, and C. Schmid. “From Images to Shape Models for Object Detection”. In: *Int. J. of Computer Vision* 87.3 (July 2009), pp. 284–303. ISSN: 0920-5691. DOI: 10.1007/s11263-009-0270-9.

[18] V. Ferrari, T. Tuytelaars, and L. Van Gool. “Object Detection by Contour Segment Networks”. In: *Computer Vision–ECCV 2006* section 4 (2006).

[19] V. Ferrari et al. “Groups of Adjacent Contour Segments for Object Detection”. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 30.1 (2008), pp. 0036–51.

[20] K. Fukushima. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. In: *Biological Cybernetics* 36.4 (1980), pp. 193–202.

[21] I. Gödecke and T. Bonhoeffer. “Development of identical orientation maps for two eyes without common visual experience”. In: *Nature* 379.6562 (1996), pp. 251–254.

[22] M. J. Hawken and A. J. Parker. “Spatial properties of neurons in the monkey striate cortex”. In: *Proceedings of the Royal society of London. Series B. Biological Sciences* 231.1263 (1987), pp. 251–288.

[23] D. H. Hubel and T. N. Wiesel. "Receptive fields and functional architecture of monkey striate cortex". In: *Journal of Physiology* 195.1 (1968), pp. 215–243.

[24] D. H. Hubel and T. N. Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". In: *Journal of Physiology* 160.1 (1962), p. 106.

[25] T. Joachims. *Text categorization with support vector machines: Learning with many relevant features*. Springer, 1998.

[26] E. R. Kandel, J. H. Schwartz, T. M. Jessell, et al. *Principles of neural science*. Vol. 4. McGraw-Hill New York, 2000.

[27] K. Koffka. *Principles of Gestalt psychology*. Routledge, 2013.

[28] P. Kontschieder et al. "Discriminative Learning of Contour Fragments for Object Detection." In: *BMVC*. 2011, pp. 1–12.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[30] N. Kruger et al. "Deep hierarchies in the primate visual cortex: What can we learn for computer vision?" In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.8 (2013), pp. 1847–1871.

[31] S. Lazebnik, C. Schmid, and J. Ponce. "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories". In: *2006 IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition - Vol. 2 (CVPR'06)* 2 (), pp. 2169–2178. DOI: 10.1109/CVPR.2006.68.

[32] N. Li and J. J. DiCarlo. "Unsupervised natural visual experience rapidly reshapes size-invariant object representation in inferior temporal cortex". In: *Neuron* 67.6 (2010), pp. 1062–1075.

[33] G. Liu, Z. Xi, and J. Lien. "Dual-Space Decomposition of 2D Complex Shapes". In: *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*. 2013, pp. 4154–4161.

[34] M. Liu et al. "Fast directional chamfer matching". In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conf. on*. IEEE. 2010, pp. 1696–1703.

[35] H. Lodhi et al. "Text classification using string kernels". In: *Journal of Machine Learning Research* 2 (2002), pp. 419–444.

[36] D. G. Lowe. "Distinctive image features from scale-invariant keypoints". In: *Int. Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[37] T. Ma and L. J. Latecki. “From partial shape matching through local deformation to robust global shape similarity for object detection”. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conf. on*. IEEE. 2011, pp. 1441–1448.

[38] S. Maji and J. Malik. “Object detection using a max-margin hough transform”. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conf. on*. IEEE. 2009, pp. 1038–1045.

[39] D. Marr and A. Vision. “A computational investigation into the human representation and processing of visual information”. In: *WH San Francisco: Freeman and Company* (1982).

[40] D. Martin et al. “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics”. In: *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE Int. Conf. on*. Vol. 2. IEEE. 2001, pp. 416–423.

[41] D.R. Martin, C.C. Fowlkes, and J. Malik. “Learning to detect natural image boundaries using local brightness, color, and texture cues.” In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 26.5 (May 2004), pp. 530–49. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2004.1273918.

[42] J. A. Martins et al. “Local Object Gist : Meaningful Shapes and Spatial Layout at a Very Early Stage of Visual Processing”. In: *Gestalt Theory* 34.3 (2012), pp. 349–380.

[43] S. McCann and D. G. Lowe. “Local Naive Bayes Nearest Neighbor for image classification”. In: *2012 IEEE Conf. on Computer Vision and Pattern Recognition* (June 2012), pp. 3650–3656. DOI: 10.1109/CVPR.2012.6248111.

[44] M. Muja and D. G. Lowe. “Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration.” In: *VISAPP (1)*. 2009, pp. 331–340.

[45] B. Ommer and J. Malik. “Multi-scale object detection by clustering lines”. In: *Computer Vision, 2009 IEEE 12th Int. Conf. on*. IEEE. 2009, pp. 484–491.

[46] G. A. Orban. “Higher order visual processing in macaque extrastriate cortex”. In: *Physiological Reviews* 88.1 (2008), pp. 59–89.

[47] G. Papari and N. Petkov. “Edge and line oriented contour detection: State of the art”. In: *Image and Vision Computing* 29.2 (2011), pp. 79–103.

[48] A. Pasupathy and C. Connor. “Responses to contour features in macaque area V4”. In: *Journal of Neurophysiology* 82.5 (1999), pp. 2490–2502.

[49] B. Pinna. “New Gestalt principles of perceptual organization: An extension from grouping to shape and meaning”. In: *Gestalt Theory* 32.1 (2010), p. 11.

[50] S. Ravishankar, A. Jain, and A. Mittal. “Multi-stage contour based detection of deformable objects”. In: *Computer Vision–ECCV 2008*. Springer, 2008, pp. 483–496.

[51] H. Riemenschneider, M. Donoser, and H. Bischof. “Using partial edge contour matches for efficient object category localization”. In: *Computer Vision–ECCV 2010*. Springer, 2010, pp. 29–42.

[52] M. Riesenhuber and T. Poggio. “Hierarchical models of object recognition in cortex”. In: *Nature Neuroscience* 2.11 (1999), pp. 1019–1025.

[53] J. Rodrigues and J. M. H. du Buf. “Multi-scale lines and edges in V1 and beyond: brightness, object categorization and recognition, and consciousness.” In: *Bio Systems* 95.3 (Mar. 2009), pp. 206–26. ISSN: 1872-8324. DOI: 10.1016/j.biosystems.2008.10.006.

[54] A. J. Rodriguez-Sanchez and J. K. Tsotsos. “The importance of intermediate representations for the modeling of 2d shape detection: Endstopping and curvature tuned computations”. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conf. on*. IEEE. 2011, pp. 4321–4326.

[55] A. J. Rodríguez-Sánchez and J. K. Tsotsos. “The roles of endstopped and curvature tuned computations in a hierarchical representation of 2D shape.” In: *PloS One* 7.8 (Jan. 2012), e42058. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0042058.

[56] E. T. Rolls, N. C. Aggelopoulos, and F. Zheng. “The receptive fields of inferior temporal cortex neurons in natural scenes”. In: *Journal of Neuroscience* 23.1 (2003), pp. 339–348.

[57] K. Schindler and D. Suter. “Object detection by global contour shape”. In: *Pattern Recognition* 41.12 (Dec. 2008), pp. 3736–3748. ISSN: 00313203. DOI: 10.1016/j.patcog.2008.05.025.

[58] E. L. Schwartz et al. “Shape recognition and inferior temporal neurons”. In: *Proceedings of the National Academy of Sciences* 80.18 (1983), pp. 5776–5778.

[59] T. Serre, A. Oliva, and T. Poggio. “A feedforward architecture accounts for rapid categorization.” In: *Proceedings of the National Academy of Sciences of the United States of America* 104.15 (Apr. 2007), pp. 6424–9. ISSN: 0027-8424. DOI: 10.1073/pnas.0700622104.

[60] T. Serre et al. “A quantitative theory of immediate visual recognition”. In: *Progress in Brain Research* 165 (2007), pp. 33–56.

[61] J. Shotton, A. Blake, and R. Cipolla. “Contour-based learning for object detection”. In: *Computer Vision, 2005. ICCV 2005. Tenth IEEE Int. Conf. on*. Vol. 1. IEEE. 2005, pp. 503–510.

[62] X. Shu and X. Wu. “A novel contour descriptor for 2D shape matching and its application to image retrieval”. In: *Image and Vision Computing* 29.4 (2011), pp. 286–294.

[63] P. Srinivasan, Q. Zhu, and J. Shi. “Many-to-one contour matching for describing and discriminating object shape”. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conf. on*. IEEE. 2010, pp. 1673–1680.

[64] J. B. Tenenbaum et al. “How to grow a mind: Statistics, structure, and abstraction”. In: *Science* 331.6022 (2011), pp. 1279–1285.

[65] K. Terzic. “A Generic Middle Layer for Image Understanding”. PhD thesis. Universität Hamburg, 2011. URL: http://ediss.sub.uni-hamburg.de/volltexte/2011/5412.

[66] K. Terzić et al. “Biological Models for Active Vision: Towards a Unified Architecture”. In: *Springer* 7963 (2013). Ed. by Mei Chen, Bastian Leibe, and Bernd Neumann, pp. 113–122.

[67] S. Tong and D. Koller. “Support vector machine active learning with applications to text classification”. In: *Journal of Machine Learning Research* 2 (2002), pp. 45–66.

[68] A. Toshev, B. Taskar, and K. Daniilidis. “Object detection via boundary structure segmentation”. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conf. on*. IEEE. 2010, pp. 950–957.

[69] A. Treisman. “The binding problem”. In: *Current Opinion in Neurobiology* 6.2 (1996), pp. 171–178.

[70] J. Wagemans et al. “A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure–ground organization.” In: *Psychological Bulletin* 138.6 (2012), p. 1172.

[71] M. Wertheimer. “Laws of organization in perceptual forms”. In: *A source book of Gestalt psychology* (1938), pp. 71–88.

[72] X. Yang, H. Liu, and L. J. Latecki. “Contour-based object detection as dominant set computation”. In: *Pattern Recognition* 45.5 (2012), pp. 1927–1936.

[73] P. Yarlagadda and B. Ommer. “From meaningful contours to discriminative object shape”. In: *Computer Vision–ECCV 2012* (2012), pp. 766–779.

[74] H. Zhou, H. S. Friedman, and R. Von Der Heydt. “Coding of border ownership in monkey visual cortex”. In: *Journal of Neuroscience* 20.17 (2000), pp. 6594–6611.

[75] Q. Zhu et al. “Contour context selection for object detection: A set-to-set contour matching approach”. In: *Computer Vision–ECCV 2008*. Springer, 2008, pp. 774–787.