Miguel Ângelo Pires Farrajota

Human pose and action recognition using neural networks

University of the Algarve

Faculty of Sciences and Technology

2017

Miguel Ângelo Pires Farrajota

Human pose and action recognition using neural networks

PhD. Thesis in Electronics and Telecomunications (Signal Processing)

Developed under supervision of:

Prof. Doutor Johannes Martinus Hubertina du Buf

Prof. Doutor João Miguel Fernandes Rodrigues

University of the Algarve

Faculty of Sciences and Technology

2017

DECLARATION

I hereby declare to be the author of this work, which is original with some parts already published. Authors and works consulted are properly cited in the text and appear in the included reference list.

Declaro ser o autor deste trabalho, que é original e contém partes já publicadas. Os autores e trabalhos consultados estão devidamente citados no texto e constam da listagem de referências incluída.

COPYRIGHT

Faro, 2017

Miguel Farrajota

Acknowledgments

First, many thanks goes to my supervisors Profs Hans du Buf and João Rodrigues, for their supervision based on their large experience, for providing a stimulating and cheerful research environment in their laboratory, for letting me participate in small projects that helped me produce article publications and research experience, and without their support this work would not have been possible.

I would like to thank my laboratory colleagues who have helped me greatly over the years, namely Jaime Martins, Mário Saleiro and Roberto Lam for discussing scientific and technical problems, but also for almost all problems in the world.

To all persons, who worked or visited the Vision Laboratory, especially those with whom I have worked with, almost on a daily basis.

A very special thank you to my family and friends for their support, understanding and patience.

Research published in this thesis was supported by the Portuguese Foundation for Science and Technology (FCT) through ISR/LARSyS pluriannual funding (UID/EEA/50009/2013), and by the FCT PhD grant (SFRH/BD/79812/2011).

NOME: Miguel Ângelo Pires Farrajota
FACULDADE: Faculdade de Ciências e Tecnologia
ORIENTADOR: Johannes Martinus Hubertina du Buf
CO-ORIENTADOR: João Miguel Fernandes Rodrigues
DATA: Abril de 2017
TÍTULO DA TESE: Reconhecimento de poses e acções humanas usando redes neuronais

Resumo

Esta tese foca a detecção de pessoas e o reconhecimento de poses usando redes neuronais. O objectivo é detectar poses humanas num ambiente (cena) com múltiplas pessoas e usar essa informação para reconhecer actividade humana. Isto é alcançado ao detectar, em primeiro lugar, pessoas numa cena e, seguidamente, estimar as suas juntas corporais de modo a inferir poses articuladas.

O trabalho desenvolvido nesta tese explorou métodos de redes neuronais e de aprendizagem profunda. A aprendizagem profunda permite que modelos computacionais compostos por múltiplas camadas de processamento aprendam representações de dados com múltiplos níveis de abstração. Estes métodos têm drasticamente melhorado o estado-da-arte em muitos domínios como o reconhecimento de fala e a classificação e o reconhecimento de objectos visuais. A aprendizagem profunda descobre estruturas intrínsecas em conjuntos de dados ao usar algoritmos de propagação inversa (*backpropagation*) para indicar como uma máquina deve alterar os seus parâmetros internos que, por sua vez, são usados para processar a representação em cada camada a partir da representação da camada anterior.

A detecção de pessoas em geral é uma tarefa difícil dado à grande variabilidade de representações devido a diferentes escalas, vistas e oclusões. Uma estrutura de detecção de objectos baseada em características convolucionais de múltiplos estágios para a detecção de pedestres é proposta nesta tese. Esta estrutura estende a estrutura *Fast R-CNN* com a combinação de várias características convolucionais de diferentes estágios da CNN (*Convolutional Neural Network*) usada de modo a melhorar a precisão do detector. Isto proporciona detecções de pessoas com elevada fiabilidade numa cena, que são posteriormente conjuntamente usadas como entrada no modelo de estimação de poses humanas de modo a estimar a localização de articulações humanas para a detecção de múltiplas pessoas numa imagem.

A estimação de poses humanas é obtido através de redes neuronais convolucionais profundas que são compostas por uma série de auto-codificadores residuais que fornecem múltiplas previsões que são, posteriormente, combinadas para fornecer um "mapa de calor" de articulações corporais. Nesta topologia de rede, as características da imagem são processadas ao longo de várias escalas, capturando as várias relações espaciais associadas com o corpo humano. Repetidos processos de baixo-para-cima e de cima-para-baixo com supervisão intermédia para cada auto-codificador são aplicados. Isto resulta em mapas de calor 2D muito precisos de estimações de articulações corporais de pessoas.

Os métodos apresentados nesta tese foram comparados com outros métodos de alto desempenho em bases de dados de detecção de pessoas e de reconhecimento de poses humanas, alcançando muito bons resultados comparando com outros algoritmos do estado-da-arte.

PALAVRAS-CHAVE: Detecção de objectos, detecção de pedestres, características em múltiplas etapas, pose humana, aprendizagem profunda, redes neuronais.

Human pose and action recognition using neural networks

Abstract

This thesis focuses on detection of persons and pose recognition using neural networks. The goal is to detect human body poses in a visual scene with multiple persons and to use this information in order to recognize human activity. This is achieved by first detecting persons in a scene and then by estimating their body joints in order to infer articulated poses.

The work developed in this thesis explored neural networks and deep learning methods. Deep learning allows to employ computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have greatly improved the state-of-the-art in many domains such as speech recognition and visual object detection and classification. Deep learning discovers intricate structure in data by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation provided by the previous one.

Person detection, in general, is a difficult task due to a large variability of representation due to different factors such as scales, views and occlusion. An object detection framework based on multi-stage convolutional features for pedestrian detection is proposed in this thesis. This framework extends the Fast R-CNN framework for the combination of several convolutional features from different stages of a CNN (Convolutional Neural Network) to improve the detector's accuracy. This provides high quality detections of persons in a visual scene, which are then used as input in conjunction with a human pose estimation model in order to estimate human body joint locations of multiple persons in an image.

Human pose estimation is done by a deep convolutional neural network composed of a series of residual auto-encoders. These produce multiple predictions which are later combined to provide a heatmap prediction of human body joints. In this network topology, features are processed across all scales capturing the various spatial relationships associated with the body. Repeated bottom-up and top-down processing with intermediate supervision for each auto-encoder network is applied. This results in very accurate 2D heatmaps of body joint predictions.

The methods presented in this thesis were benchmarked against other top-performing methods on popular datasets for human pedestrian and pose estimation, achieving good results compared with other state-of-the-art algorithms.

KEYWORDS: Object detection, pedestrian detection, multi-stage features, human pose, deep learning, neural networks.

Publications

Some of the thesis contents and figures have appeared previously in the following publications (or are being prepared for submission):

- PUBLISHED IN CONFERENCE PAPERS
  - Farrajota, M., Rodrigues, J. and du Buf, J.M.H. (2009) Multi-scale keypoint annotation - a biological approach. In Proc. 15th Portuguese Conf. on Pattern Recogn. (RECPAD 2009), Aveiro, Portugal, October 23, pp. 3.
  - Farrajota, M., Rodrigues, J.M.F., du Buf, J.M.H. (2011) Optical flow by multi-scale annotated keypoints: A biological approach. In Proc. Int. Conf. on Bio-inspired Systems and Signal Processing (BIOSIGNALS 2011), Rome, Italy, 26-29 January, pp. 307-315.
  - Farrajota, M., Saleiro, S., Terzic, K., Rodrigues, J.M.H, du Buf, J.M.H (2012) Multi-scale cortical keypoints for realtime hand tracking and gesture recognition, In Proc. 1st Int. Workshop on Cognitive Assistive Systems: Closing the Action-Perception Loop (ISBN 978-972-8822-26-2) in conjunction with IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Vilamoura, Portugal, 7-12 Oct., pp. 9-15.
  - Saleiro, M., Farrajota, M., Terzic, K., Rodrigues, J.M.F., du Buf, J.M.H (2013) A biological and realtime framework for hand gestures and head poses. In C. Stephanidis and M. Antona (Eds.) Universal Access in Human-Computer Interaction. Design Methods, Tools, and Interaction Techniques for eInclusion SE-60, vol. 8009, pp. 556-565. Springer-Verlag Berlin Heidelberg. DOI: 10.1007/978-3-642-39188-0_60.
  - Farrajota, M., Rodrigues, J.M.F., du Buf, J.M.H. (2015) Bio-Inspired Pedestrian Detection and Tracking. In Proc. 3rd Int. Conf. on Advances in Bio-Informatics, Bio-Technology and Environmental Engineering, Birminghan, UK, 26-27 May, pp. 28-33. ISBN: 978-1-63248-060-6. DOI: 10.15224/978-1-63248-060-6-07
  - Farrajota, M., Rodrigues, J.M.F., du Buf, J.M.H. (2015) Bio-Inspired Pedestrian Detection and Tracking. Int. International Journal of Business and Management Study (IJBMS), 2(2), 409-414. ISSN : 2372-3955 (invitation from ABBE 2015)
  - Saleiro, M., Farrajota, M., Terzic, K., Krishna, S., Rodrigues, J.M.F., du Buf, J.M.H. (2015) Biologically inspired vision for human-robot interaction. In M. Antona and C. Stephanidis (Eds.): Universal Access in Human-Computer Interaction 2015, Part II, LNCS 9176, pp. 505–517. DOI: 10.1007/978-3-319-20681-3_48
  - Farrajota, M., Rodrigues, J.M.F., du Buf, J.M.H. (2015) Pedestrian Detection using Spatial Pyramid Pooling in Deep Convolutional Networks. In Proc. 21th edition of the Portuguese Conference on Pattern Recognition, Faro, Portugal, 30 Oct., pp. 48-49
  - Farrajota, M., Rodrigues, J.M.F., du Buf, J.M.F. (2016) Deep neural networks video surveillance framework for elderly people monitoring. In M. Antona and C. Stephanidis (Eds.): Universal Access in Human-Computer Interaction 2016, Part II, LNCS 9738, pp. 370–381. DOI: 10.1007/978-3-319-40244-4_36

- Farrajota, M., Rodrigues, J.M.F., du Buf, J.M.H. (2016) Pedestrian Detection Using Multi-Stage Features in Fast R-CNN, In Proc. of the 22nd edition of the Portuguese Conference on Pattern Recognition, Aveiro, Portugal, 28 Oct., pp. 21
- Farrajota, M., Rodrigues, J.M.F., du Buf, J.M.H. (2016) Human Pose Estimation Using Wide Stacked Hourglass Networks, in Proc. of the 22nd edition of the Portuguese Conference on Pattern Recognition, Aveiro, Portugal, 28 Oct., pp. 89
- Farrajota, M., Rodrigues, J.M.F., du Buf, J.M.H. (2017) Using Multi-Stage Features in Fast R-CNN for Pedestrian Detection. In Proc. Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion November 9-11, 2016 - UTAD, Vila Real, Portugal. DOI: `http://dx.doi.org/10.1145/3019943.3020000`

• SUBMITTED

- Farrajota, M., Rodrigues, J. and du Buf, J.M.H. (2017) Human Pose Estimation by a Series of Residual Auto-Encoders. Submitted to the 8th Iberian Conference on Pattern Recognition and Image Analysis, Faro, Portugal. June 20-23

• OTHER RESEARCH PAPERS

- du Buf, J.M.H., Barroso, J., Rodrigues, J.M.F., Paredes, H., Farrajota, M., Fernandes, H., José, J., Teixeira, V., Saleiro, M. (2010) The SmartVision navigation prototype for the blind. In Proc. Int. Conf. on Software Development for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2010), Oxford, UK, 25-26 November, pp. 167-174.
- José, J., Farrajota, M., Rodrigues, J.M.F., du Buf, J.M.H. (2010) A vision system for detecting paths and moving obstacles for the blind. In Proc. Int. Conf. on Software Development for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2010), Oxford, UK, 25-26 November, pp. 175-182.
- José, J., Farrajota, M., Rodrigues, J.M.F., du Buf, J.M.H. (2011) The SmartVision local navigation aid for blind and visually impaired persons. JDCTA: Int. Journal of Digital Content Technology and its Applications, Vol. 5, No. 5, pp. 362 - 375. DOI: `10.4156/jdcta.vol5.issue5.40`
- du Buf, J.M.H., Barroso, J., Rodrigues, J.M.F., Paredes, H., Farrajota, M., Fernandes, H., José, J., Teixiera, V., Saleiro, M. (2011) The SmartVision navigation prototype for blind users. JDCTA: Int. Journal of Digital Content Technology and its Applications, Vol. 5, No. 5, pp. 351 - 361. DOI : `10.4156/jdcta.vol5.issue5.39`
- Farrajota, M., Martins, J.A., Rodrigues, J.M.F. and du Buf, J.M.H. (2011) Disparity energy model with keypoint disparity validation. In Proc. 17th Portuguese Conf. on Pattern Recognition, Porto, Portugal, 28 Oct., pp. 70-71.
- Martins, J.A. Farrajota, M., Lam, R., Rodrigues, J.M.F, Terzic, K., du Buf, J.M.H. (2012) A disparity energy model improved by line, edge and keypoint correspondences. In Proc. European Conf. on Visual Perception, Alghero, Italy, 2-6 Sept., Perception Vol. 41 Suppl., pp. 76.

– Terzic, K., Lobato, D., Saleiro, S., Martins, J., Farrajota, F., Rodrigues, J. M.F., du Buf, J. M. H. (2013). Biological Models for Active Vision: Towards a Unified Architecture. In M. Chen, B. Leibe, and B. Neumann (Eds.), Computer Vision Systems SE - 12, vol. 7963, pp. 113-122. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-39402-7_12

– Farrajota, M., Rodrigues, J.M.F., du Buf, J.M.H. (2013) Keypoint clustering using cortical colour opponency. In Proc. 19th Portuguese Conf. on Pattern Recognition, Lisboa, Portugal, 1 Nov., pp. 2.

– Farrajota, M., Rodrigues, J.M.F., du Buf, J.M.H. (2014) Region segregation by linking keypoints tuned to colour. In Proc. Int. Conf. on Pattern Recognition Applications and Methods, Angers, France, 6-8 Mar., pp. 247-254

Contents

Chapter 1

Introduction

This chapter introduces the scope of this thesis, namely about object recognition, action perception and deep learning.

1.1 Scope of the thesis

When watching a movie, we promptly recognize the scene and the elements in it, like persons, buildings, environments, cars, animals, etc. We can identify the location, a specific actor, the breed of a dog, or the brand of a car. Like many other natural tasks that the human brain performs with apparent ease, visual recognition has turned out to be difficult to reproduce in artificial systems [Poggio and Ullman, 2013]. This task is a very challenging computational problem that will likely play a significant role in eventually making intelligent machines, and it is also a still open key problem in neuroscience.

We perceive with apparent ease the three-dimensional (3D) structure of the world around us, taking into account many phenomena like structure, translucency, patterns of light and shading across surfaces: the visual scene. Perceptual psychologists have spent decades trying to understand how the visual system works, but a complete solution to this puzzle still remains elusive [Lindsey, 2000; Livingstone, 2008]. Researchers in computer vision have been developing mathematical techniques for recovering the shape and appearance of objects in images [Szeliski, 2011]. For example, we now have reliable techniques for accurately computing a partial 3D model of an environment from thousands of partially overlapping photographs [Snavely et al., 2006]. Given a large enough set of views of a particular object

or façade, it is now possible to create accurate and dense 3D surface models using stereo matching [Mei et al., 2011]. We can also track a person who moves against a complex background [Liu et al., 2015a], or attempt to find and name all persons in a photograph by using a combination of face, clothing and hair detection and recognition [Zhang et al., 2015a]. However, despite all of these advances, it still remains an aspiration to have a computer which is able to interpret an image at the same level as a small child, for example to count all animals in a picture.

Understanding why and how vision is such a difficult task is crucial if we want to solve biological and computer vision. But why is vision so difficult? In part, it is because vision is an inverse problem, in which we seek to recover some unknowns given insufficient information to fully specify the solution [Szeliski, 2011]. Therefore, we must resort to physical and probabilistic models to disambiguate between potential solutions. Also, modeling the visual world is presently one of the most difficult tasks. For example, modeling the visual world in all of its rich complexity is far more difficult than modeling the vocal tract. In computer vision, researchers try to describe the seen world in one or more images in ways such that the worlds can be reconstructed by their properties such as shape, illumination and color.

Object recognition remains a hot topic in computer vision research [LeCun et al., 2015; Szeliski, 2011]. This thesis provides a special focus on this particular area where objects (persons in this case) populate cluttered real-world scenes. Regarding other popular areas of research in computer vision, a wide variety of real-world applications are being tackled: optical character recognition (OCR) by reading handwritten postal codes on letters [LeCun et al., 1990] and automatic number plate recognition [Chang et al., 2004]; machine inspection with rapid parts inspection for quality assessment using stereo vision with specialized illumination to measure tolerances of aircraft wings or car parts, or looking for defects in steel castings using X-rays [Jia, 2009]; retail using object recognition for automating checkout lanes [Novak, 1996]; 3D model building (photogrammetry) with fully automated construction of 3D models from aerial photographs [Haala and Kada, 2010] as used in systems such as Bing Maps; medical imaging by registering pre-operative and intra-operative imagery or performing image-guided neurosurgery [Archip et al., 2007; Marreiros, 2016]; car safety by detecting unexpected obstacles such as pedestrians on the road, under conditions where active vision techniques such as radar or lidar do not work well [Miller et al., 2009; Urmson et al., 2009]; motion capture (mocap) by using retro-reflective markers viewed from mul-

tiple cameras or other vision-based techniques to capture actors for computer animation [Andrews et al., 2016]; surveillance by monitoring for intruders [Ahmed et al., 2010] and analyzing highway traffic [Cheng and Hsu, 2011]; fingerprint recognition and biometrics for automatic access authentication as well as forensic applications [Cappelli et al., 2010].

For many surveillance applications [Ahmed et al., 2010; Garcia-Martin and Martinez, 2010], the perception of other people's behavior is of particular importance. The perception of human action has some tradition in practical philosophy [Meggle, 1977] and a comprehensive representation of the various aspects of the perception of human action and body movements is necessary. The enormous number of possibilities requires us to syntactically describe them into easy-to-grasp concepts. For example, observers can draw an ample variety of information from the stream of behavior:

- Simple and complex body movements or actions with or without objects, such as walking, dancing, picking up a cup or tying a tie [Loula et al., 2005];
- Real or pretended internal states, i.e., intentions, motives or emotions that are particularly reflected in expressive behavior such as effort, anxiety or happiness [Dolan, 2002];
- Various verbal and paralinguistic pronouncements [Batliner et al., 2000];
- Symbolic actions such as greeting [Nehaniv et al., 2005]; and
- Social actions such as helping or cooperating [Adolphs, 2003].

We do not perceive human movements as mere changes in the locations of parts of the body when visualizing/classifying tasks like walking, dancing, playing cards or eating. This classification seems to be effortless, considering the complex and temporally extended sensory information involved in such actions. Evidence from psychology and neurology suggests an existence of a conceptual system that represents knowledge about the world and a perceptuomotor system that underlies movement specification [Cook et al., 2014].

This thesis focuses on the perception of instrumental behavior for activity recognition. Instrumental behavior can be subdivided into simple body movements (operations), actions and activities [Prinz, 1996]. Highly automated simple body movements such as walking or grasping are the basis of simple intentional actions like throwing a ball. Perceiving an action requires a relationship between movements, intentions and effects. The perception of

symbolic actions such as signing a contract or more complex activities that include many actions, such as preparing a family reunion, requires a semantic integration of visual features of movement and actions, verbal communications and prior knowledge. This information is important in order to classify human actions in visual scenes.

Concerning machine vision, perceiving humans and their activities in a visual scene usually requires the machine to be able to reason from a set of sequences of body movements and actions and to understand what activities are being performed. We, as human beings, have long dreamed of creating machines that could not only see but also think and reason. With the appearance of machine learning, this recurring dream is turning into reality. Today, machine learning is a thriving field with many practical applications and active research topics [Goodfellow et al., 2016; LeCun et al., 2015; Schmidhuber, 2015]. We develop intelligent software to automate routine labor, understand speech or images, make diagnoses in medicine and support basic scientific research.

The challenge to artificial intelligence is solving tasks that are easy for people to perform but hard to describe in a formal way that we solve intuitively, like recognizing spoken words or faces in images. Allowing computers to learn from experience and to understand the world in terms of a hierarchy of concepts, with each concept being defined in terms of its relation to simpler concepts, proved to be the solution. This approach can be viewed as gathering knowledge from experience, and avoids the need for human operators to formally specify all of the knowledge that the computer needs.

This means that the choice of representation has an enormous effect on the performance of machine learning algorithms. For many tasks, however, it is difficult to know what features should be extracted. Let us take a car as an example. It can be characterized as a composition of wheels, doors, windows, a chassis, seats, etc., so we might like to use the presence of a wheel as a feature. It is difficult to describe exactly what a wheel looks like in terms of pixel values. A wheel has a simple geometric shape, but lighting effects, shadows and occlusions produced by other objects can significantly alter the composition of this particular object. One possible solution to this problem is to use machine learning to discover not only the mapping from representation to output but also the representation itself. This approach is known as representation learning. Learned representations usually result in a much better feature representation and performance than those engineered by hand. Deep learning solves the representational problem by introducing representations that are expressed in terms of

other, simpler representations.

Deep learning allows algorithms to build complex concepts out of simpler ones. Besides learning the right representation from data, another important aspect of deep learning is that depth allows to learn a multi-step computer algorithm. Each layer of the representation can be viewed as one stage of a pipeline with growing complexity. Machine learning is the only viable approach to building AI systems that can operate in complicated, real-world environments. Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones. Broadly speaking, the deep learning field of research can be characterized by a few key trends [LeCun et al., 2015]:

- Deep learning has become more useful as the amount of available labeled training data has increased;
- Deep learning models have grown in size over time as computer hardware and software infrastructure has improved;
- Over time, deep learning has greatly improved performance in increasingly more complex applications.

The deep learning phenomenon is often linked to several biological processes of the brain [Bengio et al., 2015]. Although researchers like Bengio et al. have refered to the human brain as an influence, the purpose of deep learning is not to attempt to simulate the human brain. Modern deep learning draws inspiration from many fields, especially applied mathematics like linear algebra, probability, information theory and numerical optimization. The main reason for the small role of neuroscience in deep learning research today is that there isn't enough information about the brain to use it as a guide. To obtain a deep understanding of the actual processes used by the brain, one would need to be able to monitor the activity of thousands of interconnected neurons simultaneously. Because it is not feasible to do this (yet), we are far from understanding even some of the most simple and well-studied parts of the brain [Olshausen and Field, 2005].

The main focus on this thesis is the perception of human action in a visual scene. The main topics are:

- The computational implementation of a functional model for object detection in the particular case of persons; and
- The development of a model for human pose inference; and
- To classify activities of persons in a visual scene.

In the next section, the structure of the thesis will be presented.

1.2 Overview of the thesis

This thesis is divided into six chapters, each one corresponding to a specific subject. It is important to stress that some sections of the chapters may be slightly repeated because they were based on papers that have already been published or will be published soon.

Chapter 2 presents a small overview of object recognition, human action and deep learning. In the object recognition task, it starts by explaining the main difficulties in perceiving objects from their surroundings with a brief explanation of the visual cortex and the areas involved in perceiving objects. Then, object representation is addressed with respect to human and computer vision. Finally, an overview of classic object recognition methods from the computer vision literature is presented.

In human action perception, first human activity recognition is defined and then related to how the human brain grasps this concept and how computer vision tackles the same problem. This is achieved by first providing a brief overview of cortical areas responsive to human action and body pose, and later by providing a brief overview of classic activity recognition methods from the computer vision literature.

Finally, the concept of deep learning is generally explained, addressing some key aspects namely neural networks, supervised learning, backpropagation and convolutional neural networks. Also, a generic view about image understanding and object recognition methods employing deep learning is addressed.

Chapter 3 presents an initial study about person detection and action recognition in the case of gestures. A biologically inspired vision method for human-robot interaction is described that makes use of head and gesture recognition with biological descriptors. This method integrates head and hand detection models using keypoint templates for matching heads and hands with internal template datasets. This chapter was published by Saleiro

et al. [2013]. This work provided some insights about object detection and action recognition methods using classic methods, as described in Chapter 2, which were then explored in more detail in Chapter 4, with more accurate methods and deep learning.

During the course of the research developed for this thesis, there was a shift in methodology due to insufficient results provided by the hand-engineered features such as cortical keypoints used as the basic framework for this investigation. Although this previous framework gave some satisfactory results for some tasks, for other tasks such as pedestrian detection and body pose estimation, it did not provide sufficient results in comparison with other state-of-the-art approaches. Also, with the emergence of deep learning methods (ConvNets) which offered a better solution with better results and a simpler framework, it was necessary to depart from the previous approach from the classic computer vision literature and explore neural network methods. This proved to be the right course of action, resulting in several improvements to the state-of-the-art methodology, which are described in more detail in the contributions section in Chapter 6.

Chapter 4 presents a method for pedestrian detection using ConvNets. The proposed method is built on the popular Fast R-CNN [Girshick, 2015] framework for object detection applied to pedestrians with some important modifications. These modifications take into account complications of pedestrian detection like scales, view and occlusions, by extracting and combining features from multiple layers of a ConvNet pipeline with different feature map resolutions when classifying region proposals. A study about implementation variations of the Fast R-CNN architecture is provided along with a benchmark of the proposed method on a popular dataset for pedestrian detection with other state-of-the-art methods. This chapter was fully published by Farrajota et al. [2016b].

Chapter 5 describes a human pose detection model using a series of deep convolutional auto-encoders. The model recognizes human body joints by feeding an image of a centered person as input to a series of auto-encoders composed of many convolutional layers, which then generates a 2D heatmap of body joints. A detailed description of the model's architecture is provided, along with a benchmark comparison with other top-performing methods on two popular datasets for human body joint detection. This chapter was submitted to the 8th Iberian Conference on Pattern Recognition and Image Analysis in 2017.

The final **Chapter 6** provides a summary of major achievements, concluding remarks and ideas for future research.

Chapter 2

Overview: Object Recognition, Human Action and Deep Learning

This chapter briefly presents an overview of three major concepts explored in this thesis, namely object recognition, human action and deep learning. First, the task of visual object recognition is addressed, providing insights concerning cortical processes of the human brain involved in visual object categorization and recognition. This is followed by an overview of classic computer vision applications which are inspired by biological processes. Next, human action perception is introduced, how it is perceived by the human brain and how computer vision tackles this task. Finally, a brief introduction to deep learning is presented. It focuses on four key topics, namely neural networks, supervised learning, backpropagation and convolutional networks. Some major fields are addressed where deep learning has significantly progressed the state-of-the-art in computer vision.

2.1 Object recognition

We still do not completely know how our visual system works. However, it has been the subject of many studies [Bar et al., 2006; Hubel, 1995; Serre et al., 2005] from which some theories about its inner workings have emerged. Some of the studies were done with expensive equipment [Rodrigues, 2008] which enabled researchers to measure the activity levels in particular brain regions when scenes or objects were shown to test persons.

We human beings are able to detect and recognize an infinity of objects almost instantly, regardless of variations caused by shape, position, occlusion or illumination [Al-Absi and Abdullah, 2009]. In a house, by peeking into a room, we instantly realize whether it is a living room, a bedroom or an office, because we immediately recognize the kinds of objects in

that room. This enables us to infer about the type of room that we are looking at [Vasudevan and Gächter, 2007] or predict which other objects we may expect in that room, even if we have not yet seen those specific objects.

Object recognition is an ability that many animals, including us humans, possess. With a simple observation of an object, we are able to identify and categorize it irrespective of the countless variations that may occur due to illumination, position, occlusion or orientation (both intra-class and inter-class variations). In that regard, it is a great challenge to develop vision systems that may perform as good as we do. The main difficulties in the development of such methods lie in the variations mentioned above and in the challenge of creating an algorithm which is able to generalize the recognition of an object from a group of sample images.

Regarding the process of recognition, for some time we used to think that our visual system applies a sequence of processes: detection, segregation, categorization and recognition [DiCarlo et al., 2012]. According to recent research [Bar et al., 2006; DiCarlo et al., 2012; Oliva and Torralba, 2006], these processes cannot be completely sequential. They must occur in "parallel" or at least partially. It was common to think that in order to recognize an object, we first had to isolate it from the background. However, recent research suggests that the categorization of objects occurs before segregation [DiCarlo et al., 2012]. In other words, before we realize where the observed object is, the brain already knows which object it is [Rodrigues and du Buf, 2009a]. Apart from all the advances in research, we still do not know the exact order in which visual processes occur, nor in which cases there is parallel processing.

Object recognition is one of the most popular topics being studied in computer vision since it is connected to almost all applications [LeCun et al., 2015; Szeliski, 2011]. Significant research has been done to develop representation methods and algorithms for recognizing objects in images captured under different conditions (points of view, illumination, occlusions, etc.). In some cases of very distinct objects, such as fingerprints [Cappelli et al., 2010], faces [Taigman et al., 2014] and pedestrians [Li et al., 2015], substantial success has been accomplished. For instance, fingerprint recognition systems achieve accuracies over 98% [Wang et al., 2014]. For face recognition, the human-level performance in face verification (97.53%) on the LFW dataset [Learned-Miller et al., 2016] has been surpassed by the method of Lu and Tang [2014] with an accuracy of 98.52%. For pedestrian detection, we are getting close

to human-level performance (5.62% miss rate), the best methods scoring less than 10% miss rates [Zhang et al., 2016].

It is important to make a distinction between detection and recognition of objects. For example, in an industrial environment, the objects to detect are previously well defined and sometimes the position where they will appear is also known, making the recognition task much easier and straightforward [Al Ohali, 2011]. On the other hand, if objects can appear in any position, the task becomes a lot more challenging. If we consider that lots of different objects may appear and that they may appear with different shapes and views and in different and complex backgrounds, it becomes very hard to perform object recognition using the methods from classical computer vision. Another aspect that makes it even harder is if the objects are partially occluded. Many object recognition methods are based on the use of big image data sets, containing multiple views of each object to be detected [Meger et al., 2008]. It must be noted that the efficiency of the algorithms decreases and the processing time increases with the increase of the number of objects in the datasets, since the number of comparisons to be made between the objects from the data set and the captured image grows. Another problem emerges from small variations in shape or view that objects may have, even if they belong to the same class of objects. Taking these factors into account, it is not hard to conclude that we are still far from having an object recognition system that performs close to our own visual system.

2.1.1 Human vision

The human brain is an extremely complex and complicated machine. Estimates suggest that our brain is composed of nearly 10^{12} neuronal cells, with each one receiving and transmitting information to hundreds or even thousands of other neurons, with a total number of interconnections somewhere between 10^{14} and 10^{15} [Hubel, 1995]. This only suggests the raw performance potential of the brain, not providing any concrete information regarding the underlying, extremely optimized architecture, both anatomically and functionally, aspects which are very hard to quantify [Hubel, 1995]. While individual neurons themselves are relatively simple cells, they do not see, reason or remember, but the brain as a whole does.

Visual object recognition typically associates visual inputs - starting with an array of light intensities falling on the retina - with semantic categories, for example “horse,” “bicycle” or “face.” Any theory or computational system that attempts to implement or account for this

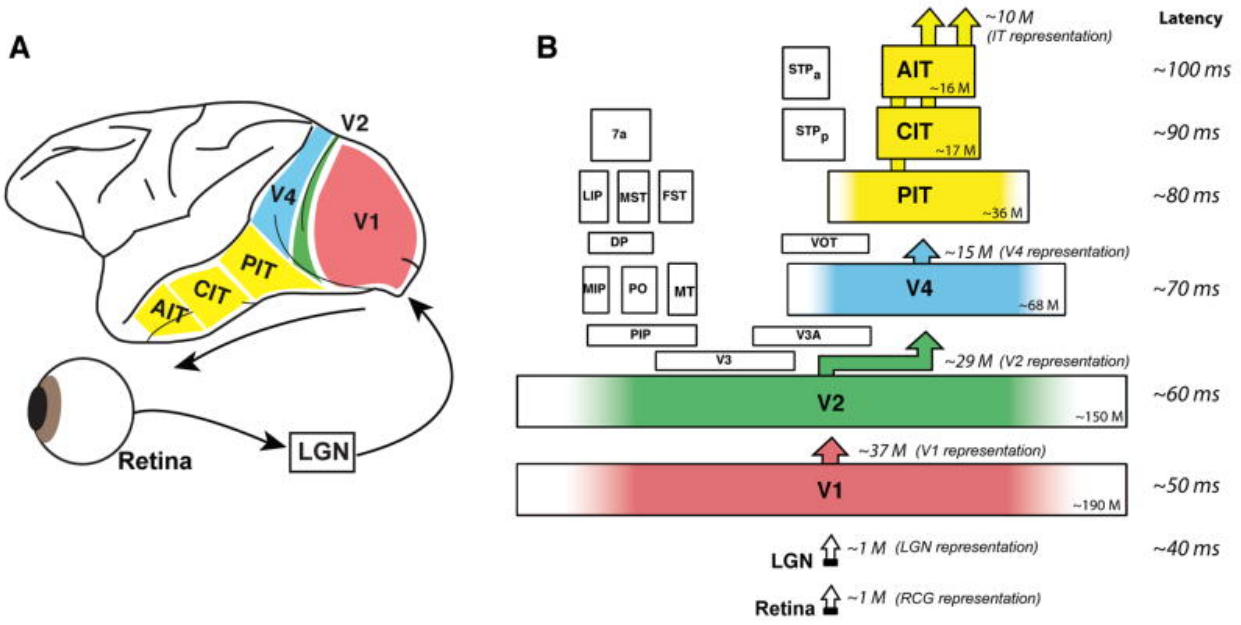

Figure 2.1: Information flow in the visual cortex. Figure from DiCarlo et al. [2012].

process, including the visual recognition system in the ventral occipito-temporal pathway of the human brain [Leeds, 2013], assumes a feedforward processing hierarchy in which the features of representation progressively increase in complexity as one moves up [Riesenhuber and Poggio, 1999]. The final output is a high-level object representation that allows to assign category-level labels. Figure 2.1 shows a feedforward framework representation of the information flow in the visual cortex proposed by DiCarlo et al. [2012]). Within this framework, it is understood that there are several levels of intermediate feature representations which, although less complex than entire objects, capture important object-level visual properties [Ullman et al., 2002]. However, at present there is little empirical data on the neural representations between input image and object representation.

Research in neuroscience has shed some light on some processes and the actual circuitry concerning how our visual system processes information. When we look at something, the information captured in both retinae is propagated in the brain through the Lateral Geniculate Nucleus (LGN) of the Thalamus into the Primary (Striate) Visual Cortex (V1), in the cortical hyper-columns, where most neurons display a property called tuning - they only respond to a specific set of stimuli within their receptive field [Felleman and Van Essen, 1991]. This selectivity means that they can effectively work as feature detectors. For example, in the early visual areas, some neurons are tuned to simple patterns like corners, bars or gratings. However, in higher areas, neurons are tuned to much more complex patterns, e.g.,

in the Inferior Temporal Cortex (IT) a neuron may only fire when a certain face appears in its receptive field.

In neuroscience, the concept of object recognition is difficult to grasp since it involves several levels of understanding, from the information processing or computational level, to the level of circuits, cellular and biophysical mechanisms. After decades of research effort, neuroscientists working on functions in striate and extrastriate cortical areas have produced a huge and still rapidly increasing amount of data, and the emerging scheme of how the cortex performs object recognition is becoming too complex for any simple model [Serre et al., 2005]. Recognition turns out to be a delicate compromise between selectivity and invariance. Therefore, the key issue in object recognition is the specificity-invariance trade-off: the system must be able to finely discriminate between different objects or object classes, at the same time being tolerant to sometimes big object transformations which include scaling, translation, rotation, changes of illumination, viewpoint, context and clutter, non-rigid transformations such as a change of facial expression and, in the case of categorization, also shape variations within a class [Serre et al., 2005].

Another problem that increases difficulty in modeling biological recognition is the definition of the instant when it all starts. Psychologists and psychophysicists, who study how we perceive patterns and images, used to think that, before the processes of object categorization and recognition could begin, the brain must first isolate a figure in an image (for example, a tree or a piece of fruit) from its background. However, recent research suggests that we actually classify objects before we have segregated them, or that both processes occur in parallel. This means that by the time we realize that we are looking at something, our brain already knows what it is [Oliva and Torralba, 2006].

2.1.2 Object representation

The human visual system has an impressive ability to recognize and categorize complex three-dimensional objects [Cutzu and Edelman, 1998]. Recognition performance of human observers is remarkable when accounting for variability in object appearance. The visual system has to pass a long way from the retinal image to the characterization in terms of geometrical structure, familiarity, etc. This variability stems from several sources. First, objects observed under different viewing conditions (for example, varying pose or illumination) generally look different. Second, the appearance of different objects of the same category

may vary significantly, often exceeding the variation between categories of related objects.

The visual system must treat these factors differently: although illumination-related changes in object appearance are mostly ignored (unless the observer makes a special effort to determine the illumination under which a given image has been taken), view-related changes must be both considered (people are usually aware of the orientation of an observed object) and compensated for if the object is to be recognized irrespective of viewpoint. In comparison, shape-related changes must be represented explicitly and acted upon if they are significant. Moreover, it has been shown that the structure of objects is fundamental for recognition [Vasudevan and Gächter, 2007].

The most common features on which classical object recognition algorithms were based are geometry, aspect and interest points. To analyze object geometry, some methods used perspective invariant geometric primitives (lines, circles, etc.). Other algorithms for aspect were based on patterns of the objects: features, textures, histograms, etc. [Shotton et al., 2008]. Finally, the methods based on interest points search for certain regions in images that are invariant to changes caused by illumination or scaling. One of the most used representation methods for vision applications was the Scale-Invariant Feature Transform (SIFT) algorithm [Lowe, 1999].

A conceptual advancement that facilitated recent progress in object recognition was the idea of learning the solution to a specific classification problem from examples, rather than focusing on the classifier design [Poggio and Ullman, 2013]. This was a pronounced departure from the dominant practices at the time. Instead of an expert program with a predetermined set of logical rules, the appropriate representational model was learned and selected from a possibly infinite set of models, based on a set of examples. During learning, a recognition scheme typically extracts a set of measurements (features), and uses them to construct new object representations. This feature representation is then used to classify and recognize objects. Feature selection and object representation are crucial because they facilitate the identification of elements that are shared by objects in the same class and support discrimination between similar objects and categories.

The organization of the visual cortex is hierarchical, with features of increasing complexity represented at successive layers. Models of the visual cortex have naturally adopted hierarchical structures: see Fig. 2.2 for a hierarchical HMAX model by Serre et al. [2007]. In computer vision, the large majority of classical algorithms were non-hierarchical, but recent

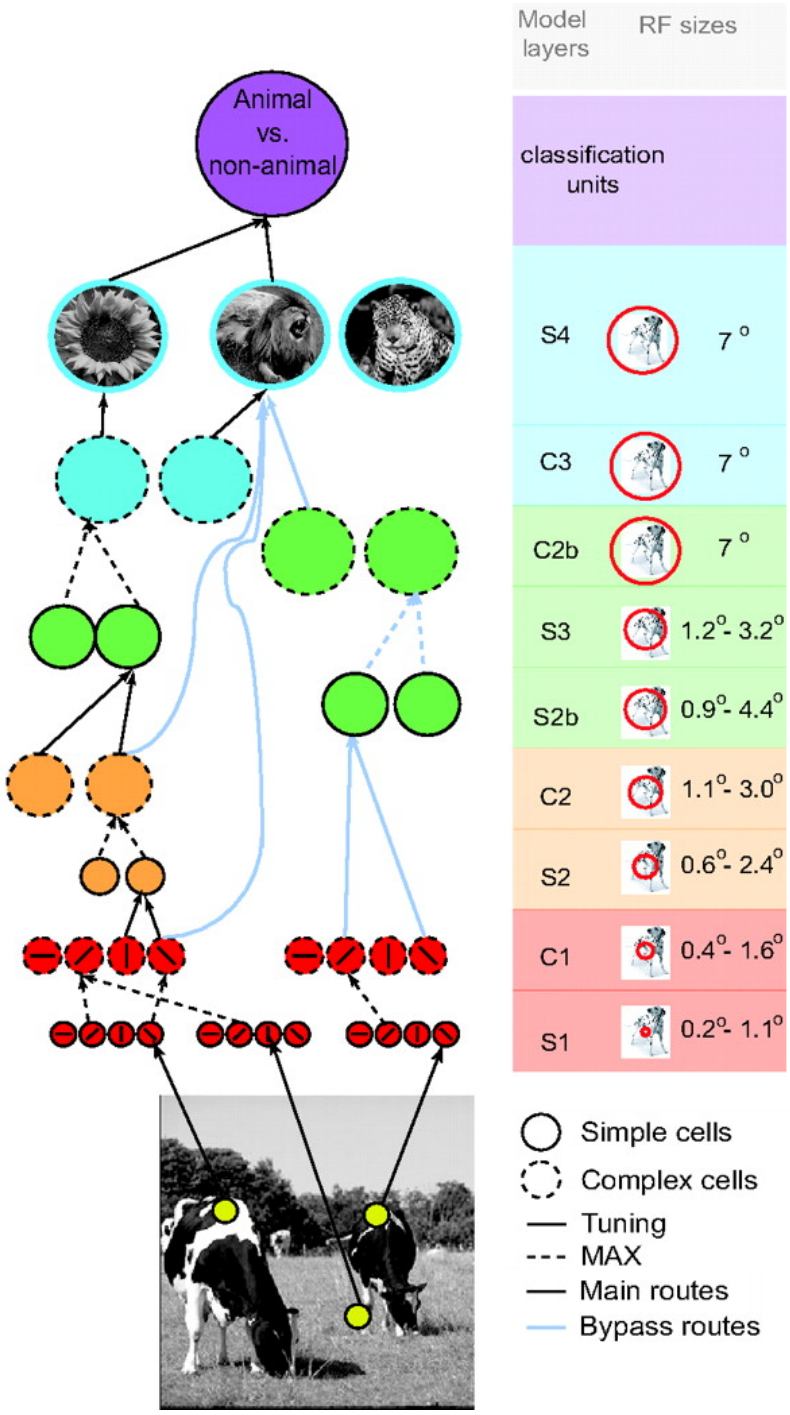

Figure 2.2: The hierarchical HMAX model of Serre et al. [2007]. In the figure, S1 corresponds to a layer of simple cells, and C1 to complex cells in V1. Higher layers correspond to higher cortical areas, with S4 possibly corresponding to IT and S5 to classification circuits in the prefrontal cortex.

learnable representations derive some of their power from a hierarchical organization. One possible role of feature hierarchies is the need to achieve a useful trade-off between selectivity to complex patterns and sufficient tolerance for changes in position and scale, as seen in

the responses of IT neurons [Poggio and Ullman, 2013]. A second advantage of hierarchical representations has to do with efficiency: computational speed and computational resources. For instance, hierarchy may increase the efficiency of dealing with multiple classes in parallel, hence by sharing features at multiple levels. Also, an increase in efficiency may be related to sample complexity. Hierarchical architectures in which each layer is adapted through learning the properties of the visual world may reduce the complexity of the learning task. Finally, hierarchies also offer an advantage in not only obtaining recognition of the object as a whole, but also in recognizing and localizing parts and subparts at multiple levels, such as a face together with its eyes, nose, mouth, etc.

2.1.3 Classic object recognition methods

Over the last 20 years, the literature in computer vision concerning object recognition was rich and diverse: a great variety of approaches. As of today, recent developments in computer vision due to machine-learning algorithms established a clear separation between approaches into two main categories: methods with hand-engineered features and methods with learned features. Here, the former category is designated as classic. The latter, modern approach, which will be introduced in Section 2.3.

Regarding classic object recognition approaches, there are several methods that can also be split into two main types: dedicated methods [Al Ohali, 2011] and general methods [Alahi et al., 2012; Bay et al., 2008; Calonder et al., 2012; Leutenegger et al., 2011; Strecha et al., 2012]. The dedicated methods are developed with the goal of recognizing a limited number of objects and they are optimized to detect only those specific objects. These methods are usually applied in industrial environments to inspect products and to monitor processes [Al Ohali, 2011]. General object recognition methods work in a wider range of applications, despite of having a bigger computational cost [Bay et al., 2008]. Object recognition methods rely on the extraction and recognition of image regularities taken under different illuminations and pose. In other words, most algorithms use certain representations and models to capture those characteristics, making it easier to identify the objects [Lowe, 2004]. The representations may be 2D images or 3D geometric models. The recognition procedure is performed after the extraction of the fundamental features of the image, based on the comparison of the models or object representations with the test image [du Buf et al., 2010].

General object recognition methods based on interest points were successful in object de-

tection and recognition tasks. The SIFT method, proposed by Lowe [2004], was tested and analyzed by Ramisa et al. [2008]. It was compared with the "bag of features" method proposed by Nistér and Stewenius [2006]. From the results obtained, Ramisa et al. verified that, for textured objects with repetitive patterns, the results were similar for both algorithms. Other methods improved the type of descriptors employed by SIFT, namely Speeded-Up Robust Features (SURF) [Bay et al., 2008]. This algorithm was developed with the purpose of being more efficient in terms of processing cost, and therefore it was more suitable for real-time applications. According to the authors, the SURF algorithm was faster, but also more robust and more precise than SIFT.

Another method based on interest points was inspired by biological processes, not only being suitable for object recognition, but also for categorization [Rodrigues and du Buf, 2009a]. This method is based on multi-scale features: lines, edges and keypoints are extracted by using the responses of simple, complex and end-stopped cells in cortical area V1. The keypoints are used to build saliency maps that are then used for Focus-of-Attention (FoA). A similar method [Saleiro et al., 2015] allowed to obtain 2D translation, rotation and scale invariance through the dynamic mapping of saliency maps based on information provided by multi-scale keypoints. The model is split into two parts: keypoints are used for object recognition, and lines and edges for categorization. Apart from this division into two parts, there is also a progression in detail of the flows of data, starting in both cases with a coarse scale (less detail) and progressively using finer scales (more detail).

Besides interest points, other approaches used for general object recognition were based on the sliding window approach [Dalal and Triggs, 2005; Dollár et al., 2009]. These methods employed detection algorithms based on lower-level features like line/edge orientations [Dalal and Triggs, 2005] and color transformations [Dollár et al., 2009], and they were successful in specific tasks in object detection and classification [Felzenszwalb et al., 2013].

2.2 Human action

Human activity recognition plays a significant role in human-to-human interaction and interpersonal relations [Vrigkas et al., 2015]. It provides information about the identity of persons, their personality and psychological state, and, therefore, it is difficult to extract. The ability of humans to recognize another person's activities is a popular subject of study in computer vision. As a result, many applications including video surveillance systems, human-computer

interaction, and robotics for human behavior characterization, require an activity recognition system.

The interest in the topic is motivated by the promise of many applications, both offline and online. For example, automatic annotation of video enables more efficient searching, like finding tackles in football matches, hand shakes in news footage or typical dance moves in music videos. Online processing allows for automatic surveillance, for example in shopping malls or in smart homes for telecare support of the elderly. Interactive applications in human–computer interaction or games also benefit from the advances in automatic human action recognition.

The classification problem of human activity comprises two main subjects: what action has been performed (i.e., the recognition problem) and where did it happen (i.e., the localization problem). When attempting to recognize human activities, it is necessary to determine the kinetic states of a person so that his/her activity can be successfully recognized. Activities such as walking and running are relatively easy to recognize and they occur very frequently in our daily life. On the other hand, more complex activities such as peeling an orange are more difficult to identify. Complex activities may be decomposed into other, simpler ones, which are generally easier to recognize. Usually, existing objects in a scene may help to better understand human activities as they may provide useful information (context) about the ongoing event [Gupta and Davis, 2007].

It is common in human activity recognition's literature to assume a figure-centric scene of uncluttered background where a person is free to perform an activity [Vrigkas et al., 2015]. The development of a fully automated human activity recognition system capable of accurately classifying a person's activities is a challenging task due to several problems like complex backgrounds, occlusions, variations in scale, viewpoint, illumination and appearance [Vrigkas et al., 2015]. Moreover, intra- and interclass similarities make the problem challenging: actions within the same class may be expressed by different people in various ways with different body movements, and actions between different classes of actions may be difficult to distinguish as they may be represented by similar information. This is due to the way that humans perform an activity which depends on their habits, making the problem of identifying the underlying activity difficult to determine.

The main goal of human activity recognition is to perceive activities from video sequences or still images by correctly classifying input data into its underlying activity category. De-

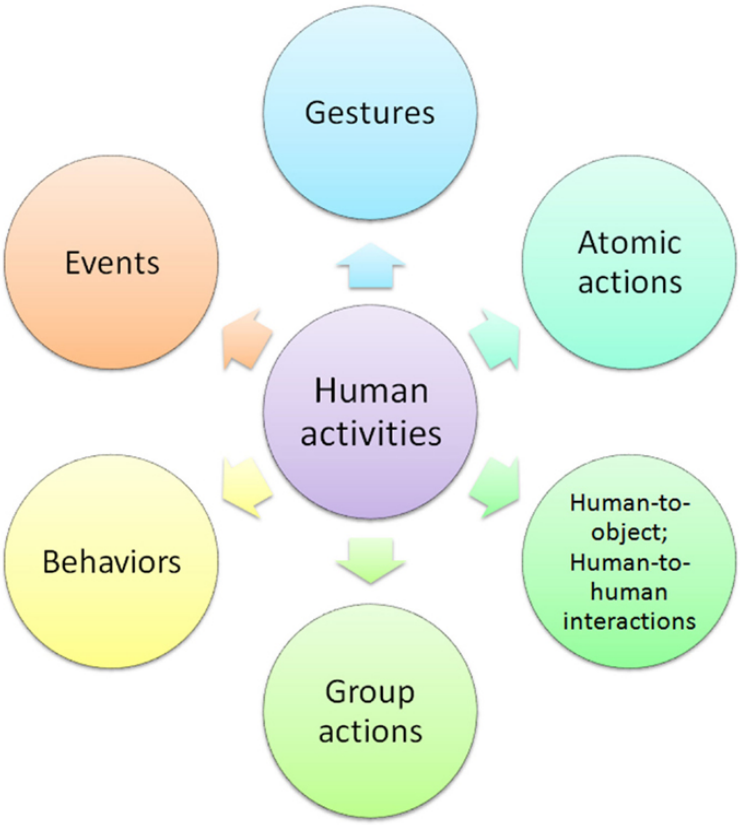

Figure 2.3: Decomposition of human activities. Figure from Vrigkas et al. [2015].

pending on their complexity, human activities can be categorized into (see Fig. 2.3) gestures, atomic actions, human-to-object or human-to-human interactions, group actions, behaviors, and events; for more information see Vrigkas et al. [2015]. Gestures are considered as basic movements of the body parts of a person that may correspond to a particular action [Yang et al., 2013]. Atomic actions are movements of a person describing a particular motion that may be part of more complex activities [Ni et al., 2015]. Human-to-object or human-to-human interactions are activities that involve two or more persons or objects [Patron-Perez et al., 2012]. Group actions are activities performed by a group of persons [Tran et al., 2014]. Human behaviors refer to physical actions that are associated with emotions, personality and the psychological state of an individual [Martinez et al., 2014]. Finally, events are high-level activities that describe social actions between individuals and indicate the intention or the social role of a person [Tian Lan et al., 2012].

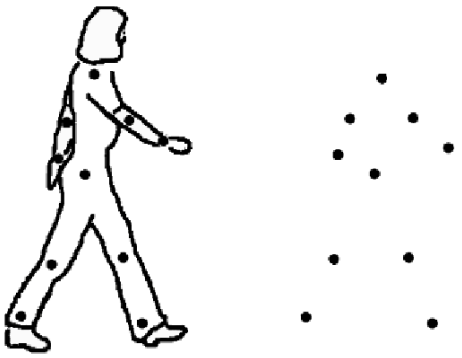

Figure 2.4: Representation of biologic motion by light points positioned at the joints of the person. Figure adapted from Swettenham and Campbell [2005].

2.2.1 Human action and body pose perception by the human brain

Humans are exceptionally adept at recognizing the actions performed by others [Grossman et al., 2000], even when the kinematic patterns of their movements are portrayed by only a handful of light points attached to the head and other body joints [Johansson, 1973] (Fig. 2.4). Static frames with such light points appear to us as meaningless clusters of dots, but when the frames are animated one immediately perceives a biological organism engaged in an easily identifiable activity. With about 12 points of light portraying biological motion, people can reliably discriminate male from female actors, friends from strangers [Grossman et al., 2000], and even identify subtle differences in complex activities like serving a tennis ball [Pollick et al., 1999]. Perception of biological motion is also robust to variations in the number of dots used and variations in exposure duration [Neri et al., 1998].

Furthermore, motion information is important for more than just identification of biological motion [Grossman et al., 2000]. Humans are experts at detecting weak coherent motion amongst a background of incoherent motion [Raymond, 1994], they are very accurate at judging the direction in which objects are moving [Gros et al., 1998], and very sensitive to slight differences in the speed at which objects are moving [Chen et al., 1998]. In addition, motion provides a strong source of information for specifying the 3D shapes of objects [Tittle and Perotti, 1997].

In neuroscience, research has depicted a shared mechanism that underlies both the control of our own bodily movements as well as the recognition of the movements of other individuals [Friston et al., 2011; Ondobaka and Bekkering, 2013; Rizzolatti and Sinigaglia, 2010]. The discovery of mirror neurons (MNs) has boosted the popularity of perceptuo-motor accounts

that explain action recognition as a direct mapping of the perceived consequences of others' movements to the perceiver's movement representations [Rizzolatti and Sinigaglia, 2010]. Subsequent neuroimaging investigations have repeatedly associated activity in the human parieto-frontal mirror neuron system (MNS) with processing of own and others' observed movements [Van Overwalle, 2009].

Mirror neurons were discovered in the 1990s [di Pellegrino et al., 1992; Gallese et al., 1996]. The striking feature of many MNs is that they not only fire when a monkey is performing an action, like grasping an object using a power grip, but also when the monkey passively observes a similar action performed by another monkey. Neurons with this capacity to match observed and executed actions were originally found in area F5 of the ventral premotor cortex (PMC) [di Pellegrino et al., 1992; Gallese et al., 1996] and the inferior parietal lobule (IPL) [Bonini et al., 2010; Fogassi, 2005] of the monkey brain. Today, there is substantial evidence suggesting that MNs are also present in the human brain [Molenberghs et al., 2012].

Since their discovery, MNs have received a great deal of attention from researchers [Cook et al., 2014], and they have been credited with a wide variety of functions like action understanding [Gallese and Sinigaglia, 2011], imitation [Iacoboni and Woods, 1999] and language processing [Rizzolatti and Arbib, 1998]. Moreover, these special neurons have also been implicated in a variety of other fields: embodied simulation [Aziz-Zadeh et al., 2006], empathy [Avenanti et al., 2005], emotion recognition [Enticott et al., 2008], intention-reading [Iacoboni et al., 2005], language acquisition [Théoret and Pascual-Leone, 2002], language evolution [Arbib, 2005], manual communication [Rizzolatti et al., 1996], sign-language processing [Corina and Knapp, 2006], speech perception [Glenberg et al., 2008], speech production [Krühn and Brass, 2008], music processing [Gridley and Hoff, 2006], and aesthetic experience [Cinzia and Vittorio, 2009].

Yet, a large controversy still exists over the neural instantiation of action processing in the perceivers' brains [Csibra, 1993; Hickok, 2009]. Many multi-tiered accounts of action control have gained interest [Kilner et al., 2007], but still an exact description of the functional relationships and dependencies between different tiers, as well as their neural bases, remain unclear [Uithol et al., 2012]. Functional magnetic resonance imaging (fMRI) has identified regions of the pontine micturition center, for example both classic Brodman's Area (BA) 6 and 44 and inferior parietal areas, which are active during both action observation and execution [Vogt et al., 2007]. Overlapping responses to action observation and execution have

been found in single-subject analyses of data [Gazzola and Keysers, 2009]. Most recently, repetition suppression protocols have been used to provide evidence of mirror populations encoding visual and motor representations of the same action [Cook et al., 2014].

2.2.1.1 Selectivity to body pose

One of the most fundamental questions about visual object recognition concerning the human brain is whether objects of all kinds are processed by the same neural mechanisms, or whether some object classes are handled by distinct processing modules [Downing et al., 2001]. The strongest evidence to date for a modular recognition system concerns the case of faces [Kanwisher, 2000]. In contrast, very few studies have considered the mechanisms involved in perceiving the rest of the human body [Downing et al., 2001]. Neuro-psychological reports suggest that semantic information of human body parts may be distinct from knowledge of other object categories [Shelton et al., 1998]. Also, functional neuroimaging studies have indicated regions of the superior temporal sulcus (STS) in the perception of biological motion [Grossman et al., 2000] and have associated regions of left parietal and prefrontal cortices with knowledge about body parts [Le Clec'H et al., 2000].

Visual information about body posture in the human brain is represented in the fusiform gyrus area [Peelen and Downing, 2005], the occipital face area [Michels et al., 2005] and the extrastriate body area [Downing et al., 2001]. Some electro-physiological studies in macaque monkeys found single neurons in the lower bank of the superior temporal sulcus (STS) and the inferior temporal cortex that respond to static images of body postures [Vangeneugden et al., 2009, 2011]. In the upper bank of the STS, neurons were found to fire to body motion [Vangeneugden et al., 2011], which is consistent with the selectivity of the STS to biological motion [Saygin, 2007]. Furthermore, single-unit recording studies in monkeys have identified neurons in the STS that respond selectively to the appearance of the body, including the face [Jellema et al., 2000]. Many psychophysical experiments over the years have shown that local motion information is not necessary to perceive the movement of the walker [Theusner et al., 2011]. Regarding biological motion stimuli, perception can be performed by analyzing first body posture and then body motion [Giese and Poggio, 2003]. This approach, which is popular in computer vision applications [Weinland et al., 2011], uses templates of the human figure to obtain articulated movement.

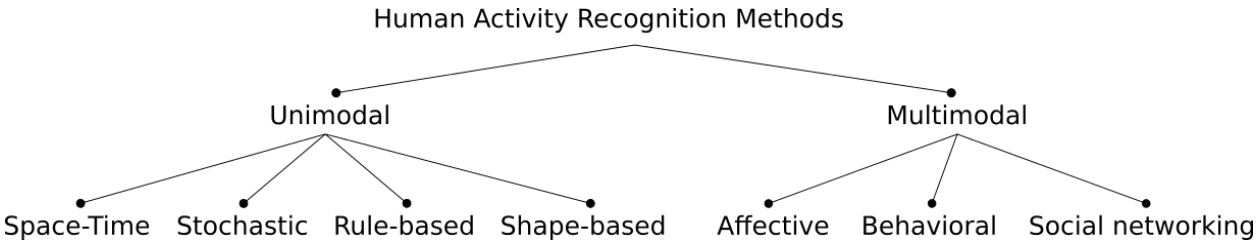

Figure 2.5: Hierarchical categorization of human activity recognition methods, proposed by Vrigkas et al. [2015].

2.2.2 Human activity categorization strategies

The human activity categorization problem has remained a challenging task in computer vision [Poppe, 2010] and previous work on classifying behavior have shown great potential in this area [Vrigkas et al., 2015]. Activity recognition methods can be divided into two main categories: unimodal and multimodal, according to the nature of sensor data that they employ [Vrigkas et al., 2015]. Furthermore, each of these two categories can be split into sub-categories depending on how they model human activities (Fig. 2.5).

Unimodal methods represent human activities from data of a single modality, such as images, and they are further categorized as space-time, stochastic, rule-based and shape-based methods. Space-time methods represent activities as sets of spatiotemporal features [Ruonan Li and Zickler, 2012] or trajectories [Vrigkas et al., 2013]. Stochastic methods apply statistical models, for example hidden Markov models, to represent actions [Iosifidis et al., 2012]. Rule-based methods apply a set of rules to describe human activities [Chao-Yeh Chen and Grauman, 2012]. Shape-based methods represent activities with high-level reasoning by modeling the motion of body parts [Tran et al., 2012].

Multimodal methods combine features gathered from different sources [Wu et al., 2013] and they are classified into three categories: affective, behavioral, and social networking methods. Affective method represent activities according to emotional communications and the affective state of a person [Martinez et al., 2014]. Behavioral methods aim to recognize behavioral attributes and non-verbal multimodal cues such as gestures, facial expressions and auditory cues [Vrigkas et al., 2014]. Social networking methods model the characteristics and the behavior of humans in multiple layers of human-to-human interactions in social events from gestures, speech and body motion [Marín-Jiménez et al., 2014].

2.3 Deep learning

Today, machine-learning powers many aspects of modern society, ranging from web searches [McCallum et al., 2000] to content filtering on social networks [Vanetti et al., 2011] to recommendations on e-commerce websites [Wei et al., 2007]. Also, it is increasingly present in consumer products like cameras and smartphones [Lane et al., 2010]. Machine-learning systems are used to identify objects in images [Krizhevsky et al., 2012], to transcribe speech into text [Hannun et al., 2014], to match news items [Radinsky et al., 2012], posts or products with users' interests [Liu et al., 2013], and to select relevant results of searches [Huang et al., 2013].

For decades, machine-learning techniques were limited in their ability to process data in its raw form [Goodfellow et al., 2016; Schmidhuber, 2015]. These machine-learning systems required resourceful engineering to design a feature extractor that transformed the raw data into a suitable internal representation or feature vector from which the learning system could detect or classify patterns in the input. Nowadays, with the mainstream of representational learning, machine-learning systems can be fed with raw data and automatically discover the representations needed for detection or classification. Deep learning methods are representation learning methods with multiple levels obtained by composing simple modules which each transform the representation at one level (starting with the raw inputs) into a representation at a higher, slightly more abstract level. With the composition of enough such transformations, very complex functions can be learned [LeCun et al., 2015]. In tasks like classification, higher representation layers augment aspects of the input that are important for discrimination and suppress irrelevant ones. For example, from an input image in the form of an array of pixels, the first representation layer learns features that typically represent the presence or absence of edges at particular orientations and locations in the image (see Fig. 2.6). Then, following layers typically detect increasingly more complex patterns by combining features like edges or blobs to form patterns that correspond to parts of objects, and subsequent layers would detect objects as combinations of these parts. The key aspect of deep learning is that these layers of features are not designed by human engineers, but instead learned from data.

Recently, deep learning has made major advances in the computer science field [Bordes et al., 2014; Ciodaro et al., 2012; Collobert et al., 2011b; Girshick, 2015; Glorot et al., 2011; Graves and Jaitly, 2014; Helmstaedter et al., 2013; Kim, 2014; Krizhevsky et al., 2012;

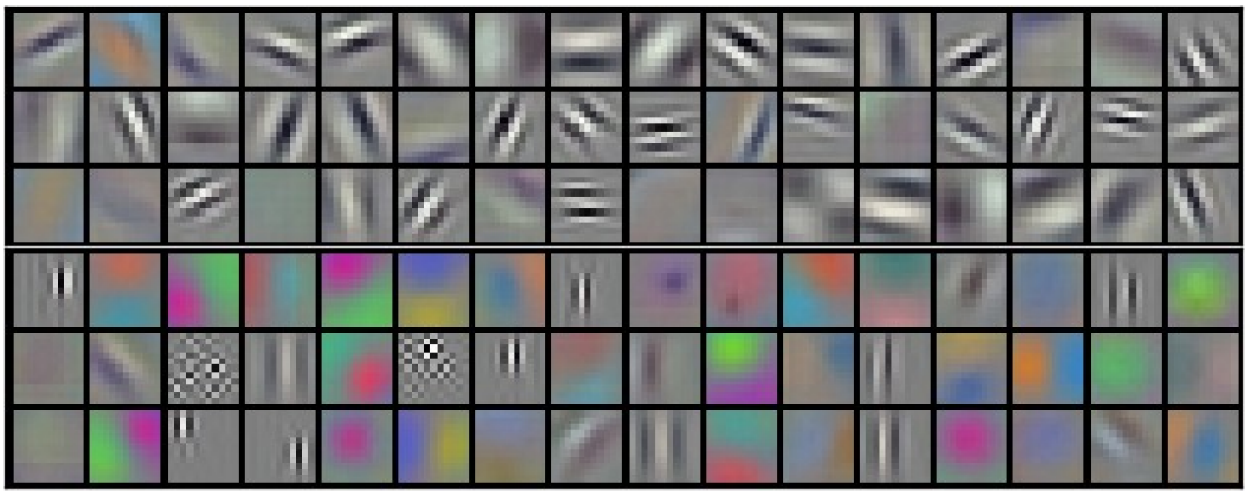

Figure 2.6: Feature representations learned from data using deep learning. These feature maps where obtained by training a deep Convolutional Neural Network (ConvNet) in the ImageNet [Russakovsky et al., 2015] dataset, which contains over 1 million images. Figure from Krizhevsky et al. [2012].

Ma et al., 2015; Xiong et al., 2015]. Its power comes from its ability to be very good at discovering intricate structures in high-dimensional data. Therefore it is applicable to many domains like science, business and government. In addition, deep learning methods achieved state-of-the-art results in many applications like image recognition [Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; Szegedy et al., 2014a], object detection [Girshick, 2015; Liu et al., 2015b; Ren et al., 2016] and speech recognition [Graves and Jaitly, 2014; Hinton et al., 2012; Mikolov et al., 2011]. Moreover, it has surpassed other machine-learning techniques in predicting the activity of potential drug molecules [Ma et al., 2015], analyzing particle accelerator data [Ciodaro et al., 2012], reconstructing brain circuits [Helmstaedter et al., 2013], and predicting the effects of mutations in non-coding DNA on gene expression and disease [Xiong et al., 2015]. Deep learning has also produced top results in various tasks like natural language understanding [Collobert et al., 2011b], particularly topic classification [Kim, 2014], sentiment analysis [Glorot et al., 2011], question answering [Bordes et al., 2014] and language translation [Sutskever et al., 2014].

In the next sections, key aspects relevant to the rise of deep learning, namely neural networks, supervised learning, backpropagation and convolutional neural networks will be briefly described.

2.3.1 Neural networks

A standard neural network (NN) consists of many simple connected processors called neurons, each producing a sequence of real-valued activations (Fig. 2.7-A). This network works in a simple way: input neurons get input data (for example, sensors perceiving the environment) and other neurons get activated through weighted connections from previously active neurons (Fig. 2.7-B). Some neurons may influence the environment by triggering actions. Learning is about finding weights that make the NN exhibit the desired behavior, like identifying a face in an image. Depending on the problem and how the neurons are connected, such behavior may require long causal chains of computational stages, where each stage transforms (often in a non-linear way) the cumulative activation of the network. Deep Learning concerns with accurately assigning credit [Minsky, 1961] across many such stages [LeCun et al., 2015]. Credit assignment is about finding internal parameters that make the networks exhibit a desired behavior like driving a car.

There are many kinds of NN methods in many applications in computer science, but most fall into two main categories: shallow NNs [McDonnell et al., 2015] and deep NNs [Krizhevsky et al., 2012]. Shallow NN models with few layers have been around for many decades, dating back to the 1960s and 1970s [Schmidhuber, 2015]. Deep NN models have many more layers than shallow NNs. These networks were only possible due to the development of an efficient gradient descent method for teacher-based supervised learning in discrete networks of arbitrary depth called backpropagation. This optimization scheme was developed in the 1960s and 1970s and later applied to NNs [Schmidhuber, 2015]. Initially, backpropagation-based training of deep NNs was difficult in practice, but by the 1990s and 2000s there were many improvements which turned NN-based methods feasible in many applications [Schmidhuber, 2015]. By this time, deep NNs have attracted wide-spread attention, mainly by outperforming alternative machine learning methods such as kernel machines [Schölkopf et al., 1999; Vapnik, 1995] in numerous important applications. Also, since 2009, supervised deep NNs have won many official international pattern recognition competitions [Schmidhuber, 2015], even outperforming humans in visual pattern recognition tasks in several domains.

2.3.2 Supervised learning

The most common form of machine learning is supervised learning [Goodfellow et al., 2016; LeCun et al., 2015; Schmidhuber, 2015]. This form of learning is characterized by using a

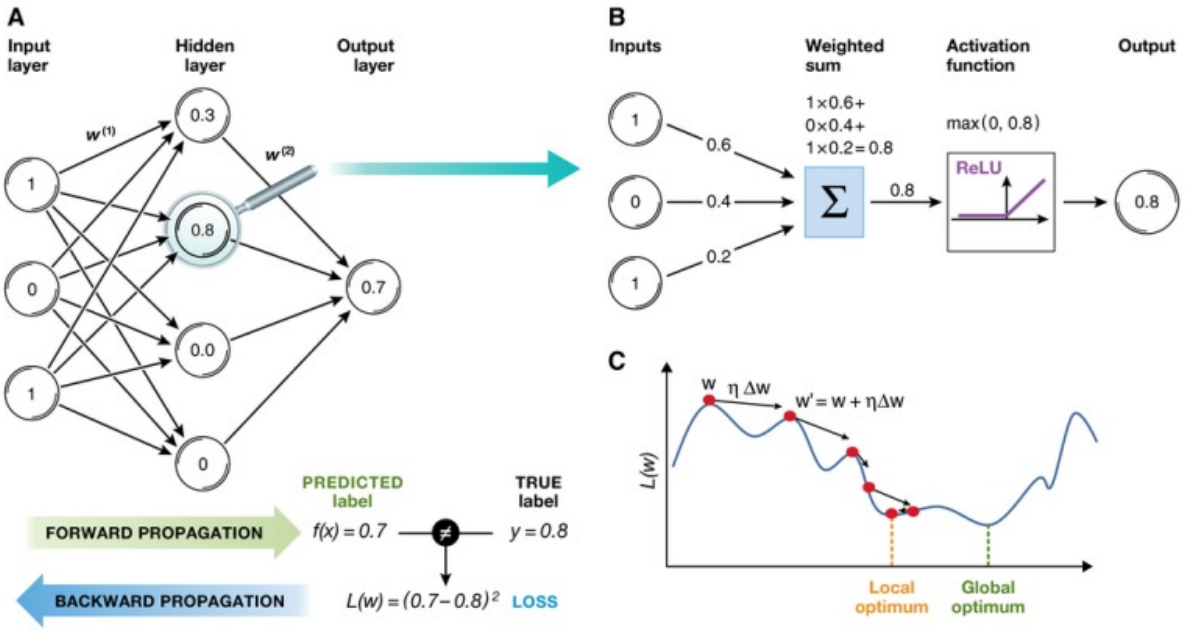

Figure 2.7: Representation of an artificial neural network, adapted from Angermueller et al. [2016]. Left (A): in its basic configuration, the network receives data in an input layer, which is then transformed in a nonlinear way through one or multiple hidden layers, before the final output is computed in the last layer. Top-right (B): neurons are connected to all neurons of the previous layer, where each neuron computes a weighted sum of its inputs and applies a nonlinear transformation before its output. Bottom (C): neurons learn to model data by minimizing a loss function that measures the fit of the model's output to the true label of a sample by means of feedforward and by gradient backpropagation. Since the function to learn is generally high-dimensional and non-convex, this minimization problem resembles a landscape with many hills and valleys.

large collection of a data, for example images of houses, cars, people, pets, words, etc., each one labeled with its category. Then, during training, the machine is shown input data and produces an output in the form of a vector of scores, one for each category in the case of a classification task. The idea is that the true category should have the highest score amongst all categories. This is done by computing an objective function that measures the error (or vector distance) between the output scores and the desired pattern of scores. The machine then modifies and adjusts its internal parameters (weights) to reduce this error. In a typical deep learning system, there may be hundreds of millions of these adjustable parameters, and these require hundreds of millions of labeled examples for the training process.

The learning algorithm adjusts the parameters by computing a gradient vector for each weight. This gradient indicates the amount that the error would increase or decrease if the weights were increased by a marginal amount. The weight vector of the internal parameters

is then adjusted in the direction opposite to the gradient vector. The objective function can be seen as a kind of a hilly landscape in the high-dimensional space of weight values. The gradient vector indicates the direction of the steepest descent in the landscape, and valleys indicate local minima where the error is lower on average (Fig. 2.7-C).

In deep learning, it is common practice to use, for the optimization of the objective function, a procedure called Stochastic Gradient Descent (SGD), or a derived method [Duchi et al., 2011; Kingma and Ba, 2014; Zeiler, 2012]. This consists of showing the input nodes a few data examples, computing the outputs and the errors, computing the average gradient for those examples and then adjusting the weights accordingly. This process is repeated with many small sets (batches) of examples from the training set until the average of the objective function plateaus and stops decreasing. The term stochastic is due to the fact that each small set of examples gives a noisy estimate of the average gradient over all examples. Usually, this simple procedure quickly finds a good set of parameters when compared to other optimization techniques [Bottou and Bousquet, 2007].

Common deep learning architectures take advantage of having lots of labeled data available for training, coupled to powerful optimization techniques. Generally, it is a multilayer stack of simple modules, all or most of which are subject to learning, and many of which compute non-linear input-output mappings. Each module in the stack transforms its input to increase both the selectivity and the invariance of the representation. With multiple non-linear layers, a system can implement very complex functions of its inputs that are simultaneously sensitive to small details (like key differences between races of dogs), and insensitive to large, irrelevant variations such as the background, pose, lighting and surrounding objects.

2.3.3 Backpropagation

Since a long time ago, the aim of some researchers has been to replace hand-engineered features with trainable multilayer networks, but the solution was not widely understood until the mid 1980s [Schmidhuber, 2015]. It turned out that multi-layer models can be trained by the simple stochastic gradient descent method. As long as the modules generate relatively smooth functions of their inputs and of their internal weights, one can compute gradients using the backpropagation procedure. The idea that this could be done goes even back to the early 1960s [Bryson, 1961; Kelley, 1960].

The backpropagation procedure to compute the gradient of an objective function with respect to the weights of a multi-layer stack of modules is basically a practical application of the chain rule for derivatives [Dreyfus, 1962; Linnainmaa, 1970; Werbos, 1982]. The key insight is that the derivative or gradient of an objective function, with respect to the input of the module, can be computed by working backwards from the gradient with respect to the output of that module. The backpropagation equation can be applied repeatedly to propagate gradients through all modules, starting from the output at the top where the prediction of the network is produced, all the way down to the bottom where the input data is fed. Once these gradients have been computed, it is straightforward to compute the gradients with respect to the weights of each module.

Many applications of deep learning use feedforward neural network architectures with backpropagation [Krizhevsky et al., 2012; Liu et al., 2015b; Simonyan and Zisserman, 2015], which learn to map a fixed-size input (for example, an image) to a fixed-size output (for example, the probabilities of several categories). One particular type of deep, feedforward network that was much easier to train and that generalized much better than networks with full connectivity between adjacent layers was the Convolutional Neural Network (CNN or ConvNet) [LeCun et al., 1990]. It achieved many practical successes when neural networks were still overlooked and it has recently been widely adopted by the computer-vision community [LeCun et al., 2015].

2.3.4 Convolutional neural networks

Neural networks and backpropagation became very popular after Krizhevsky et al. [2012] published results on the ImageNet classification challenge [Russakovsky et al., 2015]. This was due to the significant leap in performance relative to the previous methods in that competition. Krizhevsky et al. [2012] achieved significantly better state-of-the-art results compared to previous top results by using a deep, feed-forward ConvNet.

ConvNets are designed to process data that come in the form of multiple arrays (for example, a color image composed of three 2D arrays containing pixel intensities in the three color channels). These networks can be applied to a variety of data modalities with different formats: 1D arrays for signals and sequences, including language; 2D arrays for images or audio spectrograms; and 3D arrays for video or volumetric images. There are four key concepts behind ConvNets [LeCun et al., 2015; Schmidhuber, 2015] that take advantage of

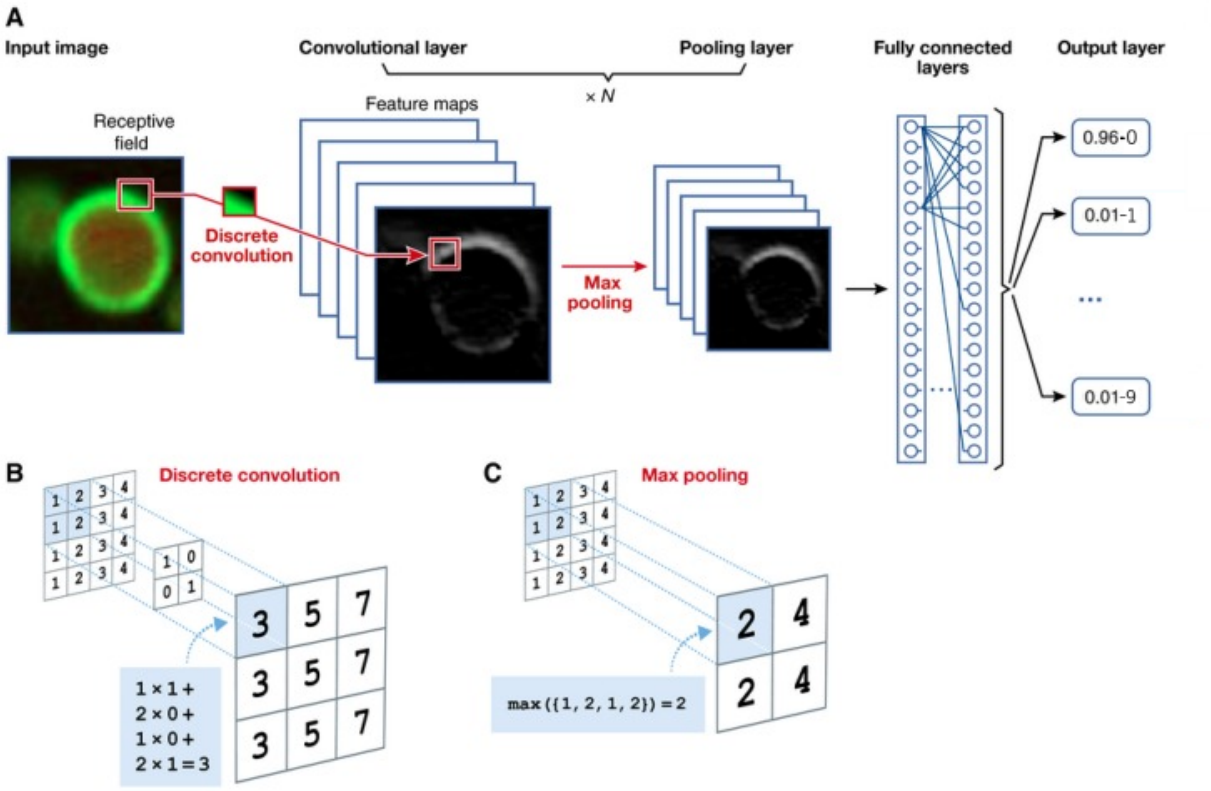

Figure 2.8: Core operations of a simple convolutional neural network composed by one convolutional layer, adapted from Angermueller et al. [2016]. Top (A): a convolutional layer applies multiple filter kernels called feature maps over the previous layer features (in this case over the input image) which are then fed to a sequence of fully-connected layers for classification. In this first layer, different feature maps might, for example, detect edges of different orientations in an image. Bottom-left (B): the activity of a "neuron" of a feature map in the convolutional layer is obtained by computing a discrete convolution of its receptive field, i.e., computing the weighted sum of input neurons, and applying an activation function. Bottom-right (C): The exact position and frequency of features are of little importance for the final prediction. Consequently, the pooling layer summarizes adjacent neurons by computing the maximum over their activity, resulting in a smoother representation of feature activities and robustness to variations in position.

the properties of natural signals: local connections, shared weights, pooling and the use of many layers.

A typical ConvNet is structured as a series of stages (Fig. 2.8-A). The first few stages are composed of two types of layers: convolutional layers and pooling layers. The convolutional layers are composed of feature maps: the units are connected to local patches in the feature maps of the previous layer through a set of weights. This connection is simply a discrete convolution over the previous layer's feature maps (Fig. 2.8-B), hence the name. The result of this local weighted sum is then passed through a non-linearity, typically a Rectified Linear

Unit (ReLU). All units in a feature map share the same filter bank, and different feature maps in a layer use different filter banks. The reason for this architecture is due to two properties. First, in data arrays like images, local groups of values are often highly correlated, forming distinct local patterns that can be easily detected. Second, the local statistics of images and other signals are invariant to location, meaning that a pattern can appear anywhere in an image. Therefore, it is good practice to have units at different locations which share the same weights for detecting the same pattern in different parts of the array.

Convolutional layers are used to detect local combinations of features in the previous layer. On the other hand, the role of the pooling layer is to merge semantically similar features. Since the relative positions of the features forming a pattern can vary slightly, detecting the pattern reliably can be done by obtaining the coarse position of each feature. A typical pooling unit computes the maximum of a local patch of units in some feature maps (Fig. 2.8-C). Neighboring pooling units take input from patches that are shifted by more than one row or column, thereby reducing the dimension of the representation and creating an invariance to small shifts and distortions.

These types of networks are usually composed of several stages of convolution, non-linearity and pooling operations, followed by fully-connected layers [Krizhevsky et al., 2012; Sermanet et al., 2013b; Szegedy et al., 2014a]. Backpropagating gradients through a ConvNet is the same as in a regular deep network, allowing all the weights in all the filter banks to be trained.

The convolutional and pooling layers in ConvNets were inspired by the concept of simple cells and complex cells in the visual cortex [Hubel and Wiesel, 1962], and the overall architecture is reminiscent of the LGN–V1–V2–V4–IT hierarchy in the ventral pathway of the visual cortex [Felleman and Van Essen, 1991]. The modern ConvNet architecture has its roots in the neocognitron [Fukushima and Miyake, 1982], which had a similar architecture but did not have an end-to-end supervised learning algorithm such as backpropagation.

ConvNets have been applied to various applications with great success [LeCun et al., 2015], including detection, segmentation and recognition of objects and regions in images, which usually have plenty of labeled data. These applications cover traffic sign recognition [Ciresan et al., 2012], the detection [Li et al., 2016] and recognition of faces [Taigman et al., 2014], text [Zhang and LeCun, 2015], pedestrians [Sermanet et al., 2013b] and human bodies [Newell et al., 2016] in natural images. Recent ConvNet architectures have hundreds of layers

of ReLUs, hundreds of millions of weights, and billions of connections [He et al., 2015]. Also, with recent progress in hardware, software and algorithm parallelization, training such networks can nowadays be done in a matter of hours, compared to weeks a few years ago.

2.3.4.1 Object recognition methods

In computer vision, object detection is a fundamental and heavily-researched problem. Until recently, the sliding window paradigm was dominant, especially for detecting faces [Viola and Jones, 2004] and pedestrians [Dollár et al., 2014]. Deformable part models [Felzenszwalb et al., 2009] followed this framework, but they allowed for more object variability and they could be used for general object categories. Sermanet et al. [2013b] demonstrated the use of ConvNets for general object detection in a sliding window fashion. More recent detectors follow the region-proposal paradigm established by Girshick [2015], in which a ConvNet is used to classify regions generated by an object-proposal algorithm [Hosang et al., 2015a]. This led to a general framework that many recent detectors apply [Gidaris and Komodakis, 2015; He et al., 2014; Ren et al., 2016; Szegedy et al., 2014b; Zagoruyko et al., 2016].

The feature extraction network used in ConvNets, combined with a classification network on top, forms an integral part of the detection pipeline and is key in determining the final detector's accuracy. The introduction of AlexNet [Krizhevsky et al., 2012] popularized the use of deep learning for visual recognition. AlexNet was composed of stacks of convolutional layers followed by ReLU non-linearities and max-pooling. It achieved in the ILSVRC-2012 competition a top-5 test error rate[1] of 15.3%, compared to 26.2% which was achieved by the second-best entry. The much deeper VGG [Simonyan and Zisserman, 2015] and GoogleNet [Szegedy et al., 2014a] models further improved accuracy by introducing more stacks of convolutional layers, improving the imagenet top-5 error rates to 7.32% and 6.67%, respectively. He et al. [2015] introduced the even deeper Residual Networks (ResNet) that have greatly improved the state-of-the-art with the introduction of dozens of residual blocks. ResNet achieved a top-5 error rate of 3.57%. A residual network is a convolutional neural network that fits a residual mapping instead of the original, desired underlying mapping of an image. Figure 2.9 shows an illustration of the arquitecture of AlexNet (left), VGG (middle) and ResNet (right). It shows how many layers each network have and how deep they are relative to the others. These network architectures, often used in object detection frameworks like

[1]The top-5 error rate is the fraction of images for which the correct label is not among the five labels with the highest probability of all scores.

that of Girshick [2015], popularized the concept of transfer-learning in object-related detection tasks [Girshick, 2015; Ren et al., 2016; Zagoruyko et al., 2016]. Transfer-learning is a machine learning technique that consists of using knowledge gained during training in one type of problem and applying it to a different but related problem. For example, in object detection, it is common to use the information obtained by a network trained in a different (and much bigger) dataset like ImageNet [Russakovsky et al., 2015] to improve detection in another dataset like the MS COCO [Lin et al., 2014].

Context is known to play an important role in visual recognition [Torralba, 2003], and numerous ideas for exploiting context in ConvNets have been proposed. Sermanet et al. [2013b] used two contextual regions, centered on each object, for pedestrian detection. They used features from two feature maps and a context ratio, where pedestrians were 90 pixels high and 36 pixels were background. In Szegedy et al. [2014b], in addition to region-specific features, features from the whole image were used to improve region classification. He et al. [2014] implemented context in a more implicit way by combining ConvNet features prior to classification, using differently sized pooling regions. By varying the size of the pooling regions, they extracted feature maps of sizes 6×6, 3×3, 2×2 and 1×1 in a total of 50 bins, which were then fed to a classifier. More recently, Gidaris and Komodakis [2015] proposed to use ten contextual regions around each object with different crops, whereas Zagoruyko et al. [2016] used a similar approach with only four contextual regions organized in a foveal structure. Hence, context can be employed in many ways, for example by training the networks on segregated objects which are embedded into different backgrounds or by using separate regions of the whole image.

The use of a "multi-stage" classifier with different features at many convolutional layers was proposed by Sermanet et al. [2013b] for pedestrian detection, showing improved results. These "skip" architectures have recently become popular for semantic segmentation [Long et al., 2015] and general object detection [Bell et al., 2015; Zagoruyko et al., 2016]. They consist of using feature maps from layers of different stages of the network's pipeline and directly connecting them to the classifier network.

When originally introduced, object/region proposals were based on low-level grouping cues, edges, and superpixels [Alexe et al., 2012; Hosang et al., 2015a; Uijlings et al., 2013]. These approaches sparked interest for object detection frameworks like that of Girshick et al. [2014]; Girshick [2015]. Girshick's Fast R-CNN framework allows to use a pre-trained

ConvNet like a VGG or ResNet network as a feature extractor, and with the use of region proposals and a Region-of-Interest (RoI) pooling layer, features produced by these networks are fed into a classifier network. This framework allows to train a system in an end-to-end fashion. An efficient object detector can be trained in a few hours, achieving top-performing scores in popular datasets such as the Microsoft Common Objects in Context (COCO) detection challenge [Lin et al., 2014] with an average precision[1] of 41.5%. This dataset consists of 2.5 million labeled instances of 91 object categories in 328,000 images. Only 80 of the 91 object categories are used in the detection challenge. More recently, significant gains in the quality of region proposals have been obtained through the use of ConvNets [Pinheiro et al., 2015, 2016; Ren et al., 2016]. The basic idea is to apply a first ConvNet to detect regions of interest, which are then analyzed by another ConvNet. This improved the performance of detection frameworks like that of Girshick [2015], which achieved in the Pascal VOC 2012 dataset [Everingham et al., 2010] an average precision of 66%. Using the same framework, Zagoruyko et al. [2016] achieved top-perfoming results in the Microsoft COCO detection challenge using the Deepmask segmentation proposals of Pinheiro et al. [2015] combined with the VGG-A architecture of Simonyan and Zisserman [2015] with an average precision of 33.2%.

[1]Average precision is related to the area under the precision-recall curve for a class.

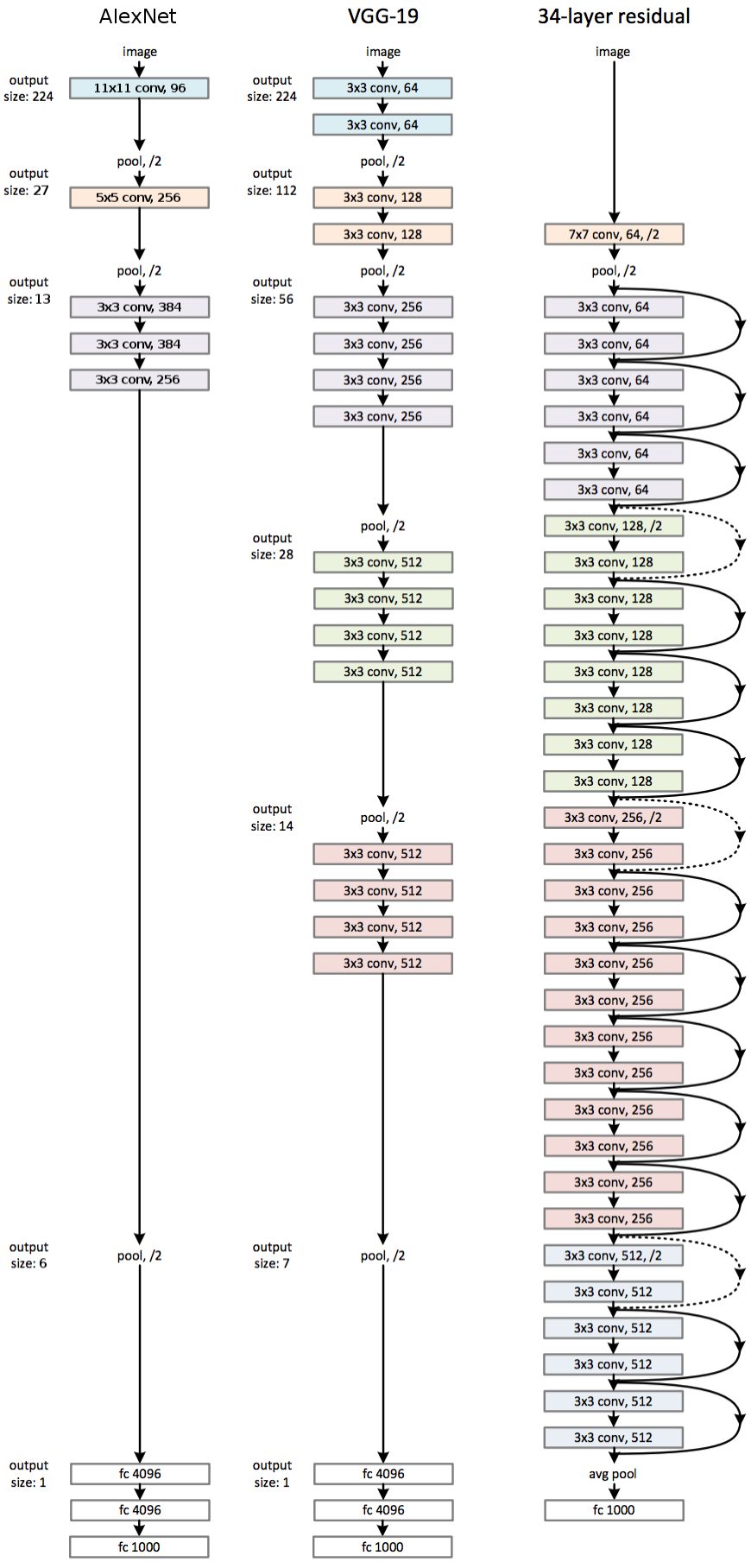

Figure 2.9: ConvNet architectures: AlexNet (left), VGG (middle) and ResNet (right).

Chapter 3

A biological and real-time framework for hand gestures and head poses

Human-robot interaction is an interdisciplinary research area that aims at the development of social robots. Since social robots are expected to interact with humans and understand their behavior through gestures and body movements, cognitive psychology and robot technology must be integrated. In this chapter we present a biological and real-time framework for detecting and tracking hands and heads. This framework is based on keypoints extracted by means of cortical V1 end-stopped cells. Detected keypoints and the cells' responses are used to classify the junction type. Through the combination of annotated keypoints in a hierarchical, multi-scale tree structure, moving and deformable hands can be segregated and tracked over time. By using hand templates with lines and edges at only a few scales, a hand's gestures can be recognized. Head tracking and pose detection are also implemented, which can be integrated with detection of facial expressions in the future. Through the combinations of head poses and hand gestures a large number of commands can be given to a robot.

Keywords: Hand gestures, Head pose, Biological framework.

3.1 Introduction

With the advent of newer and more complex technologies has come an increasing effort to make them easy to use. Some years ago computers were only used by specialized technicians, but nowadays even young children and elderly can use complex technology with great ease. The way how we use computers, cell phones and other devices has drastically changed because we began to research and implement natural ways of interacting with them. Part of that

research effort consists of the analysis of humans and their actions such that machines and software may be designed to react to our natural behaviors. One of the areas of interest for such interpretation is the recognition of human gestures, as they are used as a natural, intuitive and convenient way of communication in our daily life. The recognition of hand gestures can be widely applied in human-computer interfaces and interaction, games, human-robot interaction, augmented reality, etc.

Gesture analysis and recognition has been a popular research field for some years and numerous approaches have been developed. Interest in this area has spiked with the advent of low-cost and very reliable depth-based sensors like the Kinect [Li, 2012; Suau et al., 2012]. Although many gesture-based interfaces have been developed, to the best of our knowledge none of them is biologically inspired. Most of them are based on traditional methods from computer vision.

A method for hand tracking and motion detection using a sequence of stereo color frames was proposed by Kim et al. [2008]. Another approach, which consists of the recognition of gestures by tracking the trajectories of different body parts, was developed by Bandera et al. [2009]. In this method, trajectories are described by a set of keypoints and gestures are characterized through global properties of those trajectories. Suk et al. [2010] devised a method for recognizing hand gestures in continuous video streams by using a dynamic Bayesian network. Suau et al. [2012] presented a method to perform hand and head tracking using the Kinect. Two-handed gestures are recognized by analyzing the trajectories of both hands. Also using the Kinect, Li [2012] presented a method that is able to recognize nine different gestures and to identify fingers with high accuracy.

Although some methods do work fairly well for a specific purpose, they may not be suitable for a more profound analysis of human behavior and gestures because these are very complex. In this chapter we complement a biological and real-time framework for detecting and tracking hands [Farrajota et al., 2012] with head movements. This framework is based on multi-scale keypoints detected by means of models of cortical end-stopped cells [Rodrigues and du Buf, 2006, 2009b]. Cell responses around keypoints are used to classify the vertex type, for creating annotated keypoints [Farrajota et al., 2011]. The model has been extended by multi-scale line and edge information, also extracted by models of cortical cells. We also developed a model for optical flow based on annotated keypoints [Farrajota et al., 2012]. By integrating optical flow and annotated keypoints in a hierarchical, multi-scale tree structure,

deformable and moving objects can be segregated and tracked over time.

Hand and gesture recognition is obtained by using a simple line and edge template matching algorithm which relates previously stored templates with the acquired images across two scales. By using only five hand templates with lines and edges obtained at two different scales, a hand's gestures can be recognized. By tracking hands over time, false positives due to complex background patterns can be avoided. We also focus on head movements because they too can be an important part of human-robot interaction. When combined with the recognition of facial expressions it will provide invaluable information for natural human-computer and human-robot interaction. Our framework addresses the most common movements: leaning left/right and nodding (up/down). These can be used to give feedback to a robot, expressing doubts or affirming or criticizing actions, respectively. By combining a few head movements with hand gestures, a large number of instructions can be given.

The developed system does not require any prior calibration. Since the cell models have been optimized for running on a GPU, a speed of about 10 frames per second can be obtained, which is fast enough for real-time applications.

The chapter is organized as follows: in Section 3.2 we describe how keypoints are obtained and classified. In Section 3.3 the process to obtain the optical flow of consecutive frames from multi-scale keypoints is described in detail. Section 3.4 deals with the process to track hands and head, along with the process of recognizing hand gestures and head pose. Finally, in Section 3.5 some conclusions are provided.

3.2 Multi-scale lines, edges and keypoints

In cortical area V1 we find simple, complex and end-stopped cells [Rodrigues and du Buf, 2009b], which are thought to play an important role in coding the visual input: to extract multi-scale lines and edges and keypoint information (keypoints are line/edge vertices or junctions, but also blobs).

Responses of even and odd simple cells, corresponding to the real and imaginary parts of a Gabor filter [Rodrigues and du Buf, 2009b], are denoted by $R^E_{s,i}(x,y)$ and $R^O_{s,i}(x,y)$, i being the orientation (we use $N_\theta = 8$). The scale s is given by λ, the wavelength of the Gabor filters, in pixels. We use $4 \leq \lambda \leq 20$ with $\Delta\lambda = 4$. Responses of complex cells are modeled by the modulus $C_{s,i}(x,y) = [\{R^E_{s,i}(x,y)\}^2 + \{R^O_{s,i}(x,y)\}^2]^{1/2}$.

The basic scheme for line and edge detection is based on responses of simple cells: a positive or negative line is detected where R^E shows a local maximum or minimum, respectively, and R^O shows a zero crossing. In the case of edges the even and odd responses are swapped. This gives four possibilities for positive and negative events. An improved scheme [Rodrigues and du Buf, 2009b] consists of combining responses of simple and complex cells, i.e., simple cells serve to detect positions and event types, whereas complex cells are used to increase the confidence. Lateral and cross-orientation inhibition are used to suppress spurious cell responses beyond line and edge terminations, and assemblies of grouping cells serve to improve event continuity in the case of curved events. We denote the line and edge map by $LE_s(x, y)$.

Keypoints are based on cortical end-stopped cells [Rodrigues and du Buf, 2006]. They provide important information because they code local image complexity. Furthermore, since keypoints are caused by line and edge junctions, detected keypoints can be classified by the underlying vertex structure, such as K, L, T, + etc. This is very useful for most matching problems: object recognition, optical flow and stereo disparity. In this section we briefly describe the multi-scale keypoint detection and annotation processes. The original model has been improved such that multi-scale keypoints can be detected in real time [Terzić et al., 2013].

There are two types of end-stopped cells, single and double. These are applied to $C_{s,i}$ and are combined with tangential and radial inhibition schemes in order to obtain precise keypoint maps $K_s(x, y)$. For a detailed explanation with illustrations see Rodrigues and du Buf [2006] and Terzić et al. [2013].

In order to classify any detected keypoint, the responses of simple cells $R^E_{s,i}$ and $R^O_{s,i}$ are analyzed, but now using $N_\phi = 2N_\theta$ orientations, with $\phi_k = k\pi/N_\theta$ and $k = [0, N_\phi - 1]$. This means that for each of the 8 simple-cell orientations on $[0, \pi]$ there are two opposite analysis orientations on $[0, 2\pi]$, e.g., $\theta_1 = \pi/N_\theta$ results in $\phi_1 = \pi/N_\theta$ and $\phi_9 = 9\pi/N_\theta$. This division into response-analysis orientations is acceptable according to Hubel [1995], because a typical cell has a maximum response at some orientation and its response decreases on both sides, from 10 to 20 degrees, after which it declines steeply to zero; see also du Buf [1993].

Classifying keypoints is not a trivial task, mainly because responses of simple and complex cells, which code the underlying lines and edges at vertices, are unreliable due to response interference effects [du Buf, 1993]. This implies that responses must be analyzed in a neigh-

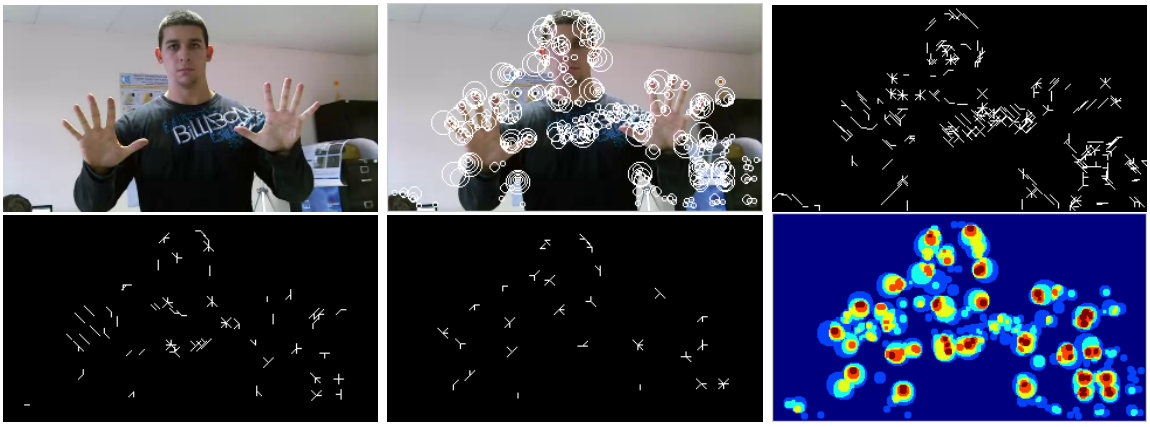

Figure 3.1: Left to right and top to bottom: input frame, keypoints detected at all 5 scales, annotated keypoints at scales $\lambda = 4$, 8 and 12, and the frame's saliency map where red indicates higher and blue lower saliency.

borhood around each keypoint, and the size of the neighborhood must be proportional to the scale of the cells. The validation of the line and edge orientations which contribute to the vertex structure is based on an analysis of the responses of complex cells $C_{s,i}(x, y)$. At a distance of λ, and for each direction ϕ_k, responses in that direction and in neighboring orientations ϕ_{k+l}, with $l = \{-2, -1, 0, 1, 2\}$, are summed with different weights equal to $1/2^{|l|}$. After this smoothing and detection of local maxima, each keypoint is then annotated by a descriptor of 16 bits which codes the detected orientations. In the case of keypoints caused by blobs with no underlying line and edge structures, all 16 bits are zero.

This method is an improvement of the previous method [Farrajota et al., 2011]. It provides a more detailed descriptor of the underlying line and edge structures, with a significant increase in performance and with a negligible loss of precision. The first five images in Fig. 3.1 illustrate keypoint detection and annotation at the given scales. For more illustrations see Rodrigues and du Buf [2006].

3.3 Optical flow

Keypoint detection may occur in cortical areas V1 and V2, whereas keypoint annotation requires bigger receptive fields and could occur in V4. Optical flow is then processed in areas V5/MT and MST, which are related to object and ego motion for controlling eye and head movements.

Optical flow is determined by matching annotated keypoints in successive camera frames, but only by matching keypoints which may belong to a same object. To this purpose we use

regions defined by saliency maps. Such maps are created by summing detected keypoints over all scales s, such that keypoints which are stable over scale intervals yield high peaks. In order to connect the individual peaks and yield larger regions, relaxation areas proportional to the filter scales are applied [Rodrigues and du Buf, 2006]. Here we simplify the computation of saliency maps by simply summing the responses of end-stopped cells at all scales, which is much faster and yields similar results. Figure 3.1 (bottom-right) illustrates a saliency map.

We apply a multi-scale tree structure in which at a very coarse scale a root keypoint defines a single object, and at progressively finer scales more keypoints are found which convey the object's details. For optical flow we use five scales: $\lambda = [4, 20]$ with $\Delta\lambda = 4$. All keypoints at $\lambda = 20$ are supposed to represent individual objects, although we know that it is possible that several of those keypoints may belong to a same object. Each keypoint at a coarse scale is related to one or more keypoints at one finer scale, which can be slightly displaced. This relation is modeled by down-projection using grouping cells with a circular axonic field, the size of which (λ) defines the region of influence, and this process continues until the finest scale is reached; see Farrajota et al. [2011].

As mentioned above, at a very coarse scale each keypoint – or central keypoint CKP – should correspond to an individual object [Rodrigues and du Buf, 2006]. However, at the coarsest scale applied here, $\lambda = 20$, this may not be the case and an object may cause several keypoints. In order to determine which keypoints could belong to the same object we combine saliency maps with the multi-scale tree structure.

At this point we have, for each frame, the tree structure which links the keypoints over scales, from coarse to fine, with associated regions of influence at the finest scale. We also have the saliency map obtained by summing responses of end-stopped cells over all scales. The latter, after thresholding, yields segregated regions which are intersected with the regions of influence of the tree. Therefore, the intersected regions link keypoints at the finest scale to the segregated regions which are supposed to represent individual objects.

Now, each annotated keypoint of frame i can be compared with all annotated keypoints in frame $i - 1$. This is done at all scales, but the comparison is restricted to an area with radius 2λ instead of λ at each scale in order to allow for larger translations and rotations. In addition, (1) at fine scales many keypoints outside the area can be skipped since they are not likely to match over large distances, and (2) at coarse scales there are less keypoints, λ is bigger, and therefore larger distances (motions) are represented there. The matching process,

as for building the tree, is now done top-down. Previously it was done bottom-up [Farrajota et al., 2011]. Due to the use of a more detailed descriptor for keypoint classification than in Farrajota et al. [2011], matching keypoints at the coarsest scale provides sufficient accuracy to correctly match entire tree structures. An additional gain in performance is due to the reduced number of comparisons at finer scales, because of existing dependencies between keypoints in the branches of the tree structure. Keypoints are matched by combining three similarity criteria with different weight factors:

(a) The distance D serves to emphasize keypoints which are closer to the center of the matching area. For having $D = 1$ at the center and $D = 0$ at radius 2λ, we use $D = (2\lambda - d)/2\lambda$ with d the Euclidean distance (this can be replaced by dynamic feature routing [Farrajota et al., 2011; Rodrigues and du Buf, 2009a]).

(b) The orientation error O measures the correlation of the attributed orientations, but with an angular relaxation interval of $\pm 2\pi/N_\theta$ applied to all orientations such that also a rotation of the vertex structure is allowed. Similar to D, the summed differences are combined such that $O = 1$ indicates good correspondence and $O = 0$ a lack of correspondence. Obviously, keypoints marked "blob" do not have orientations and are treated separately.

(c) The tree correspondence C measures the number of matched keypoints at finer scales, i.e., at any scale coarser than the finest one. The keypoint candidates to be matched in frame i and in the area with radius 2λ are linked in the tree to localized sets of keypoints at all finer scales. The number of linked keypoints which have been matched is divided by the total number of linked keypoints. This is achieved by sets of grouping cells at all but the finest scale which sum the number of linked keypoints in the tree, both matched and all; for more details see Farrajota et al. [2011].

The three parameters are combined by grouping cells which can establish a link between keypoints in frame $i - 1$ and i. Mathematically we use the similarity measure $S = \alpha O + \beta C + \gamma D$, with $\alpha = 0.4$ and $\beta = \gamma = 0.3$. These values were determined empirically. The candidate keypoint with the highest value of S in the area (radius 2λ) is selected and the vector between the keypoint in frame $i - 1$ and the matched one in frame i is computed. Remaining candidates in the area can be matched to other keypoints in frame i, provided they are in their local areas. Keypoints which cannot be matched are discarded. Figure 3.2 shows a sequence with tracked hands by using optical flow.

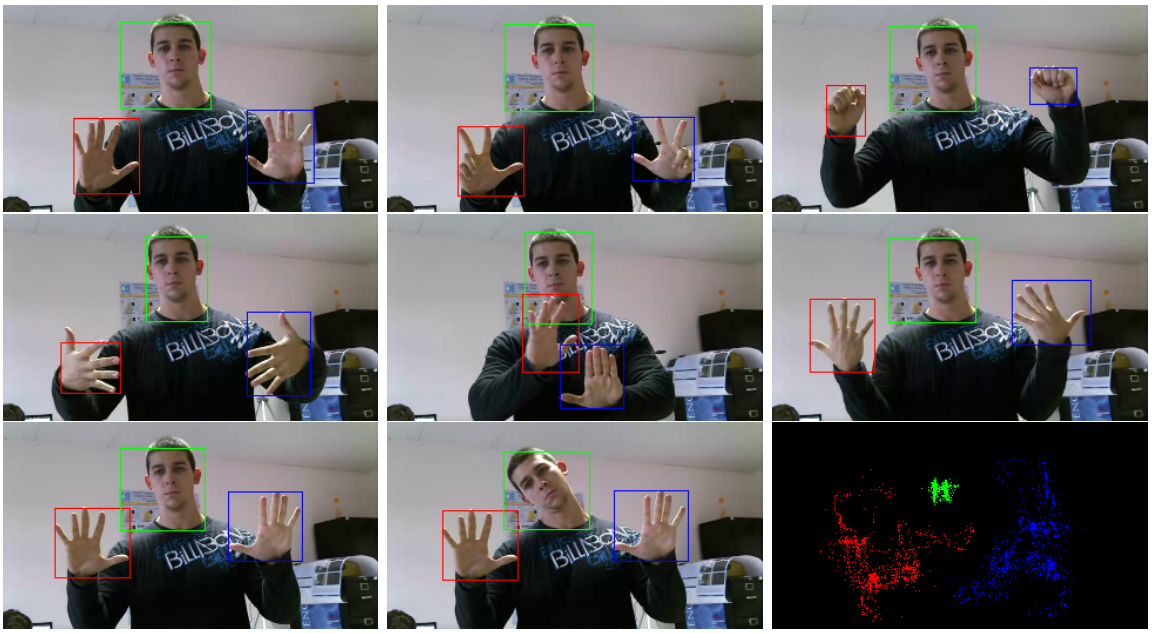

Figure 3.2: The optical flow model applied to a person while performing several hand and head gestures. Hands and head are marked by their bounding boxes. The bottom-right image shows the combined centers of the boxes.

3.4 Hand/head tracking and gesture/pose recognition

To initialize the tracking and recognition process, it is only required that at the beginning the user stands still, looking straight ahead and showing the palms of both hands to the camera. As the first step in the processes that will be described in this section, we use skin color segmentation to detect both hands and the head as previously applied in Saleiro et al. [2009]. If I is an input frame, we can use the following expression to get a binary skin image, I_s, where skin is marked in black and all the rest is white: $I_s(x, y) = 0$ if $\varphi[I(x, y)] = 1$, otherwise $I_s(x, y) = 255$, where $\varphi = [(R > 95) \wedge (G > 40) \wedge (B > 20) \wedge ((\max\{R, G, B\} - \min\{R, G, B\}) > 15) \wedge (|R - G| > 15) \wedge (R > G) \wedge (R > B)]$, with $(R, G, B) \in [0, 255]$.

After obtaining the skin regions we can obtain three regions: left hand, right hand and head. Then we apply two filters: the first one is an erosion which removes small regions, and the second one is a dilation which makes the remaining regions more homogeneous. After this we apply a fast blob detection algorithm [Saleiro et al., 2009] to obtain the coordinates and sizes of the three biggest skin regions. The region with the highest y coordinate will be considered as being the head. The system will use the head blob's dimensions to calculate the reference ratio $R_r = h/w$, with h the height and w the width, as a reference for the neutral pose. The detection of head poses is done like previously in Saleiro et al. [2009]. We

use five head poses: face straight forward, head up, head down, head leaning to the left and head leaning to the right. To detect the up and down poses we use a very simple method which consists of comparing the blob's actual ratio, R_a, to an upper (U_{thr}) and a lower threshold (L_{thr}). The latter are determined from the reference ratio R_r computed during the initialization: $U_{\text{thr}} = 1.1 \times R_r$ and $L_{\text{thr}} = 0.9 \times R_r$.

To detect the left- and right-leaning poses, two vertical lines, at distances $w/6$ and $5w/6$ inside the blob's box are considered. The average position of black pixels on each of these lines is calculated and the two resulting positions are used to detect the two poses: when the user leans the head to one side, the average positions will go up and down relative to the middle of the box ($h/2$). A minimum vertical distance MVD of $0.2 \times h$ between both positions was determined experimentally, such that small lateral movements can be ignored. The two poses are detected when (a) the vertical distance between the two positions is larger than the MVD, and (b) one of the positions is higher than $h/2$. The latter position determines the side of the movement: left or right.

While the head will normally be at a static location, hands may be constantly moving and therefore they must be tracked. To do that we employ the optical flow as explained in the previous section. The recognition of hand gestures is more complex than the detection of the head's poses. To recognize hand gestures, we need to use a single template, at a few different scales, of each gesture. The templates are previously prepared so that they are available for online matching when the system is working. To prepare the templates, we apply the previously described line- and edge-extraction algorithm at two different scales, and then dilate the resulting maps to make the templates more robust against small differences between them and the real frames containing moving hands. Each template is a binary image which contains white lines against a black background. Example templates are shown in Fig. 3.3.

To perform the template matching in a fast way, we only compare the templates with the regions tracked by optical flow. This way no processing time is wasted in other image regions. The matching process is done in two steps: (a) direct template matching and (b) template density matching. In step (a) we take a tracked region and apply the same process that was used to prepare the templates. Then we shift each template over the tracked region and at each shift position we compare them, pixel by pixel, and count the number of white pixels, P_w. Basically this is a 2D correlation process. We divide P_w by the total number of

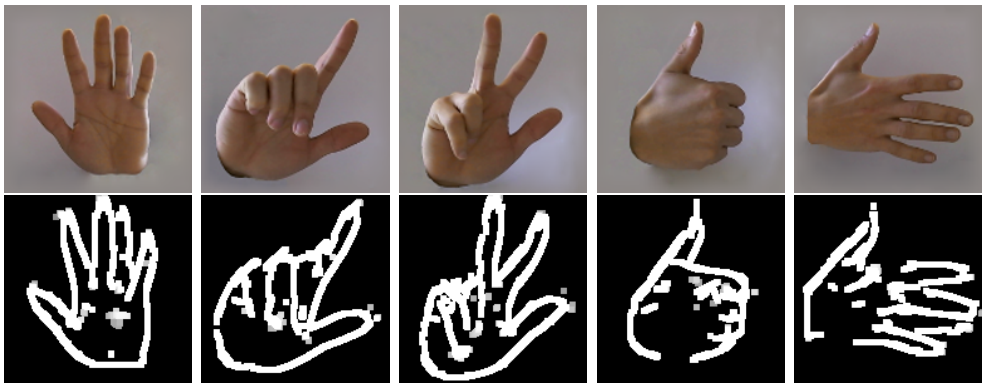

Figure 3.3: Top: five hand gestures. Bottom: their dilated templates at scale $\lambda = 8$.

white pixels in the template, P_{wt}, and store the resulting value in a probability map at the center position of the (shifted) template. The result is a 2D histogram or correlation matrix in which higher values indicate a better correspondence between the tracked region and the template.

In step (b), template density matching, we verify whether the test region has the same ratio of white and black pixels as the template. This must be done because if only direct matching was used, some complex textures, for example on a (moving) T-shirt or (static) background can result in false detections of some templates. Again we use the shifting window with the same size of the template and, for each shift position, we calculate the ratio between the number of white pixels and the total number of pixels in the window, $R = W_p/T_p$, where W_p is the number of white pixels and T_p the total number of pixels. Like before, this ratio is stored in a similar probability map.

After these two steps we combine both maps for each template, giving a 70% weight to the first map and 30% to the second one. This yields a single probability map for each template. This process is applied at the two scales used ($\lambda = \{8, 12\}$), and the two probability maps are mixed prior to multi-scale recognition, thereby giving equal weights to the two scales. At this point we have a single but multi-scale probability map for each template. Every time that a value greater than a threshold value occurs ($T = 60$ was experimentally determined), the system considers that the gesture which corresponds to that map has been recognized, and at the peak location. When more than one gesture is recognized, only the one with the greatest probability value will prevail. Figure 3.4 illustrates the matching process (top) together with the detection of head poses (bottom).

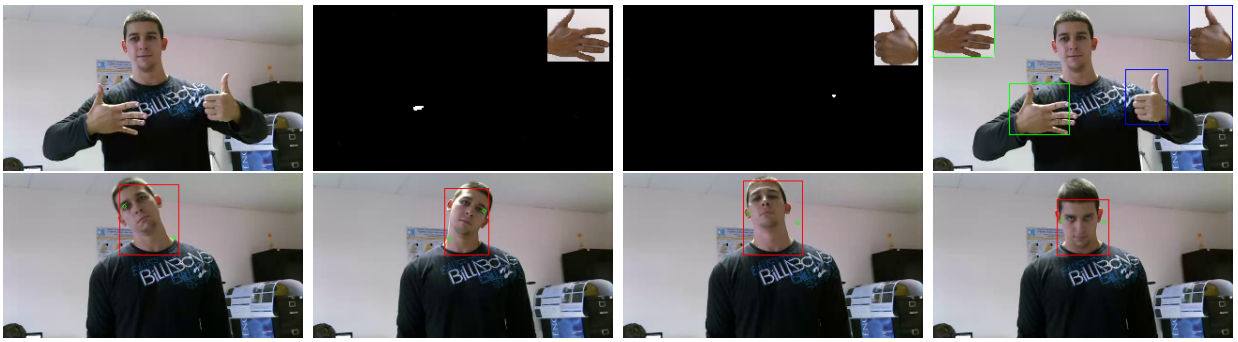

Figure 3.4: The top row illustrates hand gesture recognition: input image, thresholded probability maps of the two detected gestures with their peaks in white, and the final result. The bottom row shows examples of the recognition of four head poses: right, left, up and down. Along the vertical edges of the bounding box, the two green points show the average location of skin-colored pixels.

3.5 Discussion

In this chapter we presented a biologically inspired method for hand detection, tracking, and gesture recognition. By using optimized algorithms for the detection of keypoints plus lines and edges, and by selecting only a few scales, the method can run in real time. Even when using a cheap HD webcam, very good results can be obtained. This we expected due to our previous experience with cortical models: multi-scale keypoints, lines and edges provide very useful information to generate saliency maps for Focus of Attention or to detect faces by grouping facial landmarks defined by keypoints at eyes, nose and mouth [Rodrigues and du Buf, 2006]. De Sousa et al. [2010] were able to use lines and edges to recognize facial expressions with success, and Rodrigues and du Buf [2009b] showed that lines and edges are very useful for face and object recognition. The method included here for the detection of head poses is not biological, but in principle we can integrate our methods for face detection and recognition of facial expressions.

Biologically inspired methods involve many filter kernels, here in eight orientations and at several scales. In order to achieve real-time processing, we only use five scales for optical flow and region segregation. For gesture recognition we use lines and edges at only two scales. The system's main limitation is the costly filtering. The optimized GPU implementation allows us to process at least 10 frames/s with a maximum resolution of 600×400 pixels and using at least 6 scales if coarser scales are used. The main bottleneck for using large images and fine scales is the 1 GByte of memory of the GPU, because of the Gaussian pyramid employed in the filtering.

Future work will focus on motion prediction, a process that occurs in cortical area MST. We also intend to increase the precision such that individual fingers can be detected in combination with a larger number of gestures. The ultimate goal is to apply 3D processing in the entire process, with emphasis on body language. This can be done by using cheap off-the-shelf solutions like a Kinect or two webcams with a biological disparity model.

Chapter 4

Pedestrian Detection

Pedestrian detection and tracking remains a popular issue in computer vision, with many applications in robotics, surveillance, security and telecare systems, especially when connected with Smart Cities and Smart Destinations. As a particular case of object detection, pedestrian detection in general is a difficult task due to large variability of features caused by different scales, views and occlusions. Typically, small and occluded pedestrians are harder to detect because of fewer discriminative features if compared to large, and well-visible pedestrians. In order to overcome this we use convolutional features from different stages in a deep Convolutional Neural Network (CNN), with the idea of combining more global features with finer details. This framework extends the Fast R-CNN framework for the combination of several convolutional features from different stages of the used CNN to improve the network's detection accuracy. The Caltech Pedestrian dataset was used to train and evaluate the proposed method.

Keywords: Object Detection, Pedestrian detection, Deep learning, Multi-stage features.

4.1 Introduction

Lopez de Avila [2015] defined a Smart Destination as an innovative tourist destination with an infrastructure of state-of-the-art technology. It guarantees the sustainable development of tourist areas, facilitates the visitor's interaction with and integration into the surroundings, increases the quality of the experience at the destination, and it improves residents' quality of life. Gretzel [2011] mentioned Smart Destinations as special cases of Smart Cities, which apply smart city principles to urban or rural areas, and not only consider residents but also

Figure 4.1: Illustration of the differences of feature maps due to person size. Top: three images from the Caltech Pedestrian dataset [Dollár et al., 2012] show pedestrians of various sizes and aspect ratios. Red rectangles indicate pedestrians with a height greater than 160 pixels; blue rectangles correspond to heights between 80 and 160 pixels; green rectangles correspond to pedestrians smaller than 80 pixels. Bottom: pedestrian crops of different sizes (first row) and their corresponding feature maps (second row). Smaller pedestrians (at right) convey significantly different feature maps compared to bigger ones (at left).

tourists in efforts to support mobility, resource availability and allocation, sustainability and quality of life.

Pedestrian detection and tracking is a topic with a clear application in Smart Cities and Smart Destinations, in addition to many other applications in robotics, surveillance, security and telecare systems. For instance, the latter can be used to monitor senior persons for the detection of abnormal behavior related to chronic or new ailments [Farrajota et al., 2016a]. In terms of security, with all available cameras in the cities it is important to detect on-the-fly suspicious behaviors in order to alert authorities. Both telecare and security systems seem very distinct at the beginning, but the principle is the same: (a) detect visible persons and pedestrians in an environment; and (b) classify the movements and actions of those persons.

Detecting pedestrians by identifying visible persons is difficult because of variations in the target appearance, pose, size, lighting and occlusion. Moreover, each independent variation

affects detection differently, but the two main effects that hamper detection most are scale and occlusion [Li et al., 2015]. For example, in the Caltech dataset [Dollár et al., 2012], many pedestrians are small: over 60% of all labeled persons in the test set have a height smaller than 100 pixels. In addition to having reduced size, other effects like blurring and lighting make them difficult to distinguish from the background. Also, large-size pedestrians usually show different visual characteristics than smaller-sized ones (see Fig. 4.1).

To address these issues, existing work has tackled the scale variation problem in several ways. Data augmentation techniques [Girshick, 2015] like resizing and multiple scales have been used to increase robustness to scale variations. Other methods used a single model but with several filters tuned to specific scales which are applied to all pedestrians with various sizes. This, however, cannot solve the problems due to the large intra-class variation of small and large persons. Recently, another method [Li et al., 2015] exploited the different characteristics of pedestrians with various sizes by adopting a divide-and-conquer strategy. Li et al. combined a large-size sub-network with a small-size one for detecting pedestrians of varying sizes. The use of a weighted score of both sub-network responses significantly increased accuracy because each network is tuned to different features.

In this chapter we pursue a different strategy in order to cope with feature differences due to person sizes. We present an object detection framework which uses multi-stage features of a deep Convolutional Neural Network (CNN) to improve detection accuracy. By using feature maps from different convolutional layers with different receptive field sizes, we can cope with some ambiguity in discerning pedestrians from background due to the size variability. Since the size of a receptive field depends on the depth of its layer in the network, different fields will code different features of differently sized pedestrians. The proposed method extends the Fast R-CNN [Girshick, 2015] framework by using and combining multiple feature maps from different stages of a CNN for classification. The Caltech pedestrian dataset will be used to train and test the proposed method.

The main contribution of this chapter is the integration of multiple features from different stages of a deep CNN to improve detection accuracy. While most detection methods do not take advantage of more information available in the CNN pipeline, here we investigate the usefulness of employing more features maps besides the last convolutional layer, which holds more complex features than the previous layers. Also, we investigate various sources of features in a CNN pipeline and their effects on the final accuracy. A secondary contribution

of our work is the analysis of the performance of several different types of architectures.

The rest of the chapter is organized as follows: in Sec. 4.2 we describe related work on pedestrian detection. In Sec. 4.3 we provide an overview of the proposed method's functionality and a detailed description of our framework architecture. In Sec. 4.4 we provide details concerning the training and testing of several types of architectures and layer combinations in order to achieve the final model, and also we compare results of the proposed method with other state-of-the-art methods. Finally, in Sec. 4.5 we provide some conclusions concerning the proposed method.

4.2 Related work

Due to significant advances in recent years, deep learning methods now provide the leading artificial vision framework for classification, categorization and detection tasks. Particularly pedestrian detection has received a lot of attention over the past decade [Dalal and Triggs, 2005; Dollár et al., 2009; Li et al., 2015; Ouyang and Wang, 2012, 2013; Sermanet et al., 2013b; Wang et al., 2009]. This is due to many applications involving video surveillance, robotics and human-computer interaction. Current methods for pedestrian detection can be grouped into two categories: models based on hand-crafted features [Dalal and Triggs, 2005; Dollár et al., 2014, 2009; Viola et al., 2005; Wang et al., 2009] and deep models [Ouyang and Wang, 2012, 2013; Sermanet et al., 2013b]. In the first category, detection algorithms rely on features such as Haar wavelets [Viola et al., 2005], Histograms of Oriented Gradients (HOG) [Dalal and Triggs, 2005] or Histograms of Oriented Gradients-Local Binary Pattern (HOG-LBP) [Wang et al., 2009], which are then used to train Support Vector Machines (SVMs) [Dalal and Triggs, 2005] or to boost other classifiers [Dollár et al., 2009], in order to detect either entire persons or hierarchies of parts. In the second category, deep Convolutional Neural Networks (CNNs) [Girshick, 2015; He et al., 2014; Simonyan and Zisserman, 2015] provide a unified, jointly optimizable framework for feature extraction and classification from raw pixel images. Most methods treat pedestrian detection as a mere binary classification task and cannot grasp more difficult intra-class variations which are known to complicate person detection.

• **Hand-crafted methods** [Dalal and Triggs, 2005; Dollár et al., 2014, 2009; Viola et al., 2005; Wang et al., 2009]: These approaches use either global models with full-body appearance [Dalal and Triggs, 2005], assemblies of local features [Viola et al., 2005], or part

detectors [Mikolajczyk et al., 2004]. Many descriptor-based detectors have been used. Dalal and Triggs [2005] HOG histograms extended the idea of the popular local Scale Invariant Feature Transform (SIFT) descriptor [Lowe, 2004] to entire objects. Other authors have proposed additional features to improve the representation of the descriptor, namely the use of color through self-similarity features (CSS) [Walk et al., 2010], texture through block-based Local Binary Patterns (LBP) [Ahonen et al., 2006], and the design of efficient gradient-based features via integral channels [Dollár et al., 2009]. Moreover, the combination of such features was shown to improve the overall accuracy w.r.t. some baseline accuracy. Wang et al. [2009] combined LBP and HOG features to deal with partial occlusions of pedestrians. Dollár et al. [2009] proposed the Integral Channel Features (ICF) and Aggregated Channel Features (ACF) [Dollár et al., 2014]. Both methods consist of combining information from gradient histograms in LUV color space. Furthermore, Nam et al. [2014] extended ACF by an efficient feature transform that removes correlations in local image neighborhoods [Hariharan et al., 2012]. Cai et al. [2015] combined features of different complexities to find an optimal trade-off between complexity and accuracy. These methods are generally cheap to compute and some even perform detection at very high frame rates (+100 fps) [Benenson et al., 2012], and the best sliding window algorithm [Nam et al., 2014] scores a 25% miss rate on the Caltech Pedestrian Dataset [Dollár et al., 2012].

• **Deep learning methods** [Girshick, 2015; Ouyang and Wang, 2012, 2013; Sermanet et al., 2013b]: The advantage and usefulness of deep learning methods is based on their ability to learn complex features from raw pixels. Sermanet et al. [2013b] applied convolutional sparse coding to the unsupervised pre-training of a CNN for pedestrian detection. Tian et al. [2015b] optimized pedestrian detection by using semantic attributes of both pedestrians and scene. Xu et al. [2014] detected the input pattern at different scales in multiple columns simultaneously and concatenated the top-layer feature maps from all columns for final classification. Li et al. [2015] developed a framework which consisted of one large-size and one small-size sub-network, and fusing the results using a scale-aware weighting mechanism. These methods out-perform hand-crafted ones accuracy wise having miss rates bellow 25% on the Caltech Pedestrian Dataset [Dollár et al., 2012], although being significantly slower in detection time compared to hand-crafted methods.

Here we investigate the use of additional information from previous convolutional layers as in Sermanet et al. [2013b] to improve accuracy. Sermanet et al. [2013b] investigated the

use of additional information as provided by previous feature map stages for increasing the overall performance of a network. However, instead of using the immediate convolutional layers before the last one which have the same (coarse) spatial convolution stride, we use information from earlier layers in the convolution pipeline where finer information is available. Usually, CNNs used in classification tasks are organized in a strictly feed-forward manner where each layer takes the output of the previous layer as its input. In this way, high-level features are obtained after a few stages of convolutions and subsampling. With this in mind, by branching the outputs of lower levels into the top classifier, features that encode both global structures and local details, such as a global silhouette and face components in the case of person detection, can be useful for better class separation.

• **Fast R-CNN framework**: Shared computation of convolutions has been widely used for efficient and accurate visual recognition, and several methods take advantage of this [Girshick, 2015; Sermanet et al., 2013a]. The Fast R-CNN [Girshick, 2015] detector allows efficient end-to-end training on shared convolutional features and it showed good accuracy and speed. It takes around 350ms to classify over 1000 detection windows on 640×480 pixel images in the Caltech Pedestrian Dataset [Dollár et al., 2012], outperforming most methods in accuracy with a miss rate less than 12%. Here we use this framework and extend it to use feature maps of extra convolutional layers in order to increase accuracy.

4.3 Multi-stage networks

In this section we provide an overview of our method (Sec. 4.3.1), describe the network's architecture details (Sec. 4.3.2) and explain how Region-of-Interest (RoI) proposals are generated (Sec. 4.3.3).

4.3.1 Method Overview

The proposed method, coined Multi-Stage Feature (MSF) Fast R-CNN, is capable of integrating several feature maps from multiple convolutional layers of a CNN and to combine them in a single network of fully-connected layers for classification. It works as follows: the model takes images and a number of RoI proposals as input and then outputs detection results. The model is composed of three main components: i) a CNN to extract feature maps from convolutional layers; ii) a RoI pooling layer that extracts sub-sections defined by

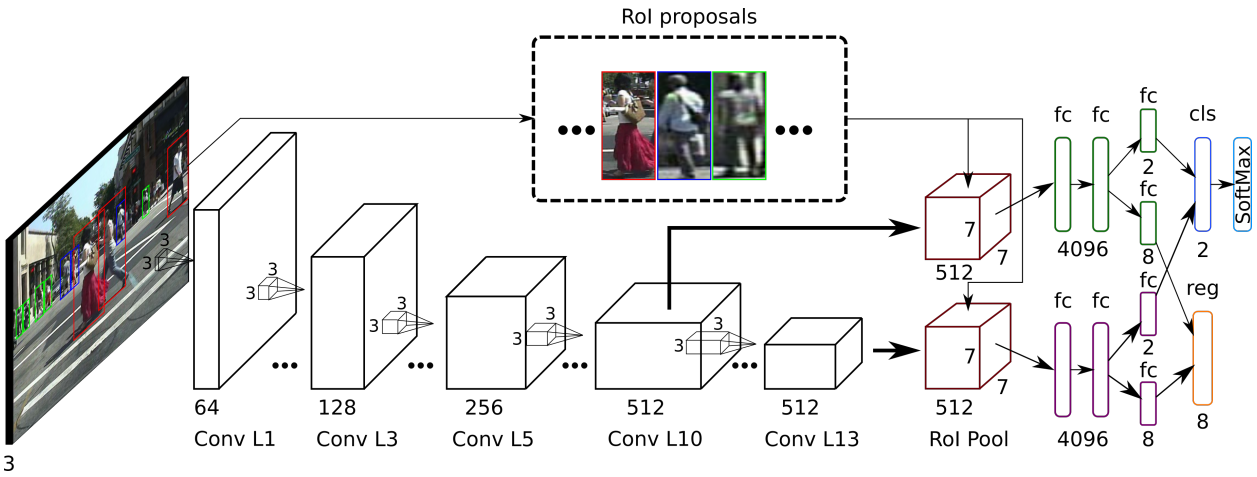

Figure 4.2: Illustration of the architecture of our Multi-Stage Features (MSF) Fast R-CNN model. First, features are extracted from an input image using a sequence of convolutional layers from a VGG16 [Simonyan and Zisserman, 2015] network, keeping only the layers up to the last max pooling layer for feature extraction. Next, convolutional feature maps are extracted from the 10th and 13th convolutional layers and each are then fed into a separate RoI pooling layer.Finally, they are fed into four fully-connected layers, ending up with two output fully-connected layers: one (*cls*) outputs classification scores over 2 object classes (pedestrian and background); the other (*reg*) outputs refined bounding-box positions.

the input RoI proposal coordinates in the image from two convolutional feature maps in the CNN pipeline; and iii) a final network classifies the extracted sub-sections (pedestrian or background class) and it also outputs refined bounding-box positions. With the integration of multiple feature maps, our method can capture more unique combinations of characteristics of pedestrians. This is useful for coping with effects like varying sizes and occlusions. The combined features provide a more detailed characterization of a region, which aids a classifier to better separate (distinguish) pedestrians from background. The proposed architecture is based on the popular Fast R-CNN Object detection framework [Girshick, 2015] because of its speed and simplicity during training and detection, and also because of its versatility for being extended to use multiple feature maps in a single feed-forward network.

4.3.2 MSF Fast R-CNN Architecture

Figure 4.2 shows the architecture of the Multi-Stage Feature (MSF) Fast R-CNN network in detail. For detection, our model receives an image and several RoI proposal coordinates as input. The image is passed through several convolutional layers, non-linear functions and max pooling layers in order to extract feature maps. Then, features from two convolutional layers are pooled in a RoI Pooling layer which selects a fixed-length feature vector of the

two feature maps, the length depending on the coordinates as provided by RoI proposal box. These two feature vectors are then fed into two separate classification networks which are composed of two sequential fully-connected layers followed by two parallel ouput layers with smaller size. Finally, the outputs of the two networks are connected to two parallel output layers which produce two output vectors per object proposal. The first one outputs classification scores of two object classes (pedestrian and background) which are fed into a SoftMax layer. The latter produces probabilities of the two classes for each input object proposal. The second layer is a bounding-box regressor which outputs bounding-box position refinements, but only for the pedestrian object class.

4.3.3 RoI Proposals Detection

We use the ACF [Dollár et al., 2014] and LDCF [Nam et al., 2014] detectors in order to generate RoI proposals. These detectors are publicly available and both use a fast sliding window strategy that performs quite well for rigid object detection. Also, they can be trained to detect specific object categories like pedestrians. This allows us to generate high quality RoI proposals quickly and efficiently. We use the Caltech dataset [Dollár et al., 2012] for training the ACF pedestrian detector, and the generated proposals are then used as input to train our Fast R-CNN network. For evaluation we use a pre-trained LDCF detector to generate proposals for test images because of its smaller miss-rate on the Caltech Pedestrian Dataset [Dollár et al., 2012] compared to the ACF detector, thus improving the overall performance of our detector.

4.4 Experiments

In this section we provide details on the dataset used for training and evaluation (Sec. 4.4.1), the actual implementation (Sec. 4.4.2), an analysis of some key aspects of the architecture (Sec. 4.4.3) and we will benchmark results with other state-of-the-art methods for pedestrian detection (Sec. 4.4.4).

4.4.1 Dataset

We only use the Caltech Pedestrian dataset [Dollár et al., 2012]. This dataset and its benchmark is one of the most popular and challenging publicly available datasets. It consists

of about 10 hours of 30 fps video collected from a vehicle which was driving in urban traffic. Each frame has been densely annotated with bounding boxes of pedestrians. There are about 350,000 bounding boxes of about 2,300 unique pedestrians labeled in 250,000 frames. The normal evaluation procedure is to use the standard "Reasonable" train/test setting, which provides 4250 frames with about 2000 annotated pedestrians for training, and for testing the set provides 4024 frames with roughly 1000 pedestrians. Since the videos are fully annotated, the amount of training data can be increased by re-sampling the videos. Following Hosang et al. [2015b], we increased the training data by selecting every third frame (instead of one out of thirty frames as in the standard setup) from each video of the training data. This resulted in a tenfold increase of available annotations: about 20,000 annotated pedestrians extracted from 42782 frames. We followed the proposed evaluation protocol by measuring the log average miss rate over nine points ranging from 10^{-2} to 10^{0} False-Positives-Per-Image (FPPI). We also compare performance with the best-performing methods as suggested by the Caltech benchmark on the "reasonable" subsets, where pedestrians have a height of at least 50 pixels and are occluded at most 65%.

4.4.2 Implementation Details

4.4.2.1 RoI Proposal Generation

We first trained an ACF detector using the full Caltech training dataset to generate high quality pedestrian proposals for training the R-CNN network. We used similar parameters as in Nam et al. [2014]. The depth of the trees was increased twofold (from 4096 to 8192), and the number of negative samples was increased to 100,000. We applied a calibration factor of 0.1 and a threshold of -1 to generate many RoI proposals which are then used as input to the R-CNN network. Additionally, we used RoI proposals generated by the LDCF detector on the Caltech training dataset to increase the number of available proposals for training. In order to further increase the number of overlapping pedestrians, we slightly jittered the box coordinates. To that purpose, we jittered the (x, y) box coordinates around an offset of $\pm 10\%$ of the RoI box width, with small step sizes of $1/4$ of the box width and height in the x and y coordinates, respectively, to generate new RoI coordinates around the original box. This increased the total number of generated proposals by a factor of sixteen. For evaluation, we used the default parameters of the publicly available LDCF detector without any additional data augmentation to generate test RoI proposals.

4.4.2.2 MSF Fast R-CNN detector

We used a VGG16 [Simonyan and Zisserman, 2015] ConvNet model for feature extraction, which has been trained on ImageNet [Russakovsky et al., 2015]. This is standard practice for deep networks, since the number of parameters is much larger than the available data for training a specific application and it provides a good starting point for the actual training. The network was trained on the ILSVR2012 [Russakovsky et al., 2015] dataset with 1 million images of 224×224 pixels. We used all layers up to the last max-pooling layer, and during training the first four convolutional layers in the network had their parameters fixed (i.e., they were not optimized). Furthermore, we extracted feature maps from two convolutional layers in the CNN pipeline, from layer 13 which is the last CNN layer and from layer 10. The RoI pooling layers extract feature maps for each RoI proposal with a fixed resolution of 7×7 grid pixels.

The networks were trained using stochastic gradient descent with a momentum of 0.9 and a weight decay of 0.0005. All network weights which were not pre-trained on ImageNet were randomly initialized with a uniform distribution on [-0.01, 0.01]. We used mini-batches of 128 randomly sampled object proposals from two images; 25% of these were positive RoI proposals having an intersect-over-union (IoU) of at least 0.5 with ground truth boxes. The remaining samples were negative object proposals; 25% of these were RoI proposals having an IoU with the ground truth box in the interval [0.1, 0.5), and the remaining 75% of the object proposals had 0% overlap with ground truth boxes. Dropout with 50% chance was applied to all fully-connected layers of the classifier except for the first one, and batch normalization [Ioffe and Szegedy, 2015] was used for faster convergence during training. We updated the network parameters with a learning rate of 0.001 for 4 epochs and then reduced it by $1/10^{th}$ for an extra 3 epochs, with a total of 7 epochs for training using the same combined loss as in Girshick [2015]. During training and testing, the scale of the input image was set to 800 pixels on the shortest side. For data augmentation, images were horizontally flipped with a probability of 50%.

The implementation was done using the popular deep learning Torch7 [Collobert et al., 2011a] platform, and the network was trained on two NVIDIA GeForce GTX TITAN Black GPUs with 6GB of memory each.

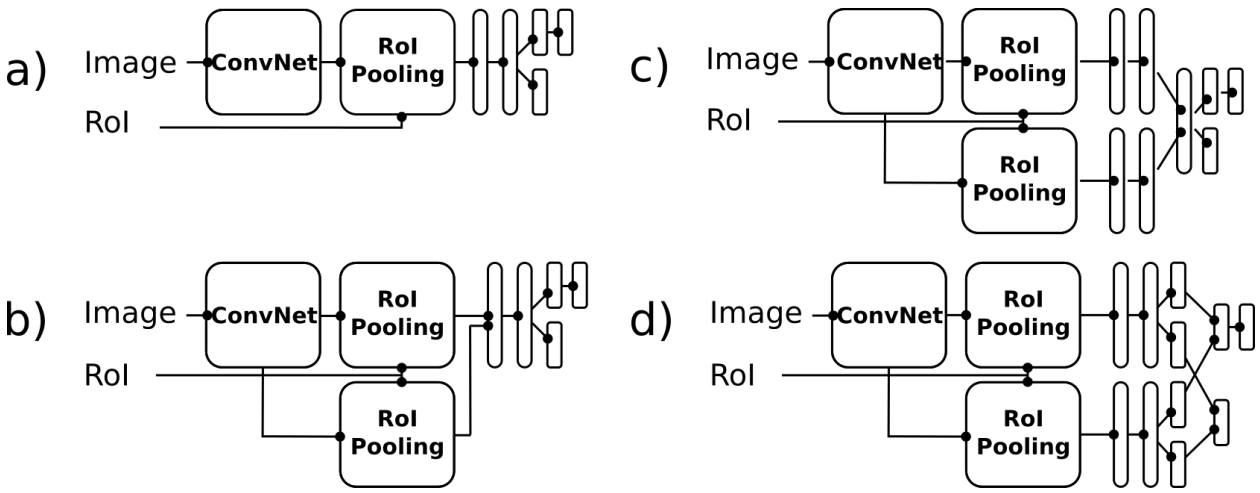

Figure 4.3: Illustration of the various architectures analyzed. Four different configurations were tested: a) the vanilla Fast R-CNN [Girshick, 2015]; b) a network with two RoI pooling layers which combine features from two different layers into a single fully-connected classifier; c) a network with two RoI pooling layers each feeding into a separate network of fully-connected layers, later joined into another network of fully-connected layers with two outputs; and d) two separate RoI pooling layers each feeding separate networks with two outputs. These are then combined into two final outputs, one for classification and the other for regression.

4.4.3 Framework Analysis

We analyzed two key components of the network design: i) which architecture provides the best framework to combine multiple feature maps (Sec. 4.4.3.1), and ii) which feature map combinations provide the best increase in accuracy (Sec. 4.4.3.2). Finally, we illustrate detection results on several test images and present results on the Caltech test set (Sec. 4.4.3.3).

4.4.3.1 Architecture

We tested three different architectures which all combine multiple feature maps and compared them with the vanilla Fast R-CNN architecture. Figure 4.3 shows the different architectures analyzed: a) the vanilla Fast R-CNN [Girshick, 2015] model using a VGG16 [Simonyan and Zisserman, 2015] network for feature extraction; b) a network in which two feature maps from different stages of the VGG16 model are each pooled by a separate RoI pooling layer, and then fed into a single fully-connected layer; c) a network in which two convolutional maps and RoI pooling layers each feed into separate, fully-connected networks, and then both are combined into a final fully-connected network with two output layers; and

Network architecture	miss rate
Fast R-CNN [Girshick, 2015]	18.86%
network b)	17.83%
network c)	41.16%
network d)	**17.40%**

Table 4.1: Performance comparison of the vanilla Fast R-CNN [Girshick, 2015] model and the three different architectures which apply feature combinations.

d) the same as c) but with two separate networks each with two outputs (one for classification and another for bounding-box regression) and then the outputs are separately fed into two final, fully-connected layers, one only for classification and the other only for box regression. All fully-connected layers, with the exception of the final output layers, have 4096 hidden units. Also, the ReLU nonlinearity is applied in each layer except for the output layers, and dropout with 50% chance is used as well. All four networks were trained and evaluated using the same parameters as described in Sec. 4.4.2.

Table 4.1 shows the miss rates of the different architectures. Relative to the vanilla Fast R-CNN, the use of multi-stage features improves miss rate by 1.03% and 1.46% when using architectures b) and d), respectively. Architecture c) showed a performance loss of about 22%. We may conclude that the fourth architecture provided the best way to combine several feature maps.

4.4.3.2 Feature maps

Several combinations of different feature maps from different stages of the CNN (architecture d) were tested in order to determine the best combination. We compared five different feature pairs and the use of only the feature map from the last convolutional layer; see Table 4.2. We only considered the last 6 convolutional layers of the VGG16 model (layers 8 to 13) mainly because all these layers have many complex kernels (512) of size 3×3 with a stride of 1. We point out that layers 8 to 10 have feature maps which are bigger than feature maps from layers 11 to 13. This is due to a max-pooling layer between layers 10 and 11. It is also important to note that pairs which include one earlier layer provide better results. The best result was obtained by combining layers 13 and 10 (Table 4.2). The reason for the use of an early layer is the bigger size of the feature map, with more features available for smaller detections than in later layers. Early layers may contain finer detail information that can be helpful to better distinguish pedestrians from background.

VGG16 Conv. Layers	miss rate
L13	18.86%
L13+L12	18.45%
L13+L11	17.67%
L13+L10	**17.40%**
L13+L9	17.49%
L13+L8	17.48%

Table 4.2: Miss rates of architecture d) using different combinations of feature maps.

4.4.3.3 Detection Results

Detection results of the MSF Fast R-CNN model are shown in Fig. 4.4. For comparison, the results of the MSF Fast R-CNN model (fourth column) are shown next to ground-truth annotations (first column), the LDCF [Nam et al., 2014] method (second column) and vanilla Fast R-CNN [Girshick, 2015] (third column). Red rectangles correspond to annotated ground-truth bounding boxes and green ones are detection results. Figure 4.4 shows that the MSF Fast R-CNN detector produces less false positives compared to the LDCF and the vanilla Fast R-CNN detectors.

4.4.4 State-or-the-art comparison

Benchmark results on the Caltech test set are reported in Fig. 4.5. We compared the result of our method with 14 other top-performing methods, including VJ [Viola and Jones, 2004], HOG [Dalal and Triggs, 2005], SCF+AlexNet [Hosang et al., 2015b], Katamari [Benenson et al., 2015], SpatialPooling+ [Paisitkriangkrai et al., 2014], SCCPriors [Yang et al., 2015b], TA-CNN [Tian et al., 2015b], CCF [Yang et al., 2015a], Checkerboards [Zhang et al., 2015b], CCF+CF [Yang et al., 2015a], Checkerboards+ [Zhang et al., 2015b], CompACT-Deep [Cai et al., 2015], DeepParts [Tian et al., 2015a], and SA-FastRCNN [Li et al., 2015]. Also, we used the public available toolbox for benchmarking our method provided by Dollár et al. [2012] which uses an evaluation metric with a log-average miss rate at 10^{-1} FPPI to summarize the detector's performance (lower is better). Our method performance is ranked at the 6th position of the top 10 performing algorithms, with an overall miss rate of 17.40%. Many of the methods used in this benchmark are based on boosted trees of simple hand-engineered features like HOG, LBP, LUV or orientation gradients and use a sliding window approach for detection (VJ, HOG, Katamari, SpatialPooling+, SCCPriors, Checkerboards,

Figure 4.4: Comparison of detection results with other state-of-the-art methods. The first column shows input test images with annotated pedestrians in red ground-truth boxes. The remaining columns show detections (green rectangles) of the following methods: LDCF [Nam et al., 2014] (second column), LDCF [Nam et al., 2014] + vanilla Fast R-CNN [Girshick, 2015] (third column) and LDCF [Nam et al., 2014] + MSF Fast R-CNN (fourth column).

Checkerboards+). Other methods replace hand-engineered features with learned features from pre-trained CNNs (SCF+AlexNet, TA-CNN, CCF, CompACT-Deep, DeepParts), and

the CFF+CF combines both hand-engineered with learned features for an increase in performance. Finally, our method and the SA-FastRCNN use learned features from a pre-trained CNN to classify region proposals. Besides the Checkerboards+ method, all top-5 methods used learned features from a CNN. The best method scored 9.68% miss rate and employs a modified Fast R-CNN framework using a VGG16 CNN as a feature extractor and region proposals generated by the ACF detector.

It is important to note that the significant gap between our method and the top-performing method SA-FastRCNN (which uses the same framework) might be related to implementation issues and inefficiencies of our method implemented in the Torch7 [Collobert et al., 2011a] platform compared with the original model in the Caffe [Jia et al., 2014] platform.

4.5 Conclusions

In this chapter we presented a method for pedestrian detection which is based on deep neural networks with multi-stage feature combination. The proposed method employs multiple convolutional features from different processing layers. This results in an increased detection performance without many extra computations. We demonstrated that combining features from an early stage with those from a later one makes it easier to distinguish pedestrians from background. Also, this combination of global features with finer details performs best when they are fed into a couple of networks that are combined in a final stage of the model. The improvements introduced can lead to better detections in surveillance, security and telecare systems.

In future work we expect to benchmark the current framework with the most popular CNNs like GoogleNet [Szegedy et al., 2014a] or ResNet [He et al., 2015], and investigate more advanced region proposal generators besides ACF and LDCF.

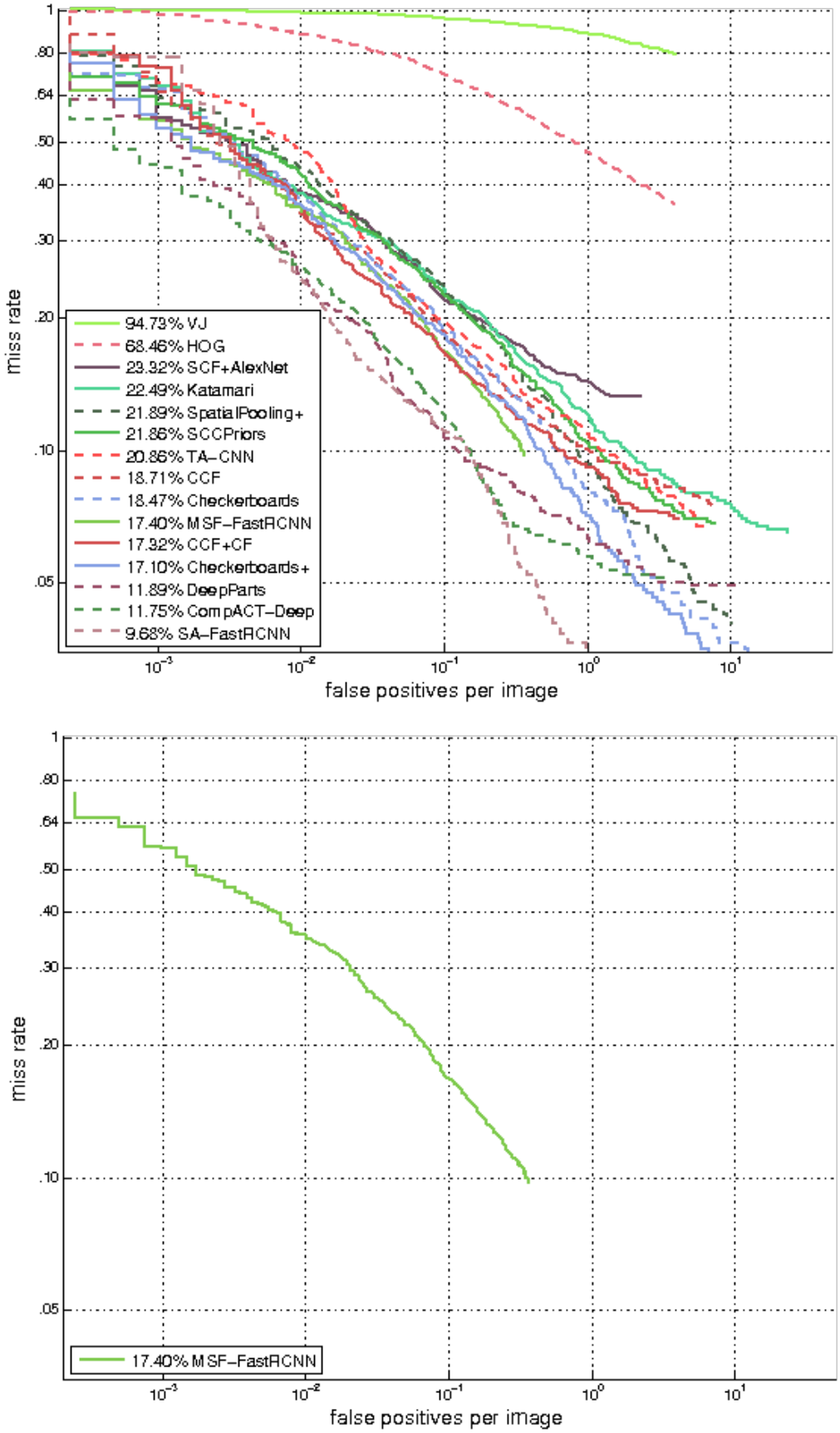

Figure 4.5: Performance comparison of our method with other state-of-the-art methods on the Caltech dataset. Our model shows competitive results, placed among the top-10 best performing algorithms with a 17.40% miss rate (lower is better).

Chapter 5

Human Joint Position Estimation

Pose estimation is the task of discovering the pose of an object in an image or in a sequence of images. Here, we focus on articulated human pose estimation in scenes with a single person. We employ a series of residual auto-encoders to produce multiple predictions which are then combined to provide a heatmap of body joints. In this network topology, features are processed across all scales which capture the various spatial relationships associated with the body. Repeated bottom-up and top-down processing with intermediate supervision of each auto-encoder network is applied. We propose some improvements of this type of regression-based networks to further increase performance, namely (a) increase the number of parameters of the auto-encoder networks in the pipeline, (b) use stronger regularization along with heavy data augmentation, (c) use sub-pixel precision for more precise joint localization, and (d) combine all auto-encoder output heatmaps into a single prediction. We demonstrate state-of-the-art results on the popular FLIC and LSP datasets.

Keywords: Human pose, ConvNet, Neural Networks, Auto-encoders

5.1 Introduction

Human pose estimation has substantially progressed on many popular benchmarks [Andriluka et al., 2014; Ess et al., 2008], including single person pose estimation [Chen and Yuille, 2014; Newell et al., 2016; Pishchulin et al., 2013b; Sapp and Taskar, 2013; Tompson et al., 2015; Wei et al., 2016]. For a pose estimation system to be effective it must be robust to deformation and occlusion, be invariant to changes in appearance due to factors like clothing and lighting, and yet be sufficiently accurate on rare and novel poses. Early work on pose estimation tackled these difficulties by using robust image features and sophisticated

structured prediction [Sapp and Taskar, 2013]. Deep learning methods [Pishchulin et al., 2013b] have replaced the conventional pipeline by convolutional neural networks (ConvNets) which constitute the main driver behind the huge leap in performance of many computer vision tasks. Pose estimation systems [Chen and Yuille, 2014; Newell et al., 2016; Tompson et al., 2015; Wei et al., 2016] adopted ConvNets as their main building block, often replacing hand-crafted features and graphical models.

In this chapter we employ a regression-based ConvNet, adding improvements for single-person pose estimation methods based on 2D heatmaps of body joints with intermediate supervision. Similar to Newell et al. [2016], we use stacks of deep residual auto-encoder networks connected end-to-end and trained jointly in a pipeline which iteratively refines the final prediction of the model. This topology allows for repeated bottom-up and top-down inferences across scales, which in conjunction with the use of intermediate supervision yields performance improvements. An auto-encoder is a feed-forward, non-recurrent convolutional neural network that aims to learn a representation (encoding) for a set of data, typically for the purpose of dimensionality reduction and input reconstruction. A residual auto-encoder is a variation of the basic auto-encoder network but aims to learn a residual mapping instead of the original underlying mapping of the image structure. This is achieved by using residual blocks instead of common convolutional layers. Intermediate supervision addresses vanishing gradients of the network by regenerating them throughout the backpropagation process. It consists of minimizing a combined loss of all auto-encoders' outputs of the pipeline. This helps to train large models which tend to suffer from vanishing gradients due to the size of the architectures. Furthermore, by combining all predictions from the auto-encoder networks in the pipeline with a weighted sum, we further increase the overall accuracy.

The main contributions of this chapter are (a) the increase of the number of parameters of the auto-encoder networks in the pipeline, (b) the use of stronger regularization along with heavy data augmentation in order to increase the robustness of the network, (c) sub-pixel precision for more precise body joint localization, and (d) the combination of multiple predictions obtained from the network's auto-encoders at all stages into a single weighted heatmap prediction. The last step provides additional accuracy at negligible cost. We demonstrate state-of-the-art results on standard benchmarks, i.e., the FLIC [Sapp and Taskar, 2013] and LSP [Johnson and Everingham, 2011] datasets.

The chapter is organized as follows: in Section 5.2 an overview of the state-of-the-art

literature in human pose estimation is presented. In Section 5.3 the model's architecture is described in detail. Section 5.4 deals with the implementation and optimization of the model, and benchmark results on the two popular pose estimation datasets are shown. Finally, in Section 5.5 some conclusions are provided.

5.2 Related Work

Early approaches to articulated pose estimation were pictorial structure models [Pishchulin et al., 2013a] in which spatial relations between parts of the body are expressed as a tree-structured graphical model with kinematic priors that link connected limbs. Other approaches like hierarchical models [Sun and Savarese, 2011] represent the relationships between parts at different scales and sizes in a hierarchical tree structure. Non-tree models [Dantone et al., 2013] refine predictions by introducing loops to augment the tree structure with additional edges that capture occlusion, symmetry and long-range relationships. In contrast, methods based on a sequential prediction framework [Ramakrishna et al., 2014] learn an implicit spatial model with complex interactions between variables by directly training an inference procedure.

Recently, there has been an increased interest in ConvNet models [Pishchulin et al., 2015; Tompson et al., 2014, 2015] which can be categorized as detection-based [Chen and Yuille, 2014; Insafutdinov et al., 2016; Pishchulin et al., 2015; Tompson et al., 2014, 2015] and regression-based [Toshev and Szegedy, 2014; Wei et al., 2016]. Detection-based methods rely on ConvNets as part detectors that are later combined with graphical models [Chen and Yuille, 2014; Tompson et al., 2014], which require hand-designed energy functions or heuristic initialization of spatial probability priors to remove outliers on the regressed confidence maps. Some of these methods also employ a dedicated network for precision refinement [Pishchulin et al., 2015; Tompson et al., 2015]. Regression-based models aim to minimize an energy function directly by using regression of a confidence map. The method by Bulat and Tzimiropoulos [2016] uses a cascade-based ConvNet for a two-step detection. A detection-based approach is used to convey confidence maps, which are then processed in a following network using an optimized regression-based approach. A recent development of regression-based methods has been the replacement of the standard L2 loss between body part predictions and ground truth locations by a confidence map regression. The L2 loss between predicted and ground truth confidence maps is encoded as 2D Gaussian blobs which

are centered at the part locations [Tompson et al., 2014]. Such maps are also called heatmaps.

Very recently, residual learning [He et al., 2015] has been applied to articulated pose estimation [Insafutdinov et al., 2016; Newell et al., 2016]. Insafutdinov et al. used residual learning for part detection, whereas Newell et al. applied stacked hourglass networks. The latter extended residual learning to fully convolutional [Long et al., 2015] and deconvolutional [Newell et al., 2016] networks, allowing for a more sophisticated top-down processing. Here, we further explore residual learning and stacked auto-encoders as in Newell et al. [2016]. We use auto-encoders with progressively more features in layers as the spatial receptive field increases, heavy data augmentation with stronger regularization in order to increase the model's generalization to novel poses, and exploit the predictive capabilities of the stacked network's pipeline by combining all inference heatmaps of body parts from all stages (i.e. from all auto-encoder outputs) to compose the final prediction. This final step takes advantage of the ability of the network to provide multiple predictions with increasingly higher fidelity of body part locations at each stage of the pipeline. Hence, more information can be used when producing the final prediction, thus increasing the overall accuracy.

5.3 Methods and Results

The articulated pose estimation scheme works as follows: (i) the model takes as input an image with a centered person and outputs a heatmap of all body joints; then (ii) the final prediction of the network consists of extracting the maximally activated locations of the heatmap for any given joint. In the following section we will describe the network's architecture.

5.3.1 Model Architecture

The architecture (Fig. 5.1, top row) is based on a deep ConvNet composed of multiple auto-encoders stacked together end-to-end, feeding the output of each into the next (Fig. 5.2, c)). This provides the network with a mechanism for repeated bottom-up and top-down inference, producing a refinement of the initial estimate and features across the stacks. We also use intermediate supervision to refine the heatmaps produced at each stage of the network. By doing so, the problem of vanishing gradients is addressed by replenishing them at each stage of the network during training. This results in faster convergence and ultimately in better

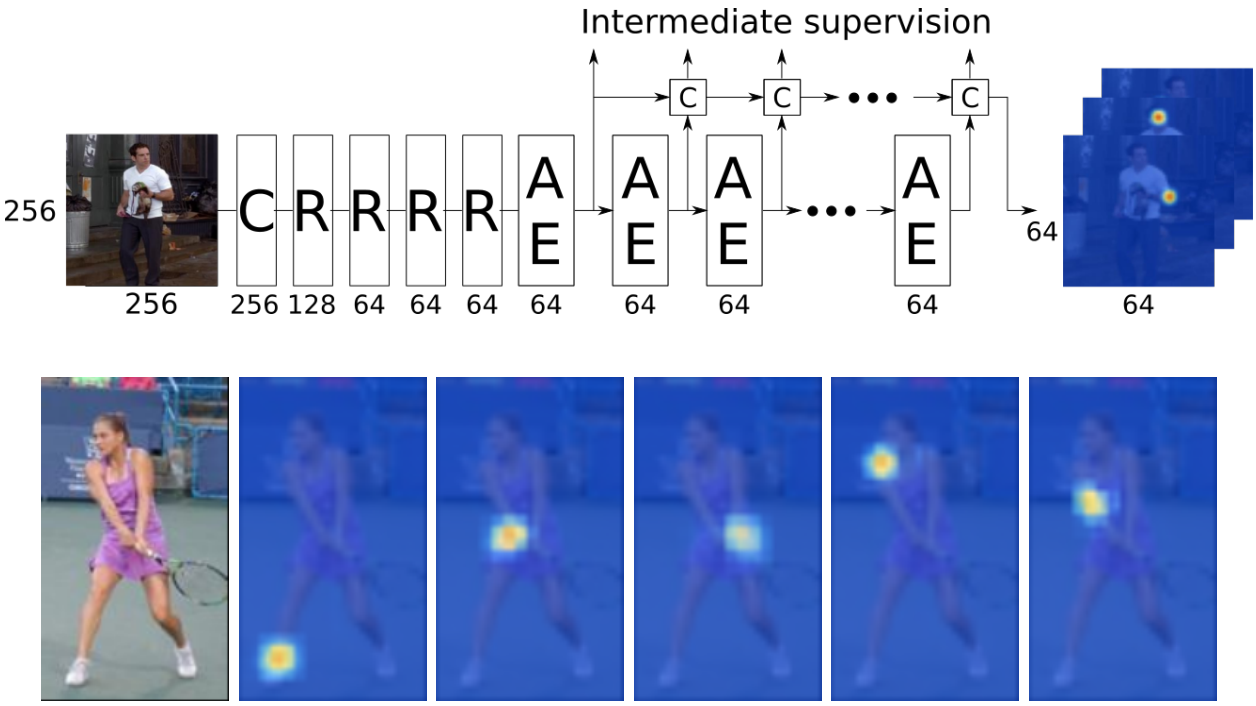

Figure 5.1: Top: Network architecture, where C represents a 7×7 convolution, R a residual block, AE an auto-encoder network, c a 1×1 convolution, and 256, 128 and 64 indicate the resolution (in pixels) of a layer/block in a stage of the pipeline. Bottom: examples of results.

heatmap predictions. Also, after the model has been trained, we combine all responses from all auto-encoders in the network to form an ensemble of predictions to produce the final heatmap. This is achieved by combining all output heatmaps and feeding them into a sequence of two 1×1 convolutional layers, which maps a weighted sum of all heatmap predictions into a single one. A 1×1 convolutional layer works like a fully-connected neural network and it works on varying-sized inputs in comparison with the fully-connected network which requires a fixed-size input. It computes a weighted sum of all features in a 1×1 grid of the feature map and is often used for dimensionality reduction of in fully convolutional networks.

An ensemble of several networks is often used to improve the overall performance of a method. We also take advantage of the multiple predictions that the networks provide. In addition, the final accuracy of the model is slightly increased without additional cost in processing time, since we can train one network once and employ this multiple times. Although the predictions are similar, their combination still improves the final score.

In Fig. 5.1, the architecture of the our model is shown. The model is a fully convolutional network which applies a 7×7 convolution (C) to the input pixels, followed by max-pooling

for resolution reduction and a series of four residual blocks (R) [He et al., 2015] to increase the feature dimensionality before the auto-encoder (AE) networks [Newell et al., 2016]. Each auto-encoder output is combined with the following one's output by a 1×1 convolutional layer (c) in order to produce a prediction. The auto-encoder networks used are composed of a sequence of residual blocks followed by max-pooling and up-sampling layers to produce a heatmap (Fig. 5.2, a)). Their architecture is similar to the hourglass networks as used in Newell et al. [2016], where consecutive residual blocks and max-pooling layers process and reduce the feature map to a low resolution (a minimum of 4×4 pixels), and then up-sampling layers with bicubic interpolation and shortcut connections to previous layers before max-pooling, restoring the feature map resolution to a size of 64×64 pixels. A series of 1×1 convolutions reduce the feature dimensionality to match the number of body joints to be detected when producing the output heatmap.

The differences between our auto-encoders and the hourglass network of Newell et al. [2016] are the following: (1) we use residual blocks with increasingly more filter kernels. than the previous block as the feature map resolution decreases, and (2) an auto-encoder's output heatmap is a combination of the current output produced by the network with the previous auto-encoder's output by using a 1×1 convolution. First, by increasing the auto-encoders total number of parameters we effectively increase the overall network's prediction capability at the cost of a larger footprint in memory and a moderate increase in processing time. Second, by combining the current prediction of an auto-encoder with the previous one in the pipeline, we further increase the network's overall performance with a negligible increase in training time. Our empirical tests showed that this combination provides a boost in accuracy with a small cost associated with the training time. These modifications significantly improve the overall accuracy of the network, justifying the increase in memory usage and processing time.

We apply residual bottleneck blocks [He et al., 2015] throughout the network (Fig. 5.2, b)). These blocks are composed of a sequence of convolutions with a maximum filter size of 3×3 with bottlenecking, combined with a shortcut connection (for more information see He et al. [2015]). Bottlenecking consists of reducing the dimensionality of an input via a 1×1 convolution before performing some costly function (here a 3×3 convolution). Then, its dimensionality is regenerated back to the original size (again, using a 1×1 convolution); see Fig. 5.2 b). This scheme helps to reduce the total memory usage and processing time of the

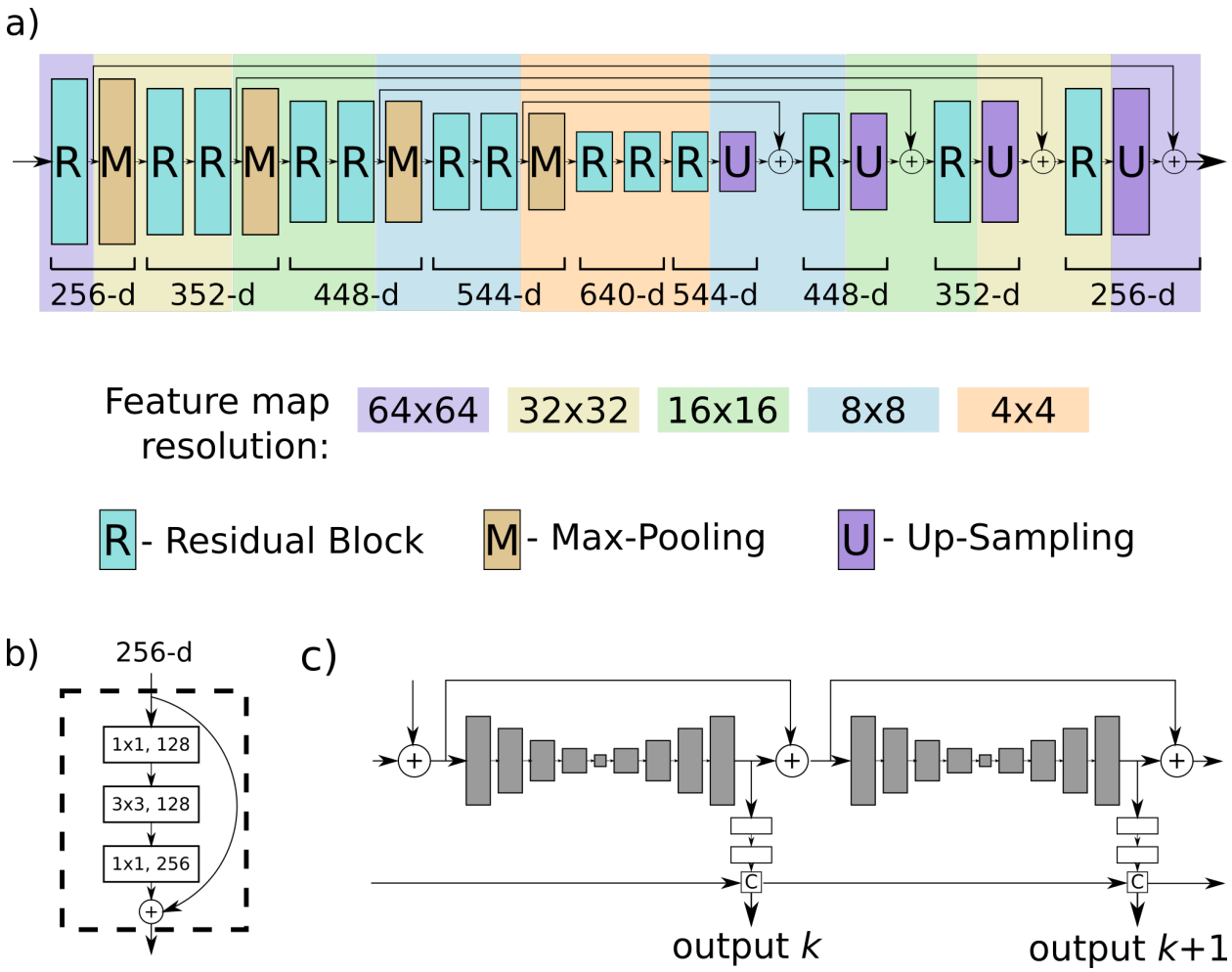

Figure 5.2: Ilustrations of the residual auto-encoder network (a), residual block (b) and the connection between auto-encoders (c). In a), the residual auto-encoder network's residual blocks, max-pooling and bilinear interpolation up-sampling layers are represented by R, M and U, respectively. For easier understanding, the feature map pixel resolution is coded by the size of the blocks and by a color scheme where the background color indicates the feature map resolution size: purple is 64×64, yellow is 32×32, green is 16×16, blue is 8×8 and salmon color is 4×4. Also, for each component, their feature dimension is indicated by a number and a bracket, for example, 256-d refers to a feature map with a dimension of size 256. Figure b) shows the composition of a residual block used throughout our network. In c), the connection between auto-encoder networks is shown. An auto-encoder network is illustrated by a sequence of grey rectangles and its output by a sequence of white rectangles which relate to 1×1 convolutions followed by the combination of the previous output and the current one defined by the square c.

network, while maintaining strong feature representations and accuracy.

5.4 Implementation, Tests and Results

In the following sections we provide implementation and optimization details of the model and show benchmark results on two popular pose estimation datasets.

5.4.1 Implementation Details

We used RGB images with 256×256 pixels as input for the network (Fig. 5.1, top row; see Sec. 5.3.1). To have normalized input samples, we centered the cropping region around the persons and only applied zero padding and resizing. This is different depending on the dataset used; see Sec. 5.4.2. In case of the Frames Labeled In Cinema [Sapp and Taskar, 2013] dataset, samples were cropped by centering on the center of the torso bounding box annotation. In case of the Leeds Sports Pose [Johnson and Everingham, 2011] dataset, we used the person-centric annotations to obtain the center coordinates of the person: by determining the minimum and maximum coordinate limits of all body joint annotations, and then computing the center coordinate. In case of the MPII Human Pose [Andriluka et al., 2014] dataset, annotations of the center of the person and size were used for cropping and scaling. During training on all sets, we applied data augmentation by image rotation [-40°, 40°], scaling [0.7, 1.3], horizontal flipping with 50% chance, and color transformations by varying the image brightness, contrast and saturation by up to 40%.

The model architecture is composed of a stack of eight auto-encoders with residual blocks, and two 1×1 convolutional layers as a final regression network for combining all the outputs of the auto-encoders in the stack. The model starts with a convolutional layer with a 7×7 kernel and stride of 2 in x and y, followed by a residual block, a max-pooling layer with stride 2 in x and y, and three more residual blocks in order to reduce the resolution from 256 to 64 pixels. Then, the last residual block is connected to the first auto-encoder network. The eight auto-encoders are connected end-to-end in the network. Finally, all output heatmaps of the auto-encoders are fed into a sequence of two 1×1 convolutional layers, one with 512 feature maps and the other with the number of body joints needed for a specific dataset. The 1×1 layers are trained separately after the model has been fully trained. All auto-encoder's residual blocks have between 256 and 640 filter kernels per residual block, where blocks with higher resolution (bigger feature maps) have a total of 256 kernels. The kernel count increases by a factor of 96 each time the feature map's resolution decreases, ending

with residual blocks with 640 filter kernels at the lowest resolution.

The networks were trained on a 6-core Intel i7-4790K CPU, 32GB ram, two NVIDIA GeForce GTX TITAN Black GPUs with 6GB of memory each, using the Torch7 library [Collobert et al., 2011a] and the Adam optimization method [Kingma and Ba, 2014] for 50 epochs with a learning rate of $2.5 \cdot 10^{-4}$, $\alpha = 0.99$ and $\epsilon = 10^{-8}$. Then the learning rate was reduced two times, to 10^{-4} for 15 epochs and to $5 \cdot 10^{-5}$ for another 10 epochs. After the network was trained, we trained the final two 1×1 convolutional layers for combining the heatmap predictions of all the auto-encoders, using again Adam for an additional 15 epochs with a learning rate of 10^{-3}, $\alpha = 0.99$ and $\epsilon = 10^{-8}$, reducing the learning rate two additional times to 10^{-4} for 5 epochs and to $5 \cdot 10^{-5}$ for 5 more epochs. All network weights were randomly initialized with a uniform distribution on [-0.01, 0.01]. We used mini-batches of 4 randomly sampled persons, batch normalization [Ioffe and Szegedy, 2015], randomized rectified linear unit (RReLU) [Xu et al., 2015] non-linearities and spatial dropout with 20% probability prior to all convolutions with filter sizes of 1×1 and 3×3, with the exception of bottlenecking convolutions in the residual block.

During back-propagation we used intermediate supervision [Tompson et al., 2014], where an L2 loss is applied for comparing the predicted heatmaps of all auto-encoders outputs to a ground-truth heatmap. Heatmaps consisted of 2D Gaussians centered on the joint locations with a standard deviation (size) of 1 pixel. Additionally, when predicting a body joint's coordinates, we refined the position localization by fitting a 1D parabola over the neighborhood of the peak's (x, y) coordinates by 1 pixel on the x and y axis separately, obtaining sub-pixel precision before resizing the coordinates to the original scale; see Fig. 5.2 (bottom). The entire training took about 24 hours on the two GPUs according to the scheme previously explained for the FLIC dataset and around 100 hours for the LSP dataset. After training, an image of 256×256 pixels took on average 0.32 seconds to infer all body joints of a person.

5.4.2 Datasets and Results

The proposed method was trained and evaluated on two popular datasets for single person-pose prediction: Leeds Sports Pose (LSP) [Johnson and Everingham, 2011] and Frames Labeled In Cinema (FLIC) [Sapp and Taskar, 2013]. These datasets were applied under the same conditions as detailed in Sec. 5.4.1. Evaluation was done using the standard

Methods	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
A	95.2	89.0	81.5	77.0	83.7	87.0	82.8	85.2
B	96.8	89.0	82.7	79.1	90.9	86.0	82.5	86.7
C	97.0	91.0	83.8	78.1	91.0	86.7	82.0	87.1
D	97.4	92.7	87.5	84.4	91.5	89.9	87.2	90.1
E	**97.8**	92.5	87.0	83.9	91.5	90.8	89.9	90.5
F	97.2	92.1	88.1	85.2	**92.2**	91.4	88.7	90.7
Ours	97.7	**93.0**	**88.9**	**85.5**	91.5	**92.0**	**92.1**	**91.5**

Table 5.1: Performance comparison on the LSP dataset (eval. protocol PCK@0.2), where method A belongs to Belagiannis and Zisserman [2016], B to Lifshitz et al. [2016] , C to Pishchulin et al. [2015], D to Insafutdinov et al. [2016], E to Wei et al. [2016] and F to Bulat and Tzimiropoulos [2016].

Methods	Elbow	Wrist	Total
Sapp and Taskar [2013]	72.5	54.5	63.5
Chen and Yuille [2014]	89.8	86.8	88.3
Wei et al. [2016]	92.5	90.0	91,3
Newell et al. [2016]	98.0	95.5	96.8
Ours	**98.3**	**96.0**	**97.2**

Table 5.2: Performance comparison on the FLIC dataset (eval. protocol PCK@0.2).

Percentage of Correct Keypoints (PCK) metric [Johnson and Everingham, 2010], which reports the percentage of detections that fall within a normalized distance of 20% of the torso size (PCK@0.2) of the ground truth joint positions.

We first evaluated our method on the LSP dataset, which consists of 10,000 images for training and 1,000 images for testing. We also used the MPII Human Pose dataset [Andriluka et al., 2014] to augment the number of training samples, as the other methods used for comparison in Table 5.1 did. This dataset's body joint annotation follows the same annotation scheme as the LSP dataset, and it contains around 28,000 annotated persons as training samples. We applied the person-centric (PC) annotations. Our model achieved top results on almost all body joints with an average PCK@0.2 of 91.5%; see Table 5.1. We also evaluated our method on the FLIC dataset, which consists of 3,987 images for training and 1,016 for testing. For this dataset, we report accuracy using the metric introduced by Sapp and Taskar [2013], who only used the elbow and wrist joints for benchmarking. Our method also shows state-of-the-art results, reaching 98.3% PCK@0.2 accuracy on the elbow and 96.0% on the wrist joints; see Table 5.2.

5.5 Conclusions

We presented a method for joint detection for articulated human joint position estimation using a series of residual auto-encoders. The proposed method employs a regression-based ConvNet composed of a series of deep residual auto-encoders connected end-to-end and trained jointly. The resulting model is then used as an ensemble of models by combining the responses of all individual auto-encoders along the pipeline, in order to convey the final prediction output. Heavy data augmentation and strong regularization were used because the network is a stack of auto-encoders with more parameters than previous state-of-the-art models [Newell et al., 2016].

We achieved top-performing results on the LSP dataset, with an average PCK of 91.5% for several body joints. On the FLIC dataset we achieved state-of-the-art results with a PCK of 98.3% for elbows and 96.0% for wrist joints.

In future work we will extend the model for the detection of joints and poses of multiple persons in a scene.

Chapter 6

Concluding remarks

This last chapter outlines the work done in the present thesis with the contributions and some guidelines for future research.

6.1 Summary

After a short introduction in Chapter 1, a brief overview of the themes of object recognition, human action recognition and deep learning was given in Chapter 2. It addressed the concept of object recognition and which cortical regions are involved in detection and representation. Also, an overview of the most relevant classic techniques for object recognition was provided. It addressed the perception of human activity in both human and computer vision, also briefly relating human action and body pose perception with cortical areas and processes involved in the brain. A brief description of several strategies for human activity recognition in computer vision was provided. For deep learning, an introduction to this sub-field of machine learning was provided, along with some examples of tasks in computer science where deep learning, and especially convolutional neural networks, has greatly progressed the state-of-the-art.

Chapter 3 presented initial work developed during the course of the thesis. This work addressed an application of biologically inspired vision processes to human-robot interaction. This involved using a single camera for detection of heads and hands. Also, by using optical flow information, a detected person could be tracked. Gestures were recognized by matching detected and annotated keypoints with those of templates in an internal database. This

allows a person to interact with a robot by giving it orders or feedback. The framework provided by this work was discontinued because it did not provide sufficiently good results for the tasks of pedestrian detection and human pose estimation. ConvNets provided a better framework with better results for these tasks.

Chapter 4 presented a method for pedestrian detection based on deep neural networks with multi-stage feature combination. The method employed and improved the Fast R-CNN framework, where multiple convolutional features from different processing stages were combined and used to increase performance of the detector. It was demonstrated that combining features from an earlier stage with those from a later one improved the accuracy in distinguishing pedestrians from background. Also, it was shown that combining global features with finer details helps to perform better when these features are fed into a couple of separate networks, which are later combined in a final stage of the model. Benchmark results on the popular Caltech dataset showed good results in comparison with other state-ot-the-art algorithms, ranking at the 6th position of the top 10 performing algorithms with an overall miss rate of 17.40%.

Chapter 5 presented a method for joint detection for articulated human pose estimation by using a series of residual auto-encoders. The proposed method employed a regression-based ConvNet composed of a series of deep residual auto-encoders which are connected end-to-end and which were trained jointly. The resulting model was then used as an ensemble of models by combining the responses of all individual auto-encoders along the pipeline, in order to convey the final prediction output. Heavy data augmentation and strong regularization were used because the network is a stack of auto-encoders with many more parameters than previous state-of-the-art models. The method achieved very good results on two popular datasets for human pose estimation. On the Leeds Sports Pose (LSP) dataset, it scored top-results with a PCK of 91.5%. On the Frames Labeled In Cinema (FLIC) dataset the proposed method achieved state-of-the-art results with a PCK@0.2 of 98.3% for the elbows and 96.0% for the wrist joints.

6.2 Contributions

This work provided several improvements which can be extended to other ConvNet-based methods/frameworks involved in human pedestrian detection and body pose estimation. These are summarized in this section into two topics concerning pedestrian detection and

human body pose estimation, and they are also the basis for future research that will be presented in the next section:

- **Pedestrian detection.** The main contribution was the integration of multiple features from different stages of a deep ConvNet to improve detection accuracy. While most detection methods do not take advantage of more information available in the ConvNet pipeline, here the usefulness of employing more features maps besides the last convolutional layer was investigated. Additionally, it was also investigated how several sources of features in a ConvNet pipeline can be combined and how their combination affects the final accuracy. Another contribution of this work was the analysis of the performance of several different types of architectures for using multiple feature maps. The knowledge provided by this work can be extended to general object detection, increasing the importance of the research developed in this thesis.

- **Human body pose estimation.** The main contributions proposed in this work can be summarized in four distinct areas: (1) With the increase of the number of parameters of the auto-encoder networks in the pipeline (by increasing the number of feature maps as the resolution decreased), the overall accuracy of the network increased, yielding a small trade-off in processing time and memory usage. (2) The use of stronger regularization along with heavy data augmentation provided an increase of the robustness of the network to over-training. (3) Sub-pixel precision for more precise body joint localization helped to produce better body pose predictions. (4) The combination of multiple predictions obtained from the network's auto-encoders at all stages into a single weighted heatmap prediction effectively provided additional accuracy to the performance at negligible cost. These improvements helped to increase the efficiency of the proposed ConvNet method, but all improvements can be used in other ConvNets and other applications.

6.3 Directions for further research

The work of this thesis yielded several achievements in both human person detection and body pose estimation tasks. Despite the good results in detecting persons and recognizing human body poses, further improvements can be introduced. Regarding person/pedestrian detection, it would be of interest to do more extensive testing on other datasets to assess

the performance of the proposed method. Also, it would be of interest to try other types of region proposal algorithms besides the ACF and LDCF methods, like the region proposal network used in Faster R-CNN [Ren et al., 2016] to produce high quality detections with a much faster processing time than the current ones. Another aspect to take into consideration in future work would be the use of other types of ConvNets for feature extraction like ResNet [He et al., 2015], GoogleNet [Szegedy et al., 2014a] or DenseNet [Huang et al., 2016], and test other architectures for object detection and classification that can be trained in an end-to-end fashion without a need for region proposals such as YOLO [Redmon et al., 2015] or SSD [Liu et al., 2015b]. Finally, it would be interesting to use additional information like optic flow and stereo disparity for object recognition. However, this can be tricky because very few available datasets provide video and/or stereo images.

Regarding human body pose estimation, there are also improvements that can be done. For example, residual blocks can be designed in other ways which may yield improvements. For example, using inception-like schemes, wider features, bigger kernels, etc. Auto-encoders can also be designed and combined in many different ways [Baldi, 2012], and there exists many different types of auto-encoders [Bengio et al., 2013; Makhzani et al., 2015]. Extensive testing on more challenging datasets would help to improve the current method. It would be also important to extend the current network detection scheme from single detections to multiple detections of body poses in an image.

Since many architectural choices in deep learning schemes take inspiration from many brain processes (like ConvNets), it would make sense to spend more time and attention to the brain with the purpose to both augment our knowledge of how our brain functions and how to take advantage of its processes and transfer them to the computer vision field. This does require a collective effort from both human and computer vision research, but the combined effort could definitely unlock new solutions and lead to a whole new way to address tasks in computer vision.

Bibliography

Adolphs, R., 2003. Cognitive neuroscience of human social behaviour. Nature Reviews Neuroscience 4 (3), 165–178.
URL http://www.nature.com/doifinder/10.1038/nrn1056

Ahmed, T., Ahmed, S., Ahmed, S., Motiwala, M., 2010. Real-time intruder detection in surveillance networks using adaptive kernel methods. IEEE International Conference on Communications, 1–5.
URL http://ieeexplore.ieee.org/document/5502592/

Ahonen, T., Hadid, A., Pietikäinen, M., 2006. Face description with local binary patterns: Application to face recognition. IEEE Pattern Analysis and Machine Intelligence 28 (12), 2037–2041.

Al-Absi, H., Abdullah, A., 2009. A proposed niologically inspired model for object recognition. IEEE Systems, Man, and Cybernetics, Part B 41 (6), 213–222.
URL http://link.springer.com/10.1007/978-3-642-05036-7_21

Al Ohali, Y., 2011. Computer vision based date fruit grading system: Design and implementation. Journal of King Saud University - Computer and Information Sciences 23 (1), 29–36.
URL http://www.sciencedirect.com/science/article/pii/S1319157810000054

Alahi, A., Ortiz, R., Vandergheynst, P., jun 2012. FREAK: fast retina keypoint. IEEE Computer Vision and Pattern Recognition, 510–517.
URL http://ieeexplore.ieee.org/document/6247715/

Alexe, B., Deselaers, T., Ferrari, V., 2012. Measuring the objectness of image windows. IEEE Pattern Analysis and Machine Intelligence 34 (11), 2189–2202.
URL http://ieeexplore.ieee.org/document/6133291/

Andrews, S., Huerta, I., Komura, T., Sigal, L., Mitchell, K., 2016. Real-time physics-based motion capture with sparse sensors. European Conference on Visual Media Production, 1–10.
URL http://cs.brown.edu/ ls/Publications/cvmp2016andrews.pdf

Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B., 2014. 2D human pose estimation: new benchmark and state of the art analysis. IEEE Computer Vision and Pattern Recognition, 3686–3693.
URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6909866

Angermueller, C., Pärnamaa, T., Parts, L., Stegle, O., 2016. Deep learning for computational biology. Molecular Systems Biology 12 (7), 878.
URL http://msb.embopress.org/lookup/doi/10.15252/msb.20156651

Arbib, M., 2005. From monkey-like action recognition to human language: an evolutionary framework for neurolinguistics. The Behavioral and Brain Sciences 28 (2), 105–124; discussion 125–167.

Archip, N., Clatz, O., Whalen, S., Kacher, D., Fedorov, A., Kot, A., Chrisochoides, N., Jolesz, F., Golby, A., Black, P., Warfield, S. K., 2007. Non-rigid alignment of pre-operative MRI, fMRI, and DT-MRI with intra-operative MRI for enhanced visualization and navigation in image-guided neurosurgery. NeuroImage 35 (2), 609–624.
URL http://linkinghub.elsevier.com/retrieve/pii/S1053811906011694

Avenanti, A., Bueti, D., Galati, G., Aglioti, S., 2005. Transcranial magnetic stimulation highlights the sensorimotor side of empathy for pain. Nature neuroscience 8 (7), 955–960.
URL http://www.ncbi.nlm.nih.gov/pubmed/15937484

Aziz-Zadeh, L., Wilson, S., Rizzolatti, G., Iacoboni, M., 2006. Congruent embodied representations for visually presented actions and linguistic phrases describing actions. Current Biology 16 (18), 1818–1823.
URL http://linkinghub.elsevier.com/retrieve/pii/S0960982206019683

Baldi, P., 2012. Autoencoders, unsupervised learning, and deep architectures. ICML Unsupervised and Transfer Learning 27, 37–50.

Bandera, J., Marfil, R., Bandera, A., Rodríguez, J., Molina-Tanco, L., Sandoval, F., 2009. Fast gesture recognition based on a two-level representation. Pattern Recognition Letters 30 (13), 1181–1189.

Bar, M., Kassam, K., Ghuman, A., Boshyan, J., Schmid, A., Dale, A., Hamalainen, M., Marinkovic, K., Schacter, D., Rosen, B., Halgren, E., 2006. Top-down facilitation of visual recognition. Proceedings of the National Academy of Sciences 103 (2), 449–454.

Batliner, A., Fischer, K., Huber, R., Spilker, J., Noth, E., 2000. Desperately seeking emotions or: actors, wizards, and human beings. ISCA Tutorial and Research Workshop (ITRW) on Speech and Emotion (September), 195–200.
URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.7104

Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., 2008. Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding 110 (3), 346–359.
URL http://linkinghub.elsevier.com/retrieve/pii/S1077314207001555

Belagiannis, V., Zisserman, A., 2016. Recurrent human pose estimation. arXiv preprint arXiv:1605.02914 (i), 1–16.
URL http://arxiv.org/abs/1605.02914

Bell, S., Zitnick, C., Bala, K., Girshick, R., 2015. Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks. arXiv preprint, 1–24.
URL http://arxiv.org/abs/1512.04143

Benenson, R., Mathias, M., Timofte, R., Van Gool, L., 2012. Pedestrian detection at 100 frames per second. IEEE Computer Vision and Pattern Recognition, 2903–2910.
URL http://ieeexplore.ieee.org/document/6248017/

Benenson, R., Omran, M., Hosang, J., Schiele, B., 2015. Ten years of pedestrian detection, what have we learned? Springer LNCS 8926, 613–627.

Bengio, Y., Lee, D., Bornschein, J., Lin, Z., 2015. Towards biologically plausible deep learning. arXiv preprint arxiv:1502.0415, 1–18.
URL http://arxiv.org/abs/1502.0415

Bengio, Y., Yao, L., Alain, G., Vincent, P., 2013. Generalized denoising auto-encoders as generative models. arXiv preprint arXiv:1305.6663, 1–9.
URL http://arxiv.org/abs/1305.6663

Bonini, L., Rozzi, S., Serventi, F., Simone, L., Ferrari, P., Fogassi, L., 2010. Ventral premotor and inferior parietal cortices make distinct contribution to action organization and intention understanding. Cerebral Cortex 20 (6), 1372–1385.
URL https://academic.oup.com/cercor/article-lookup/doi/10.1093/cercor/bhp200

Bordes, A., Chopra, S., Weston, J., 2014. Question answering with subgraph embeddings. arXiv preprint arXiv:1406.3676, 1–5.
URL http://arxiv.org/abs/1406.3676

Bottou, L., Bousquet, O., 2007. The tradeoffs of large scale learning. Neural Information Processing Systems 20, 161–168.
URL https://papers.nips.cc/paper/3323-the-tradeoffs-of-large-scale-learning.pdf

Bryson, A., 1961. A gradient method for optimizing multi-stage allocation processes. Proc. Harvard Univ. Symposium on Digital Computers and their Applications.

Bulat, A., Tzimiropoulos, G., 2016. Human pose estimation via convolutional part heatmap regression. arXiv preprint arXiv:1609.01743, 1–8.
URL http://arxiv.org/abs/1609.01743

Cai, Z., Saberian, M., Vasconcelos, N., 2015. Learning complexity-aware cascades for deep pedestrian detection. IEEE International Conference on Computer Vision, 3361–3369.
URL http://ieeexplore.ieee.org/document/7410741/

Calonder, M., Lepetit, V., Ozuysal, M., Trzcinski, T., Strecha, C., Fua, P., 2012. BRIEF: computing a local binary descriptor very fast. IEEE Pattern Analysis and Machine Intelligence 34 (7), 1281–1298.
URL http://ieeexplore.ieee.org/document/6081878/

Cappelli, R., Ferrara, M., Maltoni, D., 2010. Minutia cylinder-code: a new representation and matching technique for fingerprint recognition. IEEE Pattern Analysis and Machine Intelligence 32 (12), 2128–2141.
URL http://ieeexplore.ieee.org/document/5432197/

Chang, S., Chen, L., Chung, Y., Chen, S., 2004. Automatic license plate recognition. IEEE Intelligent Transportation Systems 5 (1), 42–53.
URL http://ieeexplore.ieee.org/document/1271288/

Chao-Yeh Chen, Grauman, K., 2012. Efficient activity detection with max-subgraph search. IEEE Computer Vision and Pattern Recognition, 1274–1281.
URL http://ieeexplore.ieee.org/document/6247811/

Chen, X., Yuille, A., 2014. Articulated pose estimation by a graphical model with image dependent pairwise relations. Advances in Neural Information Processing Systems, 1736–1744.
URL http://arxiv.org/abs/1407.3399

Chen, Y., Bedell, H., Frishman, L., 1998. The precision of velocity discrimination across spatial frequency. Perception & Psychophysics 60 (8), 1329–1336.
URL http://www.springerlink.com/index/10.3758/BF03207995

Cheng, H., Hsu, S., 2011. Intelligent highway traffic surveillance with self-diagnosis abilities. IEEE Intelligent Transportation Systems 12 (4), 1462–1472.
URL http://ieeexplore.ieee.org/document/6026251/

Cinzia, D., Vittorio, G., 2009. Neuroaesthetics: a review. Current Opinion in Neurobiology 19 (6), 682–687.
URL http://linkinghub.elsevier.com/retrieve/pii/S0959438809001238

Ciodaro, T., Deva, D., de Seixas, J., Damazio, D., 2012. Online particle detection with Neural Networks based on topological calorimetry information. Journal of Physics: Conference Series 368, 012030.

Ciresan, D., Meier, U., Masci, J., Schmidhuber, J., 2012. Multi-column deep neural network for traffic sign classification. Neural Networks 32, 333–338.
URL http://linkinghub.elsevier.com/retrieve/pii/S0893608012000524

Collobert, R., Kavukcuoglu, K., Farabet, C., 2011a. Torch7: A matlab-like environment for machine learning. Tech. rep.
URL http://publications.idiap.ch/downloads/papers/2011/Collobert_NIPSWORKSHOP_2011.pdf

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P., 2011b. Natural language processing (almost) from scratch. Journal of Machine Learning Research 12, 2493–2537.
URL http://arxiv.org/abs/1103.0398

Cook, R., Bird, G., Catmur, C., Press, C., Heyes, C., 2014. Mirror neurons: from origin to function. Behavioral and Brain Sciences 37 (02), 177–192.
URL http://www.ncbi.nlm.nih.gov/pubmed/24775147%5Cn

Corina, D., Knapp, H., 2006. Sign language processing and the mirror neuron system. Cortex 42 (4), 529–539.
URL http://linkinghub.elsevier.com/retrieve/pii/S0010945208703939

Csibra, G., 1993. Action mirroring and action understanding: an alternative account. In: Sensorimotor Foundations of Higher Cognition. Oxford University Press, pp. 435–459.

Cutzu, F., Edelman, S., aug 1998. Representation of object similarity in human vision: psychophysics and a computational model. Vision Research 38 (15-16), 2229–2257.
URL http://linkinghub.elsevier.com/retrieve/pii/S0042698997001867

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. IEEE Computer Vision and Pattern Recognition 1, 886–893.
URL http://ieeexplore.ieee.org/document/1467360/

Dantone, M., Gall, J., Leistner, C., Van Gool, L., 2013. Human Pose Estimation Using Body Parts Dependent Joint Regressors. IEEE Computer Vision and Pattern Recognition 36 (11), 3041–3048.
URL http://ieeexplore.ieee.org/document/6619235/

De Sousa, R., Rodrigues, J., Du Buf, J., 2010. Recognition of facial expressions by cortical multi-scale line and edge coding. Springer LNCS 6111 (PART 1), 415–424.

di Pellegrino, G., Fadiga, L., Fogassi, L., Gallese, V., Rizzolatti, G., 1992. Understanding motor events: a neurophysiological study. Experimental Brain Research 91 (1), 176–180.
URL http://link.springer.com/10.1007/BF00230027

DiCarlo, J., Zoccolan, D., Rust, N., 2012. How does the brain solve visual object recognition? Neuron 73 (3), 415–434.
URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3306444

Dolan, R., 2002. Emotion, cognition, and behavior. Science (New York, N.Y.) 298 (5596), 1191–1194.

Dollár, P., Appel, R., Belongie, S., Perona, P., 2014. Fast feature pyramids for object detection. IEEE Pattern Analysis and Machine Intelligence 36 (8), 1532–1545.
URL http://ieeexplore.ieee.org/document/6714453/

Dollár, P., Tu, Z., Perona, P., Belongie, S., 2009. Integral channel features. British Machine Vision Conference, 91.1–91.11.
URL http://www.bmva.org/bmvc/2009/Papers/Paper244/Paper244.html

Dollár, P., Wojek, C., Schiele, B., Perona, P., 2012. Pedestrian detection: an evaluation of the state of the art. IEEE Pattern Analysis and Machine Intelligence 34 (4), 743–761.
URL http://ieeexplore.ieee.org/document/5975165/

Downing, P., Jiang, Y., Shuman, M., Kanwisher, N., 2001. A cortical area selective for visual processing of the human body. Science (New York, N.Y.) 293 (5539), 2470–3.
URL http://www.ncbi.nlm.nih.gov/pubmed/11577239

Dreyfus, S., 1962. The numerical solution of variational problems. Journal of Mathematical Analysis and Applications 5 (1), 30–45.
URL http://linkinghub.elsevier.com/retrieve/pii/0022247X62900045

du Buf, J., 1993. Responses of simple cells: events, interferences, and ambiguities. Biological Cybernetics 68 (4), 321–333.
URL http://link.springer.com/10.1007/BF00201857

du Buf, J., Barroso, J., Rodrigues, J., Paredes, H., Farrajota, M., Fernandes, H., José, J., Teixiera, V., Saleiro, M., 2010. The SmartVision navigation prototype for the blind. In Proceedings for the International Conference on Software Development for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2010), 167–174.

Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research 12, 2121–2159.
URL http://jmlr.org/papers/v12/duchi11a.html

Enticott, P., Johnston, P., Herring, S., Hoy, K., Fitzgerald, P., 2008. Mirror neuron activation is associated with facial emotion processing. Neuropsychologia 46 (11), 2851–2854.
URL http://linkinghub.elsevier.com/retrieve/pii/S0028393208001905

Ess, A., Leibe, B., Schindler, K., Van Gool, L., 2008. A mobile vision system for robust multi-person tracking. IEEE Computer Vision and Pattern Recognition, 1–8.
URL http://ieeexplore.ieee.org/document/4587581/

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., Zisserman, A., 2010. The pascal visual object classes (VOC) challenge. International Journal of Computer Vision 88 (2), 303–338.

Farrajota, M., Rodrigues, J., du Buf, J., 2011. Optical flow by multi-scale annotated keypoints: a biological approach. Proc. Int. Conf. on Bio-inspired Systems and Signal Processing (BIOSIGNALS 2011), 307–315.

Farrajota, M., Rodrigues, J., du Buf, J., 2016a. A deep neural network video framework for monitoring elderly persons. In: Universal Access in Human-Computer Interaction. pp. 370–381.
URL http://link.springer.com/10.1007/978-3-319-40244-4_36

Farrajota, M., Rodrigues, J., du Buf, J., 2016b. Using Multi-Stage Features in Fast R-CNN for Pedestrian Detection. Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion.

Farrajota, M., Saleiro, M., Terzic, K., Rodrigues, J., du Buf, J., 2012. Multi-scale cortical keypoints for realtime hand tracking and gesture recognition. Proc. 1st International Workshop on Cognitive Assistive Systems: Closing the Action-Perception Loop, 9–15.
URL http://sapientia.ualg.pt/handle/10400.1/2105

Felleman, D., Van Essen, D., 1991. Distributed hierarchical processing in the primate cerebral cortex. Cerebral cortex (New York, N.Y. : 1991) 1 (1), 1–47.
URL http://cercor.oxfordjournals.org/cgi/doi/10.1093/cercor/1.1.1-a

Felzenszwalb, P., Girshick, R., Mcallester, D., Ramanan, D., 2009. Object detection with discriminatively trained part based models. IEEE Pattern Analysis and Machine Intelligence, 1–20.

Felzenszwalb, P., McAllester, D., Girshick, R., Ramanan, D., 2013. Visual object detection with deformable part models. Communications of the ACM 56 (9), 97.

Fogassi, L., 2005. Parietal lobe: from action organization to intention understanding. Science 308 (5722), 662–667.
URL http://www.sciencemag.org/cgi/doi/10.1126/science.1106138

Friston, K., Mattout, J., Kilner, J., 2011. Action understanding and active inference. Biological Cybernetics 104 (1-2), 137–160.
URL http://link.springer.com/10.1007/s00422-011-0424-z

Fukushima, K., Miyake, S., 1982. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. Pattern Recognition 15 (6), 455–469.
URL http://linkinghub.elsevier.com/retrieve/pii/0031320382900243

Gallese, V., Fadiga, L., Fogassi, L., Rizzolatti, G., 1996. Action recognition in the premotor cortex. Brain : A Journal of Neurology (2), 593–609.
URL http://www.ncbi.nlm.nih.gov/pubmed/8800951

Gallese, V., Sinigaglia, C., 2011. What is so special about embodied simulation? Trends in Cognitive Sciences 15 (11), 512–519.
URL http://linkinghub.elsevier.com/retrieve/pii/S136466131100194X

Garcia-Martin, A., Martinez, J. M., 2010. Robust real time moving people detection in surveillance scenarios. IEEE Advanced Video and Signal Based Surveillance, 241–247.
URL http://ieeexplore.ieee.org/document/5597118/

Gazzola, V., Keysers, C., 2009. The observation and execution of actions share motor and somatosensory voxels in all tested subjects: single-subject analyses of unsmoothed fMRI data. Cerebral Cortex 19 (6), 1239–1255.
URL https://academic.oup.com/cercor/article-lookup/doi/10.1093/cercor/bhn181

Gidaris, S., Komodakis, N., 2015. Object detection via a multi-region & semantic segmentation-aware CNN model. arXiv preprint arXiv:1505.01749, 1–29.
URL http://arxiv.org/abs/1505.01749

Giese, M., Poggio, T., 2003. Cognitive neuroscience: Neural mechanisms for the recognition of biological movements. Nature Reviews Neuroscience 4 (3), 179–192.
URL http://www.nature.com/doifinder/10.1038/nrn1057

Girshick, R., 2015. Fast R-CNN. arXiv preprint arXiv:1504.08083, 1–9.
URL http://arxiv.org/abs/1504.08083

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. IEEE Computer Vision and Pattern Recognition, 580–587.
URL http://ieeexplore.ieee.org/document/6909475/

Glenberg, A., Sato, M., Cattaneo, L., Riggio, L., Palumbo, D., Buccino, G., 2008. Processing abstract language modulates motor system activity. The Quarterly Journal of Experimental Psychology 61 (6), 905–919.
URL http://dx.doi.org/10.1080/17470210701625550

Glorot, X., Bordes, A., Bengio, Y., 2011. Domain adaptation for large-scale sentiment classification: a deep learning approach. International Conference on Machine Learning, 513–520.
URL http://www.icml-2011.org/papers/342_icmlpaper.pdf

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning. Book in preparation for MIT Press.
URL http://www.deeplearningbook.org

Graves, A., Jaitly, N., 2014. Towards end-to-end speech recognition with recurrent neural networks. International Conference on Machine Learning 32 (1), 1764–1772.
URL http://jmlr.org/proceedings/papers/v32/graves14.pdf

Gretzel, U., 2011. Intelligent systems in tourism. Annals of Tourism Research 38 (3), 757–779.
URL http://linkinghub.elsevier.com/retrieve/pii/S0160738311000776

Gridley, M., Hoff, R., 2006. Do mirror neurons explain misattribution of emotions in music? Perceptual and Motor Skills 102 (2), 600–602.
URL http://pms.sagepub.com/lookup/doi/10.2466/pms.102.2.600-602

Gros, B., Blake, R., Hiris, E., 1998. Anisotropies in visual motion perception: a fresh look. Journal of the Optical Society of America. A, Optics, image science, and vision 15 (8), 2003–2011.
URL http://www.ncbi.nlm.nih.gov/pubmed/9691484

Grossman, E., Donnelly, M., Price, R., Pickens, D., Morgan, V., Neighbor, G., Blake, R., 2000. Brain areas involved in perception of biological motion. Journal of cognitive neuroscience 12 (5), 711–20.
URL http://www.ncbi.nlm.nih.gov/pubmed/11054914

Gupta, A., Davis, L., 2007. Objects in action: an approach for combining action understanding and object perception. IEEE Computer Vision and Pattern Recognition, 1–8.
URL http://ieeexplore.ieee.org/document/4270329/

Haala, N., Kada, M., 2010. An update on automatic 3D building reconstruction. ISPRS Journal of Photogrammetry and Remote Sensing 65 (6), 570–580.
URL http://linkinghub.elsevier.com/retrieve/pii/S0924271610000894

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., Ng, A. Y., 2014. Deep speech: scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567, 1–12.
URL http://arxiv.org/abs/1412.5567

Hariharan, B., Malik, J., Ramanan, D., 2012. Discriminative decorrelation for clustering and classification. In: Springer LNCS. Vol. 7575 LNCS. pp. 459–472.
URL http://link.springer.com/10.1007/978-3-642-33765-9_33

He, K., Zhang, X., Ren, S., Sun, J., 2014. Spatial pyramid pooling in deep convolutional networks for visual recognition. arXiv preprint arXiv:1406.4729, 1–14.
URL http://arxiv.org/abs/1406.4729

He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385, 1–12.
URL http://arxiv.org/abs/1512.03385

Helmstaedter, M., Briggman, K., Turaga, S., Jain, V., Seung, H. S., Denk, W., 2013. Connectomic reconstruction of the inner plexiform layer in the mouse retina. Nature 500 (7461), 168–174.
URL http://www.ncbi.nlm.nih.gov/pubmed/23925239

Hickok, G., 2009. Eight problems for the mirror neuron theory of action understanding in monkeys and humans. Journal of Cognitive Neuroscience 21 (7), 1229–1243.
URL http://www.mitpressjournals.org/doi/10.1162/jocn.2009.21189

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., Kingsbury, B., 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. IEEE Signal Processing Magazine 29 (6), 82–97.
URL http://ieeexplore.ieee.org/document/6296526/

Hosang, J., Benenson, R., Dollár, P., Schiele, B., 2015a. What makes for effective detection proposals? arXiv preprint arXiv:1502.05082, 1–16.
URL http://arxiv.org/abs/1502.05082

Hosang, J., Omran, M., Benenson, R., Schiele, B., 2015b. Taking a deeper look at pedestrians. IEEE Computer Vision and Pattern Recognition 07-12-June, 4073–4082.

Huang, G., Liu, Z., Weinberger, K., van der Maaten, L., 2016. Densely connected convolutional networks. arXiv preprint arXiv:1608.06993, 1–12.
URL http://arxiv.org/abs/1608.06993

Huang, P., Urbana, N., He, X., Gao, J., Deng, L., Acero, A., Heck, L., 2013. Learning deep structured semantic models for web search using clickthrough data. ACM International Conference on Information and Knowledge Management, 2333–2338.
URL http://dl.acm.org/citation.cfm?id=2505665

Hubel, D., 1995. Eye, brain and vision. In: Scientific American Library series. New York.

Hubel, D., Wiesel, T., 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of Physiology 160 (1), 106–154.
URL http://doi.wiley.com/10.1113/jphysiol.1962.sp006837

Iacoboni, M., Molnar-Szakacs, I., Gallese, V., Buccino, G., Mazziotta, J., Rizzolatti, G., 2005. Grasping the intentions of others with one's own mirror neuron system. PLoS Biology 3 (3), e79.
URL http://dx.plos.org/10.1371/journal.pbio.0030079

Iacoboni, M., Woods, R., 1999. Cortical mechanisms of human imitation. Science (New York, N.Y.) 286, 2526.

Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., Schiele, B., 2016. DeeperCut: a deeper, stronger, and faster multi-person pose estimation model. arXiv preprint arXiv:1605.03170, 1–22.
URL http://arxiv.org/abs/1605.03170

Ioffe, S., Szegedy, C., 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 1–11.
URL http://arxiv.org/abs/1502.03167

Iosifidis, A., Tefas, A., Pitas, I., 2012. Activity-based person identification using fuzzy representation and discriminant learning. IEEE on Information Forensics and Security 7 (2), 530–542.
URL http://ieeexplore.ieee.org/document/6080731/

Jellema, T., Baker, C., Wicker, B., Perrett, D., 2000. Neural representation for the perception of the intentionality of actions. Brain and Cognition 44 (2), 280–302.
URL http://linkinghub.elsevier.com/retrieve/pii/S0278262600912314

Jia, J., 2009. A machine vision application for industrial assembly inspection. IEEE International Conference on Machine Vision, 172–176.
URL http://ieeexplore.ieee.org/document/5381107/

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., 2014. Caffe: convolutional architecture for fast feature embedding. Proceedings of the ACM International Conference on Multimedia, 675–678.
URL http://arxiv.org/abs/1408.5093

Johansson, G., 1973. Visual perception of biological motion and a model for its analysis. Attention, Perception, & Psychophysics 14 (2), 201–211.
URL http://www.springerlink.com/content/f07t232637745786

Johnson, S., Everingham, M., 2010. Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. British Machine Vision Conference 12, 1–11.
URL http://www.bmva.org/bmvc/2010/conference/paper12/index.html

Johnson, S., Everingham, M., 2011. Learning effective human pose estimation from inaccurate annotation. IEEE Computer Vision and Pattern Recognition, 1465–1472.
URL http://ieeexplore.ieee.org/document/5995318/

Kanwisher, N., 2000. Domain specificity in face perception. Nature Neuroscience 3 (8), 759–763.
URL http://www.ncbi.nlm.nih.gov/pubmed/10903567

Kelley, H., 1960. Gradient theory of optimal flight paths. ARS Journal 30, 947–954.

Kilner, J., Friston, K., Frith, C., 2007. Predictive coding: an account of the mirror neuron system. Cognitive Processing 8 (3), 159–166.
URL http://link.springer.com/10.1007/s10339-007-0170-2

Kim, H., Kurillo, G., Bajcsy, R., 2008. Hand tracking and motion detection from the sequence of stereo color image frames. IEEE International Conference on Industrial Technology, 1–6.
URL http://ieeexplore.ieee.org/document/4608702/

Kim, Y., 2014. Convolutional neural networks for sentence classification. Conference on Empirical Methods in Natural Language Processing, 1746–1751.
URL http://emnlp2014.org/papers/pdf/EMNLP2014181.pdf

Kingma, D., Ba, J., 2014. Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980, 1–13.
URL http://arxiv.org/abs/1412.6980

Krizhevsky, A., Sutskever, I., Hinton, G., 2012. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 1097–1105.
URL http://arxiv.org/abs/1102.0183

Krühn, S., Brass, M., 2008. Testing the connection of the mirror system and speech. Neuropsychologia 46 (5), 1513–1521.
URL http://linkinghub.elsevier.com/retrieve/pii/S0028393208000109

Lane, N., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., Campbell, A., 2010. A survey of mobile phone sensing. IEEE Communications Magazine 48 (9), 140–150.
URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5560598

Le Clec'H, G., Dehaene, S., Cohen, L., Mehler, J., Dupoux, E., Poline, J., Lehéricy, S., van de Moortele, P., Le Bihan, D., 2000. Distinct cortical areas for names of numbers and body parts independent of language and input modality. NeuroImage 12 (4), 381–391.
URL http://linkinghub.elsevier.com/retrieve/pii/S105381190090627X

Learned-Miller, E., Huang, G., RoyChowdhury, A., Li, H., Hua, G., 2016. Labeled faces in the wild: A survey. In: Advances in Face Detection and Facial Image Analysis. pp. 189–248.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.
URL http://dx.doi.org/10.1038/nature14539%5Cn10.1038/nature14539

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., Jackel, L., 1990. Handwritten digit recognition with a back-propagation network. Advances in Neural Information Processing Systems, 396–404.

Leeds, D., 2013. Searching for the visual components of object perception. Ph.D. thesis, Carnegie Mellon University Pittsburgh, PA.

Leutenegger, S., Chli, M., Siegwart, R. Y., 2011. BRISK: Binary Robust invariant scalable keypoints. IEEE International Conference on Computer Vision, 2548–2555.
URL http://ieeexplore.ieee.org/document/6126542/

Li, J., Liang, X., Shen, S., Xu, T., Feng, J., Yan, S., 2015. Scale-aware Fast R-CNN for Pedestrian Detection. arXiv preprint arXiv:1510.08160, 1–9.
URL http://arxiv.org/abs/1510.08160

Li, Y., 2012. Hand gesture recognition using Kinect. IEEE Computer Science and Automation Engineering, 196–199.
URL http://ieeexplore.ieee.org/document/6269439/

Li, Y., Sun, B., Wu, T., Wang, Y., 2016. Face detection with end-to-end integration of a convNet and a 3D model. arXiv preprint arXiv:1606.00850, 1–16.
URL http://arxiv.org/abs/1606.00850

Lifshitz, I., Fetaya, E., Ullman, S., 2016. Human pose estimation using deep consensus voting. arXiv preprint arXiv:1603.08212, 1–16.
URL http://arxiv.org/abs/1603.08212

Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L., 2014. Microsoft COCO: common objects in context. In: Springer LNCS. Vol. 8693 LNCS. pp. 740–755.
URL http://link.springer.com/10.1007/978-3-319-10602-1_48

Lindsey, D. T., 2000. Vision science: photons to phenomenology. Optometry and Vision Science 77 (5), 233–234.

Linnainmaa, S., 1970. The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors. Master's Thesis (in Finnish), Univ. Helsinki.

Liu, B., Fu, Y., Yao, Z., Xiong, H., 2013. Learning geographical preferences for point-of-interest recommendation. Sigkdd, 1043.
URL http://dl.acm.org/citation.cfm?id=2487575.2487673

Liu, J., Liu, Y., Zhang, G., Zhu, P., Chen, Y. Q., 2015a. Detecting and tracking people in real time with RGB-D camera. Pattern Recognition Letters 53, 16–23.
URL http://linkinghub.elsevier.com/retrieve/pii/S016786551400302X

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., 2015b. SSD: single shot multiBox detector. arXiv preprint arXiv:1512.02325, 1–15.
URL http://arxiv.org/abs/1512.02325

Livingstone, M., 2008. Vision and art: the biology of seeing. New York: Harry N. Abrams.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. IEEE Computer Vision and Pattern Recognition, 3431–3440.
URL http://ieeexplore.ieee.org/document/7298965/

Lopez de Avila, A., 2015. Smart destinations: XXI century tourism. Conference on information and communication technologies in Tourism.

Loula, F., Prasad, S., Harber, K., Shiffrar, M., 2005. Recognizing people from their movement. Journal of Experimental Psychology: Human Perception and Performance 31 (1), 210–220.
URL http://doi.apa.org/getdoi.cfm?doi=10.1037/0096-1523.31.1.210

Lowe, D., 1999. Object recognition from local scale-invariant features. IEEE International Conference on Computer Vision 2 (8), 1150–1157.
URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=790410

Lowe, D., 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60 (2), 91–110.
URL http://link.springer.com/10.1023/B:VISI.0000029664.99615.94

Lu, C., Tang, X., 2014. Surpassing human-level face verification performance on LFW with GaussianFace. arXiv preprint arXiv:1404.3840, 1–19.
URL http://arxiv.org/abs/1404.3840

Ma, J., Sheridan, R., Liaw, A., Dahl, G., Svetnik, V., 2015. Deep neural nets as a method for quantitative structure-activity relationships. Journal of Chemical Information and Modeling 55 (2), 263–274.
URL http://pubs.acs.org/doi/abs/10.1021/ci500747n

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B., 2015. Adversarial autoencoders. arXiv preprint arXiv:1511.05644, 1–10.
URL http://arxiv.org/abs/1511.05644

Marín-Jiménez, M., Muñoz-Salinas, R., Yeguas-Bolivar, E., Pérez de la Blanca, N., 2014. Human interaction categorization by using audio-visual cues. Machine Vision and Applications 25 (1), 71–84.
URL http://link.springer.com/10.1007/s00138-013-0521-1

Marreiros, F., 2016. Guidance and visualization for brain tumor surgery. Ph.D. thesis, Linköping University, Linköping, Sweden.
URL http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-130791

Martinez, H., Yannakakis, G., Hallam, J., 2014. Don't classify ratings of affect; rank them! IEEE Affective Computing 5 (3), 314–326.
URL http://ieeexplore.ieee.org/document/6883166/

McCallum, A., Nigam, K., Rennie, J., Seymore, K., 2000. Automating the construction of internet portals with machine learning. Information Retrieval 3 (2), 127–163.
URL http://link.springer.com/10.1023/A:1009953814988

McDonnell, M., Tissera, M., van Schaik, A., Tapson, J., 2015. Fast, simple and accurate handwritten digit classification using extreme learning machines with shaped input-weights. arXiv preprint arXiv:1412.8307, 1–13.
URL http://arxiv.org/abs/1412.8307

Meger, D., Forssén, P., Lai, K., Helmer, S., McCann, S., Southey, T., Baumann, M., Little, J., Lowe, D., 2008. Curious george: an attentive semantic robot. Robotics and Autonomous Systems 56 (6), 503–511.
URL http://linkinghub.elsevier.com/retrieve/pii/S0921889008000316

Meggle, G., 1977. Analytische handlungstheorie. Suhrkamp, Frankfurt.

Mei, X., Sun, X., Zhou, M., Jiao, S., Wang, H., Xiaopeng Zhang, 2011. On building an accurate stereo matching system on graphics hardware. IEEE International Conference on Computer Vision Workshops, 467–474.
URL http://ieeexplore.ieee.org/document/6130280/

Michels, L., Lappe, M., Vaina, L. M., 2005. Visual areas involved in the perception of human movement from dynamic form analysis. NeuroReport 16 (10), 1037–41.
URL http://www.ncbi.nlm.nih.gov/pubmed/15973144

Mikolajczyk, K., Schmid, C., Zisserman, A., 2004. Human detection based on a probabilistic assembly of robust part detectors. European Conference on Computer Vision. Springer 3021, 69–82.

Mikolov, T., Deoras, A., Povey, D., Burget, L., Cernocky, J., 2011. Strategies for training large scale neural network language models. IEEE Workshop on Automatic Speech Recognition & Understanding, 196–201.
URL http://ieeexplore.ieee.org/document/6163930/

Miller, I., Campbell, M., Huttenlocher, D., Nathan, A., Kline, F., Moran, P., Zych, N., Schimpf, B., Lupashin, S., Garcia, E., Catlin, J., Kurdziel, M., Fujishima, H., 2009. Team cornell's skynet: robust perception and planning in an urban environment. In: Springer Tracts in Advanced Robotics. Vol. 56. pp. 257–304.
URL http://link.springer.com/10.1007/978-3-642-03991-1_7

Minsky, M., 1961. Steps toward artificial intelligence. Proceedings of the IRE 49 (1), 8–30.

Molenberghs, P., Cunnington, R., Mattingley, J. B., 2012. Brain regions with mirror properties: A meta-analysis of 125 human fMRI studies. Neuroscience & Biobehavioral Reviews 36 (1), 341–349.
URL http://linkinghub.elsevier.com/retrieve/pii/S0149763411001394

Nam, W., Dollár, P., Han, J., 2014. Local decorrelation for improved detection. Advances in Neural Information Processing Systems, 1–9.
URL http://arxiv.org/abs/1406.1134

Nehaniv, C., Dautenhahn, K., Kubacki, J., Haegele, M., Parlitz, C., Alami, R., 2005. A methodological approach relating the classification of gesture to identification of human intent in the context of human-robot interaction. IEEE International Workshop on Robot and Human Interactive Communication, 371–377.
URL http://ieeexplore.ieee.org/document/1513807/

Neri, P., Morrone, M., Burr, D., 1998. Seeing biological motion. Nature 395 (6705), 894–6.
URL http://www.ncbi.nlm.nih.gov/pubmed/9804421

Newell, A., Yang, K., Deng, J., 2016. Stacked hourglass networks for human pose estimation. arXiv preprint arXiv:1603.06937, 1–17.
URL http://arxiv.org/abs/1603.06937

Ni, B., Moulin, P., Yang, X., Yan, S., 2015. Motion part regularization: improving action recognition via trajectory group selection. IEEE Computer Vision and Pattern Recognition, 3698–3706.
URL http://ieeexplore.ieee.org/document/7298993/

Nistér, D., Stewenius, H., 2006. Scalable recognition with a vocabulary tree. Computer Vision and Pattern Recognition 2, 2161–2168.
URL http://ieeexplore.ieee.org/document/1641018/

Novak, J., 1996. Automated apparatus and method for object recognition at checkout counters. Google Patents, US Patent 5,497,314.

Oliva, A., Torralba, A., 2006. Building the gist of a scene: the role of global image features in recognition. Progress in Brain Research, 23–36.
URL http://linkinghub.elsevier.com/retrieve/pii/S0079612306550022

Olshausen, B., Field, D., 2005. How close are we to understanding V1? Neural computation 17 (8), 1665–1699.
URL http://www.mitpressjournals.org/doi/abs/10.1162/0899766054026639#.VYCVBBNViko

Ondobaka, S., Bekkering, H., 2013. Conceptual and perceptuo-motor action control and action recognition. Cortex 49 (10), 2966–2967.
URL http://linkinghub.elsevier.com/retrieve/pii/S0010945213001639

Ouyang, W., Wang, X., 2012. A discriminative deep model for pedestrian detection with occlusion handling. IEEE Computer Vision and Pattern Recognition, 3258–3265.

Ouyang, W., Wang, X., 2013. Joint deep learning for pedestrian detection. IEEE International Conference on Computer Vision, 2056–2063.

Paisitkriangkrai, S., Shen, C., van den Hengel, A., 2014. Strengthening the effectiveness of pedestrian detection with spatially pooled features. In: European Conference on Computer Vision. pp. 546–561.
URL http://link.springer.com/10.1007/978-3-319-10593-2_36

Patron-Perez, A., Marszalek, M., Reid, I., Zisserman, A., dec 2012. Structured learning of human interactions in TV shows. IEEE Pattern Analysis and Machine Intelligence 34 (12), 2441–2453.
URL http://ieeexplore.ieee.org/document/6133287/

Peelen, M., Downing, P., 2005. Selectivity for the human body in the fusiform gyrus. Journal of neurophysiology 93 (1), 603–8.
URL http://www.ncbi.nlm.nih.gov/pubmed/15295012

Pinheiro, P., Collobert, R., Dollar, P., 2015. Learning to segment object candidates. arXiv preprint arXiv:1506.06204, 1–10.
URL http://arxiv.org/abs/1506.06204

Pinheiro, P., Lin, T., Collobert, R., Dollàr, P., 2016. Learning to refine object segments. arXiv preprint arXiv:1603.08695, 1–18.
URL http://arxiv.org/abs/1603.08695

Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B., 2013a. Poselet conditioned pictorial structures. IEEE Computer Vision and Pattern Recognition, 588–595.
URL http://ieeexplore.ieee.org/document/6618926/

Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B., 2013b. Strong appearance and expressive spatial models for human pose estimation. IEEE International Conference on Computer Vision, 3487–3494.
URL http://ieeexplore.ieee.org/document/6751545/

Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P., Schiele, B., 2015. DeepCut: joint subset partition and labeling for multi person pose estimation. arXiv preprint arXiv:1511.06645, 1–15.
URL http://arxiv.org/abs/1511.06645

Poggio, T., Ullman, S., 2013. Vision: are models of object recognition catching up with the brain? Annals of the New York Academy of Sciences 1305 (1), 72–82.
URL http://doi.wiley.com/10.1111/nyas.12148

Pollick, F., Fidoplastis, C., Braden, V., 1999. Training the recognition of biological motion. Investigative Ophthalmology and Visual Sciences 40, 3914.

Poppe, R., 2010. A survey on vision-based human action recognition. Image and Vision Computing 28 (6), 976–990.
URL http://linkinghub.elsevier.com/retrieve/pii/S0262885609002704

Prinz, W., 1996. Handbook of perception and action: perception. Perception 1, 448.

Radinsky, K., Davidovich, S., Markovitch, S., 2012. Learning causality for news events prediction. International conference on World Wide Web, 909.
URL http://dl.acm.org/citation.cfm?doid=2187836.2187958

Ramakrishna, V., Munoz, D., Hebert, M., Andrew Bagnell, J., Sheikh, Y., 2014. Pose machines: articulated pose estimation via inference machines. Springer LNCS 8690 LNCS, 33–47.
URL http://link.springer.com/10.1007/978-3-319-10605-2_3

Ramisa, A., Vasudevan, S., Scaramuzza, D., López de Mántaras, R., Siegwart, R., 2008. A tale of two object recognition methods for mobile robots. Computer Vision Systems, Proceedings 5008, 353–362\r560.
URL http://link.springer.com/10.1007/978-3-540-79547-6_34

Raymond, J., 1994. Directional anisotropy of motion sensitivity across the visual field. Vision Research 34 (8), 1029–1037.
URL http://linkinghub.elsevier.com/retrieve/pii/0042698994900078

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2015. You only look once: unified, real-time object detection. arXiv preprint arXiv:1506.02640, 1–10.
URL http://arxiv.org/abs/1506.02640

Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Pattern Analysis and Machine Intelligence 794, 185–192.
URL http://www.ncbi.nlm.nih.gov/pubmed/27295650

Riesenhuber, M., Poggio, T., 1999. Hierarchical models of object recognition in cortex. Nature neuroscience 2 (11), 1019–25.
URL http://www.ncbi.nlm.nih.gov/pubmed/10526343

Rizzolatti, G., Arbib, M., 1998. Language within our grasp. Trends in Neurosciences 21 (5), 188–194.
URL http://linkinghub.elsevier.com/retrieve/pii/S0166223698012600

Rizzolatti, G., Fadiga, L., Gallese, V., Fogassi, L., 1996. Premotor cortex and the recognition of motor actions. Cognitive Brain Research 3 (2), 131–141.
URL http://linkinghub.elsevier.com/retrieve/pii/0926641095000380

Rizzolatti, G., Sinigaglia, C., 2010. The functional role of the parieto-frontal mirror circuit: interpretations and misinterpretations. Nature Reviews Neuroscience 11 (4), 264–274.
URL http://www.nature.com/nrn/journal/v11/n4/full/nrn2805.html

Rodrigues, J., 2008. Integrated multi-scale architecture of the cortex with application to computer vision. Ph.D. thesis, University of the Algarve, Portugal.
URL http://sapientia.ualg.pt/handle/10400.1/413

Rodrigues, J., du Buf, J., 2006. Multi-scale keypoints in V1 and beyond: object segregation, scale selection, saliency maps and face detection. BioSystems 2, 75–90.
URL http://sapientia.ualg.pt/handle/10400.1/181

Rodrigues, J., du Buf, J., 2009a. A cortical framework for invariant object categorization and recognition. Cognitive Processing 10 (3), 243–261.
URL http://link.springer.com/10.1007/s10339-009-0262-2

Rodrigues, J., du Buf, J., 2009b. Multi-scale lines and edges in V1 and beyond: brightness, object categorization and recognition, and consciousness. BioSystems 95 (3), 206–226.
URL https://www.ncbi.nlm.nih.gov/pubmed/19026712

Ruonan Li, Zickler, T., 2012. Discriminative virtual views for cross-view action recognition. IEEE Computer Vision and Pattern Recognition, 2855–2862.
URL http://ieeexplore.ieee.org/document/6248011/

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., Fei-Fei, L., 2015. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision 115 (3), 211–252.
URL http://arxiv.org/abs/1409.0575

Saleiro, M., Farrajota, M., Terzić, K., Krishna, S., Rodrigues, J., du Buf, J., 2015. Biologically inspired vision for human-robot interaction. In: Universal Access in Human-Computer Interaction. Springer LNCS 9176, pp. 505–517.
URL http://link.springer.com/10.1007/978-3-319-20681-3_48

Saleiro, M., Farrajota, M., Terzić, K., Rodrigues, J., du Buf, J., 2013. A biological and real-time framework for hand gestures and head poses. Universal Access in Human-Computer Interaction. Design Methods, Tools, and Interaction Techniques for eInclusion 8089, 556–565.
URL http://link.springer.com/10.1007/978-3-642-39188-0_60

Saleiro, M., Rodrigues, J., du Buf, J. M. H., 2009. Automatic Hand or Head Gesture Interface for Individuals with Motor Impairments, Senior Citizens and Young Children. In Proceedings International Conference on Software Development for Enhancing Accessibility and Fighting Info-Exclusion, 165–171.
URL https://sapientia.ualg.pt/handle/10400.1/877?locale=en

Sapp, B., Taskar, B., 2013. MODEC: multimodal decomposable models for human pose estimation. IEEE Computer Vision and Pattern Recognition, 3674–3681.
URL http://ieeexplore.ieee.org/document/6619315/

Saygin, A., 2007. Superior temporal and premotor brain areas necessary for biological motion perception. Brain 130 (9), 2452–2461.
URL http://www.brain.oxfordjournals.org/cgi/doi/10.1093/brain/awm162

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural Networks 61, 85–117.
URL http://linkinghub.elsevier.com/retrieve/pii/S0893608014002135

Schölkopf, B., Burges, C., Smola, A., 1999. Advances in kernel methods: support vector learning. MIT Press Cambridge.
URL http://dl.acm.org/citation.cfm?id=299094

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y., 2013a. OverFeat: integrated recognition , localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229, 1–15.
URL http://arxiv.org/abs/1312.6229

Sermanet, P., Kavukcuoglu, K., Chintala, S., Lecun, Y., 2013b. Pedestrian detection with unsupervised multi-stage feature learning. IEEE Computer Vision and Pattern Recognition, 3626–3633.
URL http://ieeexplore.ieee.org/document/6619309/

Serre, T., Kouh, M., Cadieu, C., Knoblich, U., Kreiman, G., Poggio, T., 2005. A theory of object recognition: Computations and circuits in the feedforward path of the ventral stream in primate visual cortex. Artificial Intelligence (December), 1–130.
URL http://serre-lab.clps.brown.edu/wp-content/uploads/2012/08/GetTRDoc.pdf

Serre, T., Oliva, A., Poggio, T., 2007. A feedforward architecture accounts for rapid categorization. National Academy of Sciences 104 (15), 6424–6429.
URL http://www.pnas.org/content/104/15/6424.long

Shelton, J., Fouch, E., Caramazza, A., 1998. The selective sparing of body part knowledge: a case study. Neurocase 4 (4-5), 339–351.
URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0031713446&partnerID=40

Shotton, J., Blake, A., Cipolla, R., 2008. Efficiently combining contour and texture cues for object recognition. British Machine Vision Conference, 7.1–7.10.
URL http://www.bmva.org/bmvc/2008/papers/9.html

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations, 1–14.
URL http://arxiv.org/abs/1409.1556

Snavely, N., Seitz, S., Szeliski, R., 2006. Photo tourism: exploring photo collections in 3D. ACM Transactions on Graphics 25 (3), 835–846.
URL http://doi.acm.org/10.1145/1141911.1141964

Strecha, C., Bronstein, A., Bronstein, M., Fua, P., 2012. LDAHash: improved matching with smaller descriptors. IEEE Pattern Analysis and Machine Intelligence 34 (1), 66–78.

Suau, X., Ruiz-Hidalgo, J., Casas, J., 2012. Real-time head and hand tracking based on 2.5D data. IEEE Multimedia 14 (3 PART1), 575–585.

Suk, H., Sin, B., Lee, S., 2010. Hand gesture recognition based on dynamic Bayesian network framework. Pattern Recognition 43 (9), 3059–3072.
URL http://www.sciencedirect.com/science/article/pii/S0031320310001366

Sun, M., Savarese, S., 2011. Articulated part-based model for joint object detection and pose estimation. IEEE International Conference on Computer Vision, 723–730.
URL http://ieeexplore.ieee.org/document/6126309/

Sutskever, I., Vinyals, O., Le, Q., 2014. Sequence to sequence learning with neural networks. arXiv preprint arXiv:1409.3215, 1–9.
URL http://arxiv.org/abs/1409.3215

Swettenham, J., Campbell, R., 2005. Motion perception and autistic spectrum disorder : A review. Current Psychology of Cognition 23, 3–33.
URL http://discovery.ucl.ac.uk/77485/

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2014a. Going deeper with convolutions. International Conference on Computer Vision, 1879–1886.
URL http://arxiv.org/abs/1409.4842

Szegedy, C., Reed, S., Erhan, D., Anguelov, D., Ioffe, S., 2014b. Scalable, high-quality object detection. arXiv preprint arXiv:1412.1441, 1–10.
URL http://arxiv.org/abs/1412.1441

Szeliski, R., 2011. Computer vision. Vol. 5 of Texts in Computer Science. Springer London, London.
URL http://link.springer.com/10.1007/978-1-84882-935-0

Taigman, Y., Yang, M., Ranzato, M., Wolf, L., 2014. DeepFace: closing the gap to human-level performance in face verification. IEEE Computer Vision and Pattern Recognition, 1701–1708.
URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6909616

Terzić, K., Rodrigues, J. M. F., Du Buf, J. M. H., 2013. Real-time object recognition based on cortical multi-scale keypoints. Springer LNCS 7887, 314–321.
URL http://sapientia.ualg.pt/handle/10400.1/3374

Théoret, H., Pascual-Leone, A., 2002. Language acquisition: do as you hear. Current Biology 12 (21), R736–R737.
URL http://linkinghub.elsevier.com/retrieve/pii/S0960982202012514

Theusner, S., de Lussanet, M., Lappe, M., 2011. Adaptation to biological motion leads to a motion and a form aftereffect. Attention, Perception and Psychophysics 73 (6), 1843–1855.
URL http://link.springer.com/10.3758/s13414-011-0133-7

Tian, Y., Luo, P., Wang, X., Tang, X., 2015a. Deep learning strong parts for pedestrian detection. IEEE International Conference on Computer Vision, 1904–1912.
URL http://ieeexplore.ieee.org/document/7410578/

Tian, Y., Luo, P., Wang, X., Tang, X., 2015b. Pedestrian detection aided by deep learning semantic tasks. IEEE Computer Vision and Pattern Recognition, 5079–5087.

Tian Lan, Sigal, L., Mori, G., 2012. Social roles in hierarchical models for human activity recognition. IEEE Computer Vision and Pattern Recognition, 1354–1361.
URL http://ieeexplore.ieee.org/document/6247821/

Tittle, J., Perotti, V., 1997. The perception of shape and curvedness from binocular stereopsis and structure from motion. Perception and Psychophysics 59 (8), 1167–1179.
URL http://www.springerlink.com/index/10.3758/BF03214205

Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C., 2015. Efficient object localization using Convolutional Networks. IEEE Computer Vision and Pattern Recognition, 648–656.
URL http://ieeexplore.ieee.org/document/7298664/

Tompson, J., Jain, A., LeCun, Y., Bregler, C., 2014. Joint training of a convolutional network and graphical model for human pose estimation. Advances in neural information processing systems, 1799–1807.
URL http://arxiv.org/abs/1406.2984

Torralba, A., 2003. Contextual priming for object detection. International Journal of Computer Vision 53 (2), 169–191.
URL http://link.springer.com/10.1023/A:1023052124951

Toshev, A., Szegedy, C., 2014. DeepPose: human pose estimation via deep neural networks. IEEE Computer Vision and Pattern Recognition, 1653–1660.
URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6909610

Tran, K., Gala, A., Kakadiaris, I., Shah, S., 2014. Activity analysis in crowded environments using social cues for group discovery and human interaction modeling. Pattern Recognition Letters 44, 49–57.
URL http://linkinghub.elsevier.com/retrieve/pii/S0167865513003516

Tran, K., Kakadiaris, I., Shah, S., 2012. Part-based motion descriptor image for human action recognition. Pattern Recognition 45 (7), 2562–2572.
URL http://linkinghub.elsevier.com/retrieve/pii/S0031320312000222

Uijlings, J., van de Sande, K., Gevers, T., Smeulders, A., 2013. Selective search for object recognition. International Journal of Computer Vision 104 (2), 154–171.
URL http://link.springer.com/10.1007/s11263-013-0620-5

Uithol, S., van Rooij, I., Bekkering, H., Haselager, P., Rooij, I., 2012. Hierarchies in action and motor control. Journal of Cognitive Neuroscience 24 (5), 1077–86.
URL http://www.ncbi.nlm.nih.gov/pubmed/22288396

Ullman, S., Vidal-Naquet, M., Sali, E., 2002. Visual features of intermediate complexity and their use in classification. Nature Neuroscience 5 (7), 682–687.
URL http://www.nature.com/doifinder/10.1038/nn870

Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M., Dolan, J., Duggins, D., Galatali, T., Geyer, C., Gittleman, M., Harbaugh, S., Hebert, M., Howard, T., Kolski, S., Kelly, A., Likhachev, M., McNaughton, M., Miller, N., Peterson, K., Pilnick, B., Rajkumar, R., Rybski, P., Salesky, B., Seo, Y., Singh, S., Snider, J., Stentz, A., Whittaker, W., Wolkowicki, Z., Ziglar, J., Bae, H., Brown, T., Demitrish, D., Litkouhi, B., Nickolaou, J., Sadekar, V., Zhang, W., Struble, J., Taylor, M., Darms, M., Ferguson, D., 2009. Autonomous driving in urban environments: boss and the urban challenge. In: Springer Tracts in Advanced Robotics. Vol. 56. pp. 1–59.
URL http://link.springer.com/10.1007/978-3-642-03991-1_1

Van Overwalle, F., 2009. Social cognition and the brain: A meta-analysis. Human Brain Mapping 30 (3), 829–858.
URL http://doi.wiley.com/10.1002/hbm.20547

Vanetti, M., Binaghi, E., Carminati, B., Carullo, M., Ferrari, E., 2011. Content-based filtering in on-line social networks. Springer LNCS 6549, 127–140.
URL http://link.springer.com/10.1007/978-3-642-19896-0_11

Vangeneugden, J., De Mazière, P., Van Hulle, M., Jaeggli, T., Van Gool, L., Vogels, R., 2011. Distinct mechanisms for coding of visual actions in macaque temporal cortex. Journal of Neuroscience 31 (2), 385–401.
URL http://www.ncbi.nlm.nih.gov/pubmed/21228150

Vangeneugden, J., Pollick, F., Vogels, R., 2009. Functional differentiation of macaque visual temporal cortical neurons using a parametric action space. Cerebral Cortex 19 (3), 593–611.
URL https://academic.oup.com/cercor/article-lookup/doi/10.1093/cercor/bhn109

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Vol. 8.
URL http://portal.acm.org/citation.cfm?id=211359

Vasudevan, S., Gächter, S., 2007. Cognitive spatial representations for mobile robots - perspectives from a user study. IEEE International Conference on Robotics and Automation.
URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.149.5269

Viola, P., Jones, M., 2004. Robust real-time face detection. International Journal of Computer Vision 57 (2), 137–154.
URL http://link.springer.com/10.1023/B:VISI.0000013087.49260.fb

Viola, P., Jones, M., Snow, D., 2005. Detecting pedestrians using patterns of motion and appearance. International Journal of Computer Vision 63 (2), 153–161.

Vogt, S., Buccino, G., Wohlschläger, A., Canessa, N., Shah, N. J., Zilles, K., Eickhoff, S., Freund, H., Rizzolatti, G., Fink, G., 2007. Prefrontal involvement in imitation learning of hand actions: effects of practice and expertise. NeuroImage 37 (4), 1371–1383.
URL http://linkinghub.elsevier.com/retrieve/pii/S1053811907006076

Vrigkas, M., Karavasilis, V., Nikou, C., Kakadiaris, I., 2013. Action recognition by matching clustered trajectories of motion vectors. International Conference on Computer Vision Theory and Applications, 112–117.

Vrigkas, M., Nikou, C., Kakadiadis, I., 2014. Classifying behavioral attributes using conditional random fields. Hellenic Conference on Artificial Intelligence, 95–104.
URL http://link.springer.com/10.1007/978-3-319-07064-3_8

Vrigkas, M., Nikou, C., Kakadiaris, I., 2015. A review of human activity recognition methods. Frontiers in Robotics and AI 2 (November), 1–28.
URL http://journal.frontiersin.org/Article/10.3389/frobt.2015.00028/abstract

Walk, S., Majer, N., Schindler, K., Schiele, B., 2010. New features and insights for pedestrian detection. IEEE Computer Vision and Pattern Recognition, 1030–1037.
URL http://ieeexplore.ieee.org/document/5540102/

Wang, R., Han, C., Wu, Y., Guo, T., 2014. Fingerprint classification based on depth neural network. arXiv preprint arXiv:1409.5188, 1–14.
URL http://arxiv.org/abs/1409.5188

Wang, X., Han, T., Yan, S., 2009. An HOG-LBP human detector with partial occlusion handling. IEEE International Conference on Computer Vision (Iccv), 32–39.
URL http://ieeexplore.ieee.org/document/5459207/

Wei, K., Huang, J., Fu, S., 2007. A survey of e-commerce recommender systems. IEEE International Conference on Service Systems and Service Management, 1–5.
URL http://ieeexplore.ieee.org/document/4280214/

Wei, S., Ramakrishna, V., Kanade, T., Sheikh, Y., 2016. Convolutional pose machines. Computer Vision and Pattern Recognition, 4724–4732.
URL http://arxiv.org/abs/1602.00134

Weinland, D., Ronfard, R., Boyer, E., 2011. A survey of vision-based methods for action representation, segmentation and recognition. Computer Vision and Image Understanding 115 (2), 224–241.
URL http://linkinghub.elsevier.com/retrieve/pii/S1077314210002171

Werbos, P., 1982. Applications of advances in nonlinear sensitivity analysis. In: System Modeling and Optimization. Springer-Verlag, Berlin/Heidelberg, pp. 762–770.
URL http://www.springerlink.com/index/10.1007/BFb0006203

Wu, Q., Wang, Z., Deng, F., Chi, Z., Feng, D., 2013. Realistic human action recognition with multimodal feature selection and fusion. IEEE Systems, Man, and Cybernetics: Systems 43 (4), 875–885.
URL http://ieeexplore.ieee.org/document/6493474/

Xiong, H., Alipanahi, B., Lee, L., Bretschneider, H., Merico, D., Yuen, R., Hua, Y., Gueroussov, S., Najafabadi, H., Hughes, T., Morris, Q., Barash, Y., Krainer, A., Jojic, N., Scherer, S., Blencowe, B., Frey, B., 2015. The human splicing code reveals new insights into the genetic determinants of disease. Science. 347 (6218), 1254806.
URL http://www.sciencemag.org.libproxy.mit.edu/content/347/6218/1254806.full

Xu, B., Wang, N., Chen, T., Li, M., 2015. Empirical evaluation of rectified activations in convolutional network. ICML Deep Learning Workshop, 1–5.
URL http://arxiv.org/abs/1505.00853

Xu, Y., Xiao, T., Zhang, J., Yang, K., Zhang, Z., 2014. Scale-invariant convolutional neural networks. arXiv preprint arXiv:1411.6369, 1–9.
URL http://arxiv.org/abs/1411.6369

Yang, B., Yan, J., Lei, Z., Li, S., 2015a. Convolutional channel features. IEEE International Conference on Computer Vision, 82–90.
URL http://ieeexplore.ieee.org/document/7410375/

Yang, Y., Saleemi, I., Shah, M., jul 2013. Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. IEEE Pattern Analysis and Machine Intelligence 35 (7), 1635–1648.
URL http://ieeexplore.ieee.org/document/6365192/

Yang, Y., Wang, Z., Wu, F., 2015b. Exploring prior knowledge for pedestrian detection. British Machine Vision Association, 1–12.
URL http://www.bmva.org/bmvc/2015/papers/paper176/index.html

Zagoruyko, S., Lerer, A., Lin, T., Pinheiro, P., Gross, S., Chintala, S., Dollár, P., 2016. A multiPath network for object detection. arXiv preprint arXiv:1604.02135, 1–14.
URL http://arxiv.org/abs/1604.02135

Zeiler, M., 2012. ADADELTA: an adaptive learning rate method. arXiv preprint arXiv:1212.5701, 1–6.
URL http://arxiv.org/abs/1212.5701

Zhang, N., Paluri, M., Taigman, Y., Fergus, R., Bourdev, L., 2015a. Beyond frontal faces: improving person recognition using multiple cues. IEEE Computer Vision and Pattern Recognition, 4804–4813.
URL http://ieeexplore.ieee.org/document/7299113/

Zhang, S., Benenson, R., Omran, M., Hosang, J., Schiele, B., 2016. How far are we from solving pedestrian detection? IEEE Computer Vision and Pattern Recognition, 1259–1267.
URL http://ieeexplore.ieee.org/document/7780510/

Zhang, S., Benenson, R., Schiele, B., 2015b. Filtered channel features for pedestrian detection. IEEE Computer Vision and Pattern Recognition, 1751–1760.
URL http://ieeexplore.ieee.org/document/7298784/

Zhang, X., LeCun, Y., 2015. Text understanding from scratch. arXiv preprint arXiv:1502.01710, 1–10.
URL http://arxiv.org/abs/1502.01710