Multi-object detection and tracking in video sequences

Ala Mhalla

UNIVERSITY of Clermont Auvergne - Clermont Ferrand

EDSPI DOCTORAL SCHOOL

P H D T H E S I S

to obtain the title of

DOCTOR OF THE UNIVERSITY

of Clermont Auvergne University - Clermont-France

Speciality : Computer Science

Defended by

Ala Mhalla

Multi-object Detection and Tracking in Video Sequences

Defended on 4 Avril 2018

Jury :

Reviewer:	M. Christophe Garcia	-	LIRIS laboratory
Reviewer:	M. Adel Alimi	-	REGIM laboratory
Examiner	M. Christophe Ducottet	-	Hubert Curian laboratory
Co-director:	Mme. Najoua Essoukri ben Amara	-	LATIS laboratory
Co-director:	M.Thierry Chateau	-	Institut Pascal laboratry
Co-supervisor:	M. Sami Gazzah	-	LATIS laboratory

Abstract:

The work developed in this PhD thesis is focused on video sequence analysis. The latter consists of object detection, categorization and tracking. The development of reliable solutions for the analysis of video sequences opens new horizons for several applications such as intelligent transport systems, video surveillance and robotics.

In this thesis, we put forward several contributions to deal with the problems of detecting and tracking multi-objects on video sequences. The proposed frameworks are based on deep learning networks and transfer learning approaches.

In a first contribution, we tackle the problem of multi-object detection by putting forward a new transfer learning framework based on the formalism and the theory of a Sequential Monte Carlo (SMC) filter to automatically specialize a Deep Convolutional Neural Network (DCNN) detector towards a target scene. The suggested specialization framework is used in order to transfer the knowledge from the source and the target domain to the target scene and to estimate the unknown target distribution as a specialized dataset composed of samples from the target domain. These samples are selected according to the importance of their weights which reflects the likelihood that they belong to the target distribution. The obtained specialized dataset allows training a specialized DCNN detector to a target scene without human intervention.

In a second contribution, we propose an original multi-object tracking framework based on spatio-temporal strategies (interlacing/inverse interlacing) and an interlaced deep detector, which improves the performances of tracking-by-detection algorithms and helps to track objects in complex videos (occlusion, intersection, strong motion).

In a third contribution, we provide an embedded system for traffic surveillance, which integrates an extension of the SMC framework so as to improve the detection accuracy in both day and night conditions and to specialize any DCNN detector for both mobile and stationary cameras.

Throughout this report, we provide both quantitative and qualitative results. On several aspects related to video sequence analysis, this work outperforms the state-of-the-art detection and tracking frameworks. In addition, we have successfully implemented our frameworks in an embedded hardware platform for road traffic safety and monitoring.

Keywords:

Artificial intelligence, Computer vision, Transfer learning, Deep learning, Multi-object detection, Specialization, Tracking-by-detection, Multi-object tracking, Embedded system.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

One of the many human capacities is the remarkable ability to understand and analyze the environment. From the signals provided by their ocular system, the human being is able to describe the objects that surround them in a very precise and quick way. We can notably emphasize the capacity of the human being to locate and categorize objects while characterizing them by their forms, colors and orientations. One of the many goals of computer vision researchers is to build an intelligent system capable of efficiently analyzing images as human beings. To be reliable, the analysis algorithms must adapt to changes in the appearance of objects related to the context in which they are observed. For example, the appearance of an object may vary depending on the brightness of the environment or it may be partially hidden by another object. This is done by taking into account these constraints, naturally managed by the human brain, which the artificial system must integrate in order to hope to behave like a true intelligent system.

1.1 Context of the thesis

The work done in this PhD project focuses on multi-object detection and tracking, which is based on supervised learning for the automatic analysis of video sequences. Figure 1.1 illustrates different applications of this work. Detecting and tracking objects remain an important issue because of the number of applications they generate. Among them, we can cite video-surveillance or robotics.

In the context of surveillance for the security of transport infrastructure, the system must be capable of analyzing and monitoring traffic flows in urban or high-speed areas and collecting statistics, thereby improving safety of road transport. These include video surveillance applications such as monitoring and securing transport infrastructure. We can imagine other video surveillance applications using detection and tracking such as access control, which requires special surveillance and/or high security (metro stations, supermarkets, government institutions, companies, airports, hospitals, research laboratories, etc.).

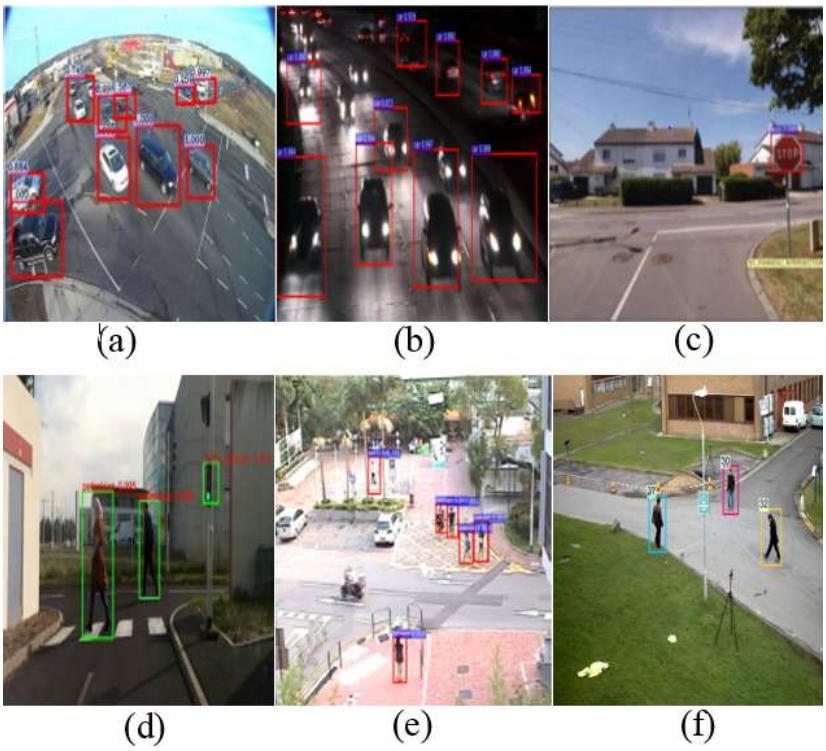


Figure 1.1: Various applications of the work carried out in this thesis. Detection and tracking objects in different scenarios. (a) and (b): Object detection in both day and night conditions. (c): Traffic sign detection. (d): Object detection system for an autonomous vehicle. (e) and (f): Multi-object tracking in several conditions.

.

Other types of applications are inherent to the implementation of an analysis system within a vehicle: These are intelligent vehicle applications. These include driver assistance, automatic parking and self-driving. In this context, the detection and the tracking of objects present around the intelligent vehicle is necessary. This predicts the trajectory and speed appropriate to the situation. The constantly evolving performances of these systems will certainly make it possible in the coming years to integrate into the transport infrastructures and the autonomous vehicles of artificial intelligence by vision: a driving system without drivers.

1.2 Problematics

This thesis proposes automatic frameworks for multi-object detection and tracking. In other words, the input of the developed frameworks is a video scene and a generic deep detector and the output is a video containing detected and tracked objects. We have investigated solutions based on transfer learning, deep learning and spatio-temporal strategies to develop our detection and tracking frameworks. The issues addressed are:

- **Multi-object detection:** It consists in proposing a set of rectangles (bounding boxes) containing target objects. This task is necessary to several computer vision applications, in particular in object tracking one. Object detection is a well-studied problem and the main challenges are multiple: occlusion, point of view from which objects are observed, light condition, deformation, intra-class variation, confusion with the background variation, and scale variation (as shown in Figure 2.2).

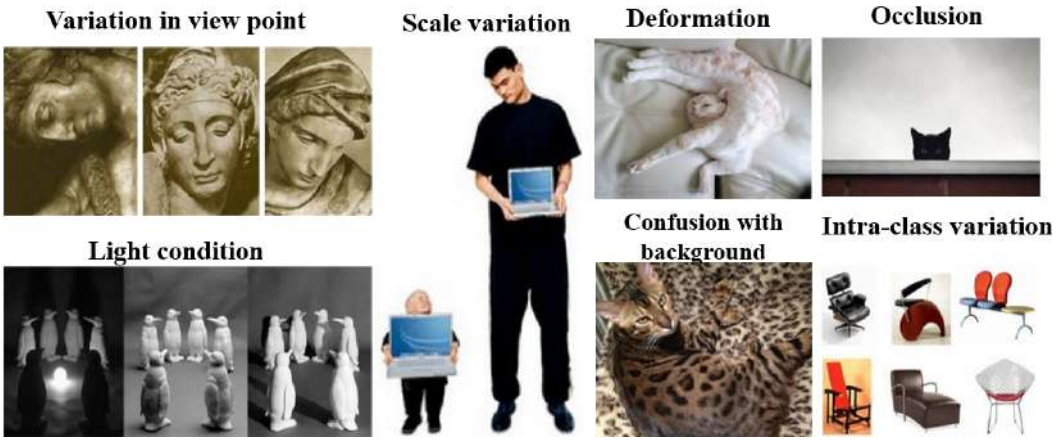


Figure 1.2: Main challenges on object detection (translated from a presentation of Andrew Zisserman, VGG, Oxford) [Simonyan 2014].

- **Multi-object tracking:** The purpose of tracking is to generate the trajectories of objects. Each object has a unique identity assigned to it by the association algorithm. The correspondence between the positions of an object in successive images constitutes the trajectory. In this thesis, we have addressed tracking-by-detection methods which associate successive detection of objects.

- **Specialization:** Generally, the performance of a generic detector decreases significantly when it is tested on a specific scene due to the large variation between the source training dataset and the target scene. This problem can be solved by specialization. The main idea of specialization is to provide a specialized detector or tracker to a particular scene in order to increase the detection or tracking performances toward a target scene. Specialization is referred to by several terms in the literature such as adaptation, contextualisation, etc.

1.3 Contributions and organization of the manuscript

This thesis integrates recent advances in the computer vision domain such as transfer learning and deep learning techniques in order to enhance the detection and the

tracking performances.

In the past five years, deep learning and transfer learning have been widely used by the computer vision community to solve a lot of tasks. This interest is due to their ability to extract highly relevant information on images, thus allowing the training of high-performance models.

In this respect, deep learning and/or transfer learning seems to be suitable for the construction of multi-object detection and tracking systems. Nevertheless, the very good results of methods using deep learning and/or transfer learning are intimately correlated with the number of data used for their learning.

On the basis of this observation, the general contribution of this thesis is to propose automatic specialization frameworks based on Deep Convolutional Neural Network (DCNN) model, in order to improve the performances of detection and tracking objects in video sequences.

This document is organized as follows. In chapter 2, we present the state-of-the-art in relation with our problematics, namely the deep learning and transfer learning. Object detection is introduced in section 1. Section 2 presents a detailed description of supervised learning as well as deep neural networks used throughout our work. An overview of approaches related to transfer learning is proposed in section 3.

Chapters 3,4 and 5 present the frameworks developed in addition to the proposed contributions.

In chapter 3, we propose a DCNN specialization framework based on a Sequential Monte Carlo (SMC) filter. The idea of this first contribution is to specialize a generic deep detector towards a target scene based on a transductive transfer learning model. The suggested specialization framework leads to improve the performance and accuracy of DCNN detectors in each specific scene.

Chapter 4 presents a novel framework for multi-object tracking based on spatio-temporal strategies and an interlaced DCNN detector. The proposed framework makes it possible to improve the tracking performances toward a tracking datasets and to handle the tracking challenges mainly occlusion and intersection.

Chapter 5 presents an applicative contribution: an embedded system for traffic surveillance that can be performed to operate under challenging conditions such as congestion, occlusion and lighting night/day and day/night transitions. This system analyses traffic and particularly focuses on the problem of detecting and categorizing traffic objects on several traffic scenes. Moreover, it contains a robust detector produced by an original specialization framework. The proposed specialization framework presents an extension of the SMC framework mentioned in chapter 3.

Finally, chapter 6 represents the conclusion and the perspectives of this research.

Figure 1.3 presents the outline of the manuscript.

1.4 Publications

The work presented in this thesis has been the subject of the following publications:

- **Journals:**

  Ala Mhalla, Thierry Chateau, Houda Maâmatou, Sami Gazzah and Najoua Essoukri Ben Amara: "SMC Faster R-CNN: Towards a Scene-Specialized Multi-Object Detector": CVIU journal "Computer Vision and Image Understanding" Special Issue on Deep Learning for Computer Vision", 2017.(Impact factor 2.5 THOMSON JCR)

  Ala Mhalla, Thierry Chateau, Sami Gazzah and Najoua Essoukri Ben Amara: "AN EMBEDDED COMPUTER-VISION SYSTEM FOR MULTI-OBJECT DETECTION IN TRAFFIC SURVEILLANCE", IEEE transaction on intelligent transportation system "ITS" 2018. (Impact factor 3.7 THOMSON).

- **International Conferences :**

  Ala Mhalla, Thierry Chateau, Sami Gazzah and Najoua Essoukri Ben Amara: "The Power of Video Interlacing for Deep Learning Based Multi Object Tracking". Submitted to the European Conference on Computer Vision (ECCV 2018).

  Ala Mhalla, Thierry Chateau , Sami Gazzah and Najoua Essoukri Ben Amara: "Specialization of a Generic Pedestrian Detector to a Specific Traffic Scene based on Transductive Transfer Learning Method and Deep learning", VISAPP'17: Proceedings of the International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (Class C).

  Ala Mhalla, Houda Maâmatou, Thierry Chateau, Sami Gazzah and Najoua Essoukri Ben Amara:"Faster R-CNN Scene Specialization with a Sequential Monte-Carlo Framework" in DICTA'16: Proceedings of International Conference on Digital Image Computing: Techniques and Applications (Class B).

  Ala Mhalla, Thierry Chateau, Sami Gazzah, Najoua Essoukri Ben Amara: Scene-Specific Pedestrian Detector Using Monte Carlo Framework and Faster R-CNN Deep Model: PhD Forum. Proceedings of the 10th International Conference on Distributed Smart Camera (ICDSC 2016), France, (Class B).

Sami Gazzah, Ala Mhalla, Najoua Essoukri Ben Amara: Vehicle detection on a video traffic: review and new perpectives. 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunication, SETIT'17, Hammamet, Tunisia, December 18-20, 2016.

Ala Mhalla, Thierry Chateau, Sami Gazzah, Najoua Essoukri Ben Amara: A Faster R-CNN Multi-Object Detector on a Nvidia Jetson TX1 Embedded System: Demo. Proceedings of the 10th International Conference on Distributed Smart Camera, 208-209, Paris/France, September 12-15, 2016 (Class B).

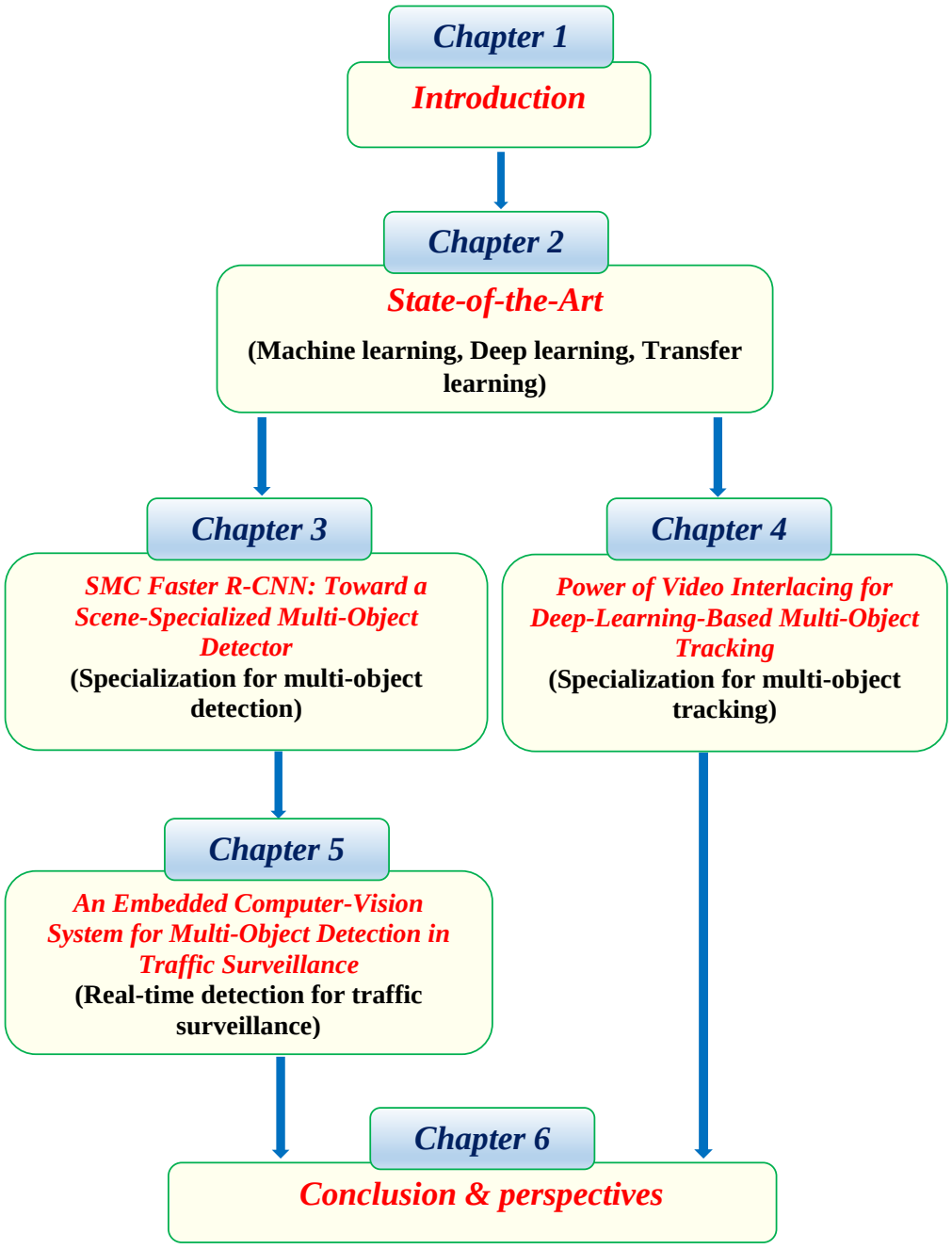


Figure 1.3: Outline of manuscript

Chapter 2

State-of-the-Art

Contents

2.1 Introduction

The aim of this chapter is to present the state-of-the-art and the work in relation with our problematics, namely the deep learning and transfer learning. We have divided this state-of-the-art into several sections. Section 1 refers to the different strategies of object detection. Section 2 presents the Deep Convolutional Neural Networks (DCNN) and reviews the existing work performed in deep neural networks. After that, a detailed description of transfer learning is provided in section 3. The applications of transfer learning in object detection are described in section 4. Finally, the conclusion of this chapter is given in section 5.

2.2 Object detection

Object detection is one of the most studied problems in several computer vision applications such as object tracking [Nam 2016][Wang 2016b], semantic segmentation [Long 2015][Noh 2015] and object recognition [Huang 2012][Taigman 2014]. The aim of object detection is to find in an input image a set of Regions of Interest (RoI) containing target objects. Object detection approaches can be divided into two categories: single-object detection and multi-object detection. The first category focuses on detecting only one type of objects. The detector must be able to decide whether a region of an image corresponds to an object or a background. The second category concentrates on multi-object detection where the detector must be able to predict what type of objects is concerned. There are many public datasets for training and evaluating detectors. We can cite among them PascalVOC [Everingham 2010], KITTI [Geiger 2012], MSCOC [Lin 2014] and ILSVRC [Deng 2009]. Figure 2.1 illustrates the main objective of object detection.

Figure 2.1: Examples of object detection returned by a multi-object detector. Each bounding box color corresponds to a class of objects. Source [Ren 2015c]

.

Detection challenges: An object detector faces generally several challenges. We quote first the computation time: A robust detector is a detector that should maximize performance while detecting objects as fast as possible. This notion of computation time is particularly important in the design of multi-class detectors because of the number, often very large of object classes to be detected.

The second challenge concerns the variation in object appearances. The latter can vary according to several factors such as: the variation in image resolutions, the size of objects, the points of view under which objects are observed, and light condition.

The third challenge is related to the variation in the appearance of regions that do not correspond to objects (the background). A detector must be able to predict whether a region corresponds to the background even in images resulting from complex environments. Figure 2.2 depicts major challenges on object detection.

In the following, we propose more details of the strategies for extracting regions on which the model will be applied. In the next section, we first develop the detection by the sliding window then the detection by object proposals.

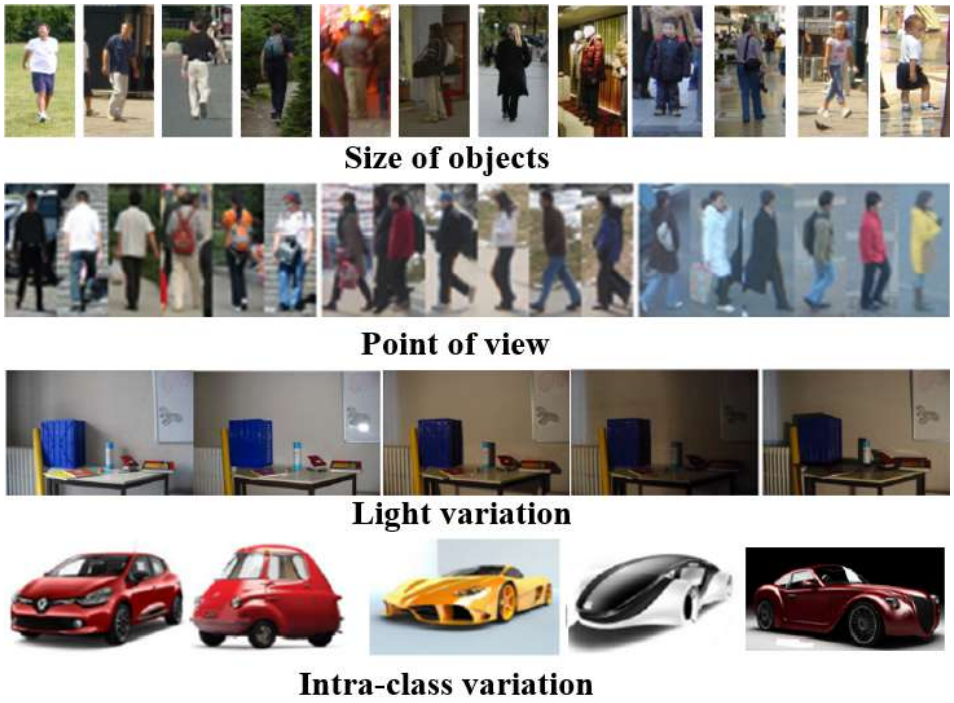


Figure 2.2: Detection challenges (translated from a presentation of Ala Mhalla [Mhalla 2016b], DICTA, Australia).

2.2.1 Detection by sliding window

Sliding-window approaches use a previously learnt classification model which consists of a 2D window with fixed size permitting the discrimination of the background of objects. The general idea of this type of approaches is to scan the input image using a sliding window. On the other hand, this window goes over the input image and produces a confidence score in each image position. In what follows we recall the main steps of the classical sliding window approach based on the pyramid representation (Figure 2.3).

- Given an input image, an image pyramid is computed. The computation of this latter consists in resizing an image using several scale factors and several ratios. The set of these images forms the pyramid and each image corresponds to one level. It is necessary to compute a pyramid of images because the sliding window to be applied to the image is of a fixed size. The detection model is generally learnt for a single scale and a single ratio. However, the objects in an image can be of different sizes and have variable ratios. In this way, the objects in the image correspond to the size of the model at least for one level of the pyramid.

- Visual characteristics are extracted on different levels of the pyramid, which makes it possible to obtain a feature map for each pyramidal level.

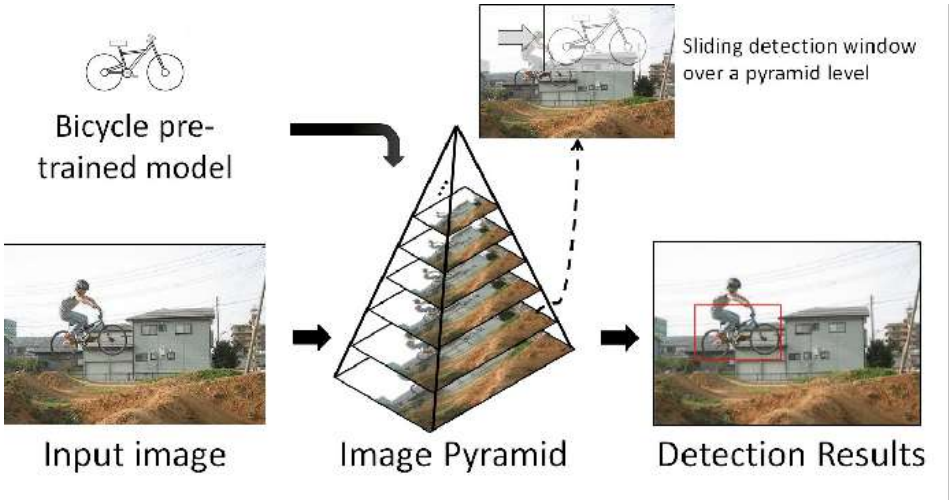


Figure 2.3: Classical diagram of object detection by sliding window. A pyramid of images is created from an input image. Visual characteristics are computed. A model is applied in each position and each pyramid level returns candidate boxes. An NMS algorithm is then applied to delete the boxes corresponding to the same object. Source [Suleiman 2017]

- A previously learnt model is applied in each position of the different feature maps, returning a confidence score for each position and each level of the pyramid. This step makes it possible to select candidate boxes (those with the highest confidence scores).

- In detection systems, it is very frequent that candidate boxes are agglomerated around the same object. In other words, several boxes can correspond to the same object. To remove any redundant detection, the Non-Maxima Suppression (NMS) algorithm is applied. The idea of this algorithm is based on the fact that detection cannot spatially overlap beyond a certain threshold (the overlap). If detected bounding boxes overlap too much, the bounding box with the best confidence score is kept and the others are removed.

The sliding window strategy has been widely used for object detection.The different methods that utilized it are distinguished by the nature of the model to be applied to the pyramid and the used visual characteristics (shape characteristics [Ferrari 2010][Ferrari 2007], Histograms of Oriented Gradients (HOG) [Dalal 2005][Felzenszwalb 2010], deep characteristics [Sermanet 2013][Szegedy 2013]...).

The sliding window has become unavoidable, especially after the publication of [Viola 2001a]. The authors introduced an approach based on the boosting concept [Freund 1995]. The idea of this latter was to scan the image with "weak" classifiers called Haar filters. The sum of the responses of these weak classifiers enables the decision making of the detector. This method has long been considered as a reference

in face detection applications. Its major advantage is computation time. Moreover, in this method, weak classifiers are used in cascade. In other words, they allow, at first, the quickly removal of regions that do not contain objects.

For the most problematic regions, the aggregation of weak classifiers makes it possible to build classifiers that are increasingly robust. Extensions based on boosting have also been introduced in particular to solve the multi-object detection problem [Torralba 2004][Torralba 2007]. These approaches have proposed to share visual characteristics between object classes to be detected.

In 2005, the authors of [Dalal 2005] suggested new visual characteristics: the HOG. These characteristics, based on the gradients of the image made a leap forward in the performance of object detection systems. The authors proposed the use of the HOGs and the SVMs to separate the learning examples in the space of characteristics. The HOGs and the SVMs were also used in the deformable part-model. [Felzenszwalb 2010] approach where the detection model was based on local and global representations of objects. This method remained a few years in the state-of-the-art person detection.

Approaches based on the sliding window and the Convolutional Neural Network (CNN) were also introduced. Among them, we can cite [Garcia 2002] and lately [Sermanet 2013][Szegedy 2013]. In [Sermanet 2013], the authors proposed to transform a classical neural network by replacing the full connected layers by convolutional ones. This allowed the application of the neural network on any image size, which was very interesting to specially pass in the network images from different pyramid levels. The fully convolutional network is trained to return confidence scores for each class of objects and the four corners of their bounding box. The CNN theory will be explained in more details in section 2.3.

2.2.2 Detection by object proposal

An alternative to the exhaustive search of objects (the sliding window) is the use of object proposal algorithms. Recently, the latter had improved the performance and computation time in object detection systems. The aim of these methods is to propose boxes with a high probability of being a target object. These boxes are then extracted and sent to a classifier for the final decision. These methods reduce the computation time because they considerably decrease the space of search compared to the exhaustive search methods of the sliding window type: The object detection model is not applied to all the positions of the image but only on a small set of regions. In the following, we present the existing methods permitting the generation of object propositions.

One of the most algorithms for the object proposition is the *selective search* introduced in [Uijlings 2013]. Compared to other approaches [Carreira 2010][Endres 2010], the *selective search* approach is based on a segmentation of an image at different resolutions. Using the segmentation method introduced in [Felzenszwalb 2004], *selective search* segments the input image on several scales. This produces a first set of RoI. The authors of [Uijlings 2013]

introduced then a similarity computation between regions based on color, texture, size and inclusion information. This similarity made it possible to merge redundant regions (too similar) and to return a set of propositions of relevant objects. Figure 2.4 illustrates the *selective search* algorithm.

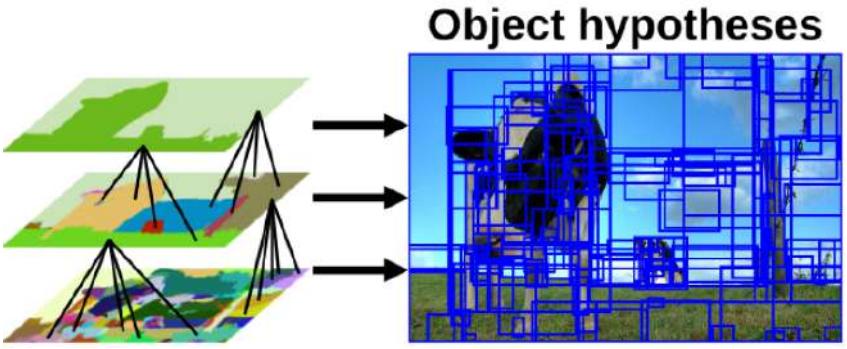


Figure 2.4: Illustration of selective search. On the left, segmentation maps at different resolutions. On the right, different propositions of objects returned by the algorithm after fusion of regions. Source [Uijlings 2013].

This method was subsequently used in two popular state-of-the-art references for object detection by CNN: R-CNN [Girshick 2014a] and Fast R-CNN [Girshick 2015a]. In the R-CNN [Girshick 2014a], object propositions resulting from the *selective search* were extracted in an image and resized to a fixed size. These regions were then used by a CNN to determine their classes. This approach increased the computation time (for learning and testing) because each region passed through all layers of the CNN. It allowed extracting regions from the *selective search* on a deep feature map; i.e., the entire input image was passed in a CNN providing a low-resolution feature map (due to the successive pooling). The object propositions were then extracted on this map and send to a classifier consisting generally of two hidden layers (full-connected layers) and an output layer enabling the classification of the object (class of object or background). In these two approaches, an additional function was learnt by the network making it possible to transform the propositions of original objects produced by the *selective search* so that these would stick to the object as well as possible. This function was called "regression on the boxes".

To further reduce the computing time and increase performance, the object proposal network [Ren 2015c], "Region Proposal Network (RPN)", was introduced. The authors in [Ren 2015c] proposed to create a single CNN capable of suggesting interest objects, extracting them on a feature map and classifying each region. This method for object detection has been widely used and modified [Yang 2016] [Xiang 2017] [Kong 2016] thanks to its performance and speed on object detection. Recent work [He 2017] has used the RPN even for the segmentation of instances and the estimation of person positions. The approach of [Ren 2015c] will be explained in more details in chapter 3. Other CNN detection algorithms have been introduced [Liu 2016] [Redmon 2016a]. They are based on the one-shot concept;

i.e., they no longer use a step of extracting object proposals on the feature maps. This saves even more computing time when utilizing the CNN on an image. A very interesting article [Huang 2016b] provides a very thorough analysis of different state-of-the-art detectors based on CNN [Ren 2015c][Liu 2016] by testing various CNN architectures.

2.3 Initiation of deep neural network

This section aims to present the Deep Neural Networks (DNN) used throughout this thesis.

Generally, neural networks encode a mathematical function to be applied to an input signal (in our case, an image) and making it possible to predict an output signal. This function is a composition of several nonlinearn or linear functions. These networks are inspired from the functioning of the human brain. They are made up of a large number of artificial neurons (introduced for the first time in 1943 by McCulloch *et al.* [McCulloch 1943]) connected together, which model the running of biological neurons. Figure 2.5 illustrates the analogy between the human brain and the neural networks. These networks have existed for a long time (McCulloch 1943, Rosenblatt 1958, Minsky 1969), but their study stagnated until the end of the 1990s. Since then, they outperformed many methods in several computer vision applications such as image classification, detection, segmentation and recognition.

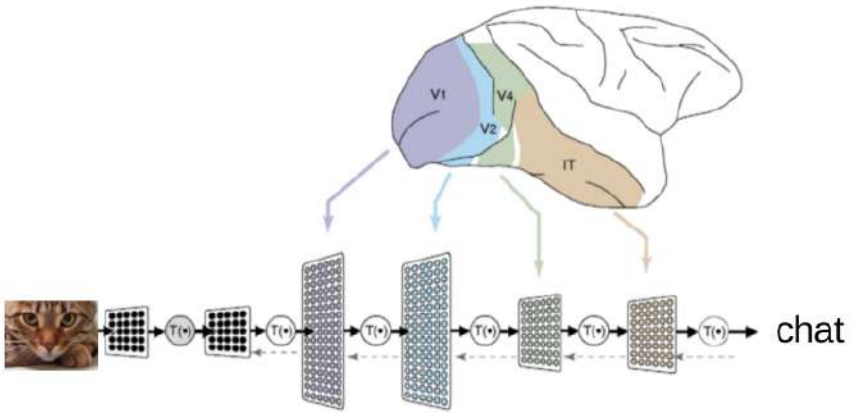


Figure 2.5: Analogy between the human brain and neural networks. A signal (here an image) activates neural responses in series to response (here "cat"). Source [DiCarlo 2007]

This recent enthusiasm, particularly in the field of computer vision, can be explained in several ways. First, enormous annotated public databases are currently available like, Image-Net [Deng 2009], MSCOCO [Lin 2014], YouTube-8M [Abu-El-Haija 2016] and Cityscapes [Cordts 2016]. These latter datasets make it possible to learn complex neural networks. For example, the ImageNet dataset

[Deng 2009] contains approximately 14 million images corresponding to 22,000 object classes. Thus, this type of database enabled [Krizhevsky 2012] the winning of the ImageNet competition in object classification (1.2 million images corresponding to 1000 classes) by proposing a neural architecture containing 60 million parameters. It is this large number of parameters that differentiates modern neural networks from those of the 1990s. The second reason to relaunch the study of neural networks is the ability of modern machines to perform huge computations in a reasonable time, notably thanks to the use of graphic cards (GPUs). This makes it possible to build neural networks that are increasingly complex and efficient. Furthermore, the multiplication of deep learning frameworks such as: Caffe [Jia 2014], Tensorflow [Abadi 2015] and Torch [Collobert 2002], permits the easy development of automatic learning methods.

In what follows, we will recall the objectives of supervised learning and the interest of neural networks in relation to classical learning approaches. After that, we will present the CNN.

2.3.1 Supervised learning and neural networks

Machine learning is a broad subject of research. We can distinguish different learning families (supervised, semi-supervised, unsupervised, by reinforcement ...). The purpose of automatic learning is to construct mathematical models to predict an output given an input signal from a training dataset. Neural networks present tools for learning these models. In what follows, we will contextualize automatic learning in the context of computer vision.

2.3.1.1 Supervised learning

The most used type of automatic learning is the supervised learning, which allows the machine to learn its parameters using annotated datasets. For example, in an image classification framework, a model driven by a supervised learning predicts the type of object (its class) in an input image. In computer vision, the available datasets are divided on training and testing sets. During the learning, each image of the training dataset is presented to the model, which will update its parameters to produce the desired output. The update of these parameters is carried out using the notion of risk minimization: When a learning example is presented to the model, the output predicted by the model is compared with the desired output. The error between the desired output and the predicted output is then computed. The aim of supervised learning is to find the model parameters that minimize this error in all the learning dataset examples. In the test phase, a learnt model enables predicting the output associated with an image that it did not see during the learning phase. This is called the generalization of the model.

Supervised learning in computer vision is generally divided into two stages. The first one is the extraction of visual characteristics from the images of the learning dataset. The purpose of extracting the characteristics is to provide a discriminating

description in a reduced space compared to the space of the image which is too large. The learning of linear classifiers on visual characteristic vectors is a well formulated and solvable problem, for instance with the SVM [Cortes 1995]. Figure 2.6 illustrates schematically the classical process of supervised learning. Once the model is learnt, it can be used on a new image whose class is unknown. The vector of visual characteristics is extracted on this image and the model predicts its class.

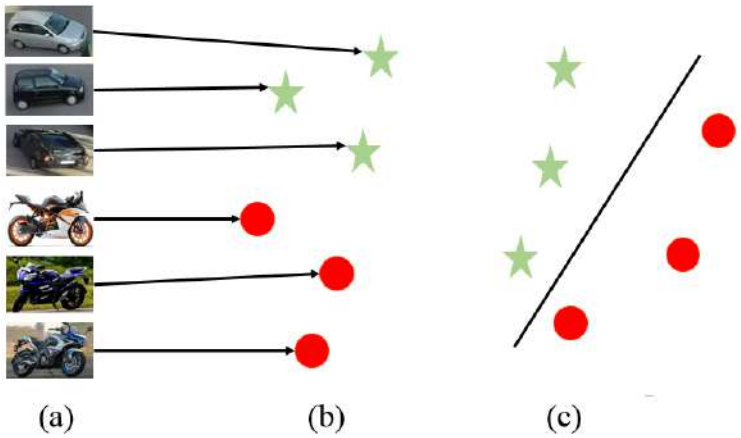


Figure 2.6: Classical schema of supervised learning for object classification. (a) A set of images corresponding to two classes (bike and car). (b) The characteristics extracted from these images. The characteristic vector of each image is here schematized by a 2D point for the legibility of the diagram but is generally of greater dimension. (c) The separation learnt using a linear classifier to discriminate the two classes. Source [Malisiewicz 2011]

There are many types of visual characteristics such as the HOG [Dalal 2005], the Scale-Invariant Features Transform (SIFT) [Lowe 1999], Local Binary Patterns (LBPs) [Ojala 1996] or the Haar characteristics [Viola 2001b]. These feature extraction approaches make it possible to extract low-level characteristics, based on primitives of an image such as gradients or contours. These extraction algorithms are completely external to the learning of a classifier and are computed beforehand on the images.

2.3.1.2 Differentiation of neural networks

Neural networks present a part of the automatic learning tools and are especially used for supervised learning. They are part of a logic that is slightly-different from "classical" learning approaches. Indeed, we previously saw that the learning process includes a step of extracting visual characteristics from the image. This step was mostly based on image-processing-oriented algorithms. Although very relevant to certain vision tasks, these characteristics may prove ineffective depending on the nature of the problem to be solved. These lead us to ask several questions. What really characterizes objects ? Are they contours ? or colors ? How can we succeed in

finding a reduced representation of an image that is as relevant and discriminatory as possible ?

It is on this point that neural networks mark a technological break. Indeed, rather than extracting visual characteristics "manually", neural networks allow the directly use of an input image and learning which visual characteristics are the most relevant for a given problem. Figure 2.7 depicts this difference. In contrast to the HOG [Dalal 2005] or SIFT [Lowe 1999] characteristics, which are low-level image representations, the neural networks permit, by a succession of non-linear functions, the extraction of increasingly high-level characteristics. It is this succession of functions that gives the name of *Deep learning* to approaches based on modern neural networks. The more functions, the deeper the network and the high level of the extracted visual characteristics.

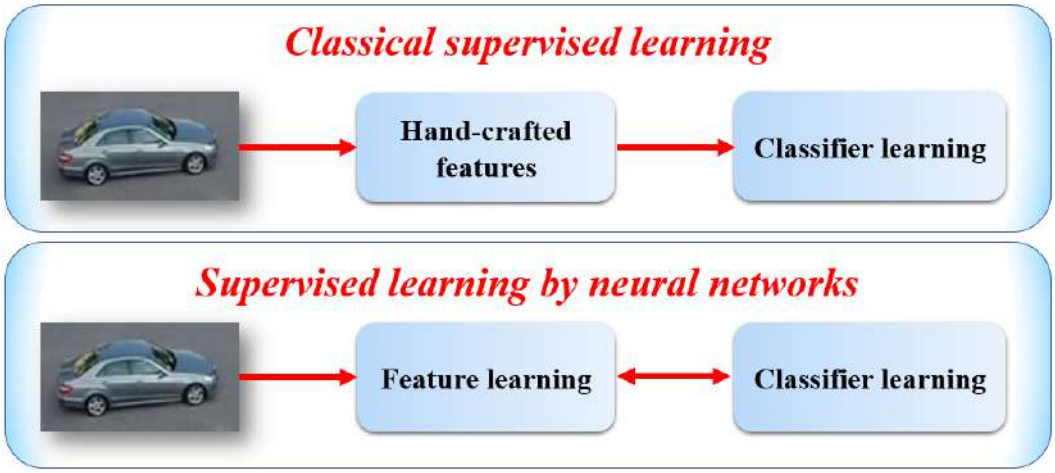


Figure 2.7: Interest of neural networks for visual feature learning in comparison with classical supervised learning approaches (translated from a presentation of Yann LeCun 2016, Paris).

2.3.2 Neural networks: Basic concept

In this section, we propose to expose basic concept of the neural networks. We first present the basic entity of neural network, the formal neuron. Then we present the Multi-Layer Perceptron (MLP).

2.3.2.1 The formal neuron:

Formal neuron (or artificial), initially introduced in [McCulloch 1943], is a mathematical modeling of the biological neuron. It consists of a mathematical function to be applied to a signal and return an activation value. Considering an input signal $X = \{x\}_{n=1,..,N}$, the artificial neuron returns the activation value y. The latter

value is computed as follows, equation (2.1):

$$y = f(b + \sum_{n=1}^{N} w_k x_k) \tag{2.1}$$

In this formulation, $\mathbf{w}_i$ are commonly called weights and b is called bias. The function f is called an activation function. In the initial formal neuron, this function is the signal function, hence returning a binary value at the output of the neuron. The weights, the bias and the activation function characterize the formal neuron. Figure 2.8 illustrates how this neuron works.

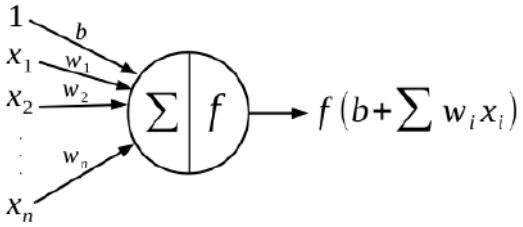


Figure 2.8: The artificial neuron.

2.3.2.2 Multi-layer perceptron

The neuron has two disadvantages. It expresses only a linear relation between the inputs and its output. Moreover, its power of expression is limited since it produces only one output. The MLP was invented to overcome these limitations. First, the activation function has been modified to include nonlinearity. A popular choice of this function is the sigmoid one as it is an approximation of the derivable Heaviside function which is an essential element during learning. The second contribution of the MLP consists in connecting several neurons together as layers.

An MLP consists of three types of layers:

- An input layer that corresponds to the input data x = [x_1, ..., x_n].
- An output layer consisting of m neurons and producing the outputs of the network y = [y_1, ..., y_m], that is to say the output values associated with the input data x.
- Hidden layers each consisting of several neurons. These layers allow the non-linear transformation of the input signal to the output one.

In the framework of the MLP, all the neurons of a layer are connected to the neurons of the previous layer. Figure 2.9 illustrates an MLP consisting of two hidden layers.

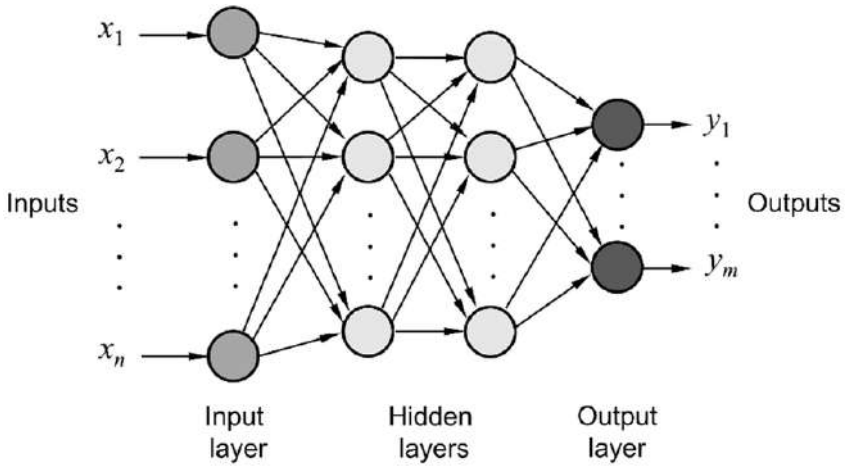


Figure 2.9: An example of an MLP consisting of two hidden layers. Each circle represents a formal neuron.

2.3.3 Deep convolutional neural networks

CNNs are a special type of neural networks that can be easily applied to images in order to extract and classify information spatially. The first CNN was introduced in the late 1980s by LeCun *et al.* [LeCun 1989] for image recognition. This network allowed the recognition of handwritten digits.

The idea is to pass an input image in a succession of convolutional filters (as shown in Figure 2.10) providing a reduced and relevant description of an image.

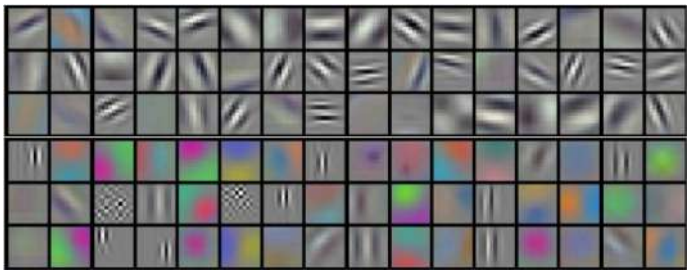


Figure 2.10: Examples of convolution filters: 96 filters of the first layer of AlexNet architecture [Kataoka 2015].

These characteristics are then sent to an MLP composed of hidden layers and fully connected output ones permitting classification of digits presented in the image. Convolution filters and fully connected layers are learned simultaneously. Figure 2.11 presents the architecture of a convolutional network.

Because of their convolutional structure, CNN makes it possible to take in input data of large dimension, which is a limit of the MLP.

For example, an image having three channels (RGB) of size 224×224 pixels represents an input vector of size $150,528$ for an MLP. This involves $150,528$ weights

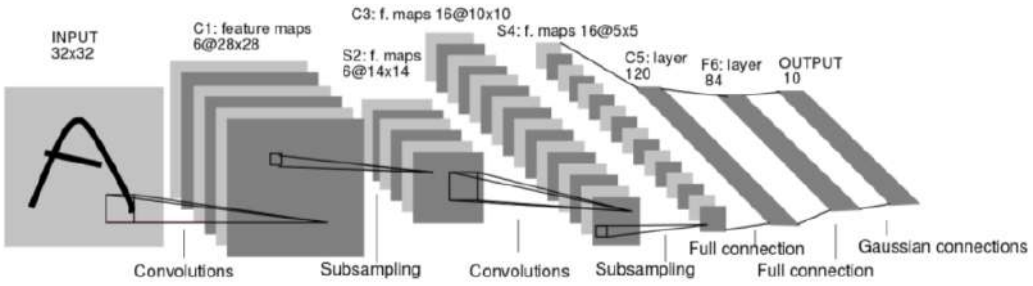


Figure 2.11: A convolutional network for the recognition of handwritten digits. Source [LeCun 1998]

to learn for each neuron of the hidden layer connected to the inputs, which is complicated to learn. The CNNs can be seen as a series assembly of modules allowing the extraction of characteristics from the pixels of an image in a hierarchical way.

2.3.3.1 Different modules of CNN

We present here the different modules used in CNNs: convolution, pooling, activation functions, dropout, batch normalization and standard error functions (loss functions) used for learning.

Convolution: The core of a CNN is the convolutional layer. The resulting output is called feature map. A convolutional layer is made up of several convolutional filters (or kernels) to be applied to each position of an input image.

Figure 2.12 shows how the convolution works. In practice, a value (bias) associated with the convolution filter is added to each position of the output of the filter. During learning, the values of the weights and biases of the neurons present the components of these filters that are learnt.

A filter of a convolutional layer is applied to all the positions of an input image, that is why we speak of shared weights.

Pooling: The pooling layer adds a spatial invariance when extracting features, reducing the size of inputs. It can be of different nature but the most used pooling types are *Max Pooling* (shown in Figure 2.13) and *Average pooling*. *Max Pooling* permits returning the maximum element of a computation window. *Average Pooling* allows returning the average of the elements on a computation window.

Activation functions: There are different activation functions allowing the non-linearity in the different CNN layers. The most famous of these functions are (as shown in Figure 2.14): The sigmoid function, the hyperbolic tangent function and the ReLU function.

The ReLU activation function is the most used in deep CNNs because it permits easier optimization. It has the advantage of providing sparse answers and makes it possible to reduce the problems of gradient disappearance. The ReLU function, for

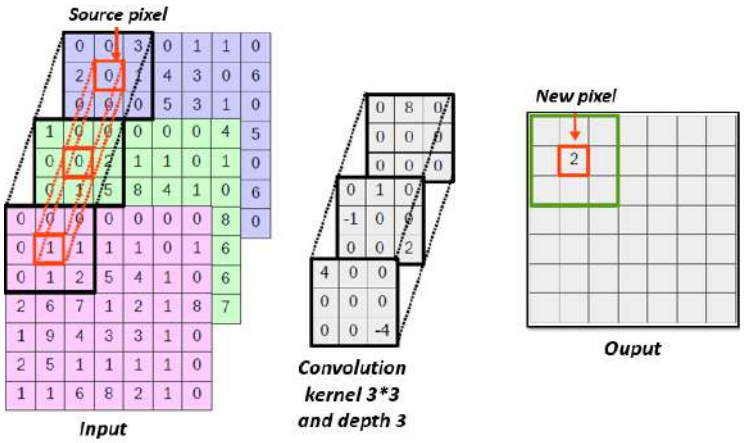


Figure 2.12: Convolution illustration. Given an input, a convolution filter (or convolution kernel) is applied for each position.The depth of the kernel depends on the depth of the input to which it is applied: In this example, the input has three channels, so the depth of the kernel is three. The result for a given position corresponds to the sum of the multiplication of the kernel elements by those of the input: In this example, $2 \times (-4) + 5 \times 2$. In the context of CNNs, the output of a convolution is called a feature map. The number of feature maps depends on the number of filters applied to the input. Source [Guerry 2017]

its part, refers to a constant gradient for a large input, enabling faster learning (in particular networks with a certain depth). There are other activation functions in the same family as ReLU such as LReLU [Maas 2013], PReLU [He 2015] and eLU [Clevert 2015].

Dropout: To avoid overfitting, the dropout layer was introduced in [Srivastava 2014]. This layer is used during learning. It allows randomly deactivate neurons during the different learning iterations. In other words, the dropout enables the network to learn subnets containing fewer parameters, hence, less overfitting subjects. This way permits learning more generic parameters that do not focus on the details of the learning dataset. Once the learning is complete, all neurons are reactivated.

Batch normalization: This technique was presented by Ioffe *et al.* [Ioffe 2015] in order to learn the CNNs more quickly and efficiently. It starts from the following observation: During learning, the distribution of the inputs of the different network layers changes at each iteration. This induces a permanent adaptation of CNN parameters to these different distributions, which increases the learning time. The idea of batch normalization is to normalize the inputs of each layer so that their distributions are of a zero mean and a unit variance. During learning, the batch normalization layers learn parameters (a scale factor and a bias) to adjust this normalization: These parameters enable applying a transformation on the normalized distribution.

Loss functions: There are several loss functions that can be used for learning

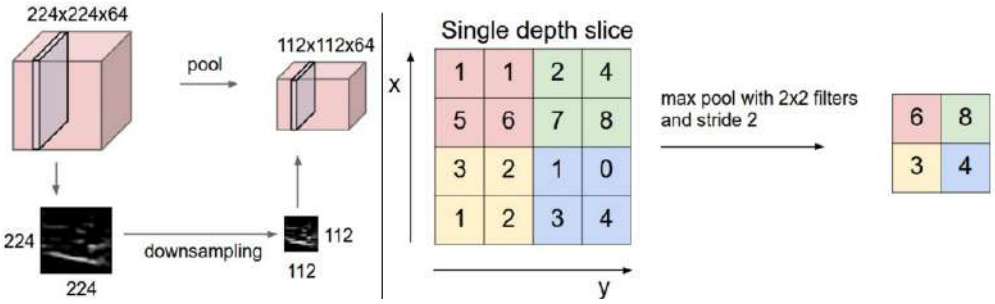


Figure 2.13: Illustration of *Max Pooling*. In this example, the pooling kernel is of size 2×2 and is applied every two pixels (stride = 2). The maximum of the four elements on a window of the input matrix is kept. Source [CS2]

.

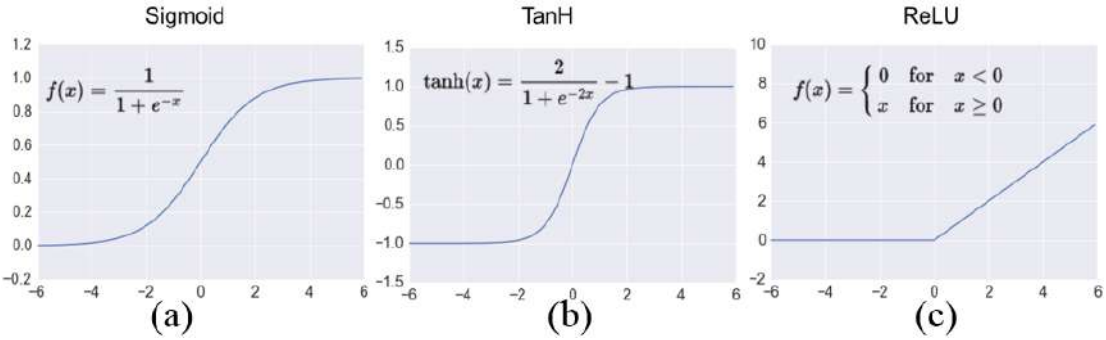


Figure 2.14: The most famous activation functions. (a) sigmoid, (b) hyperbolic tangent and (c) ReLU function. Source [CS2]

neural networks. These functions are dependent on the task that the network must perform (classification, regression, etc.). Here we list the most used loss functions.

- The Softmax loss function, commonly used for network optimization. It allows the maximization of the probability that an entry has to belong to one class rather than another.
- The loss function by sigmoid crossed entropy, allowing a regression on probabilities.
- The loss L1 smooth function, introduced in [Girshick 2015a], which permits a better optimization for regression problems.

2.3.3.2 Classical neural architectures

We present here the most popular architectures of the CNN used in computer vision research. It is important to note that the architectures proposed in the literature

have a strong tendency to become more and more deep with the years. In other words, it seems that the deeper the network is, the better the performances are (as shown in Figure 2.15). Nevertheless, this depth implies facing certain difficulties, particularly in terms of computation time and optimization during learning. This is why the community remains very active on the problem of designing CNN architectures.

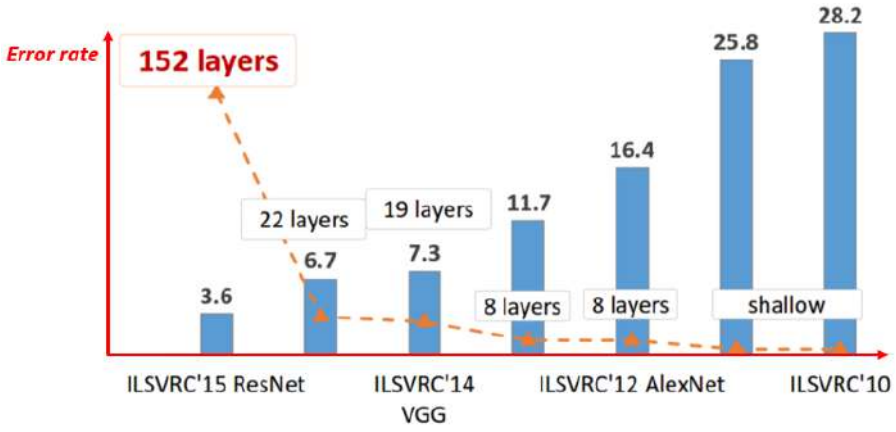


Figure 2.15: Error rate of different architectures on ImageNet for object classification. Over the years, the classification error has diminished, due to the increasing depth of architectures. Source [He 2016].

Standard architectures are widely used in vision for two main reasons. The first is that they allow for an easy comparison of CNN-based methods. In other words, although some work focuses on the study of neural architectures, the majority of vision methods reuse already learned CNNs and modify them to design new architectures that respond to particular tasks. The second reason is related to the difficulty of learning deep networks because of their large number of parameters. A common practice is the use of CNNs already learnt on huge databases and then adapt them to a specific task. This is called "fine-tuning". This practice enables learning deep networks using an initialization of the weights and bias already very relevant and generic. The adaptation of these parameters is then carried out during the phase of learning the specific task that it is desired to carry out. This result in a much faster learning speed and a virtually guaranteed convergence.

AlexNet: This architecture is the one proposed in [Krizhevsky 2012], which allows the resurgence of the study of neural networks from 2012, in particular thanks to the victory at the ImageNet image classification competition. This architecture uses five layers of convolution and three layers of pooling. The size of the convolution kernels is variable (11×11, 5×5, 3×3) as a function of the layer in question. The activation function used between each layer is the ReLU function. After passing the image in the convolution, pooling and activation layers, a feature map is obtained. This is sent in an MLP consisting of two hidden layers and one output layer. Figure 2.16 illustrates the architecture of AlexNet.

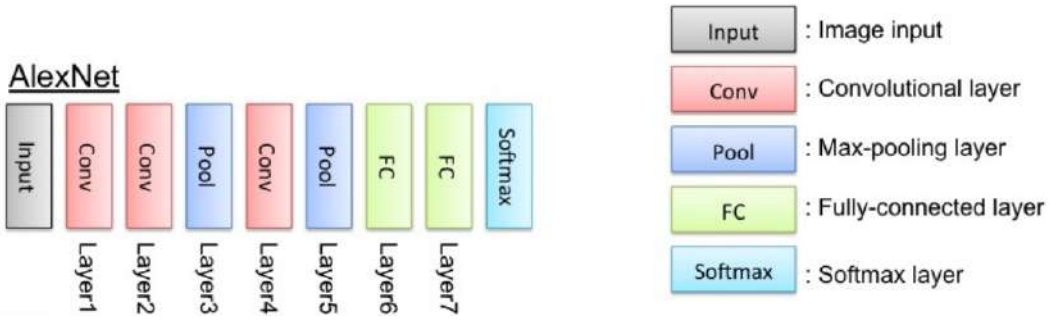


Figure 2.16: Illustration of AlexNet architecture [Kataoka 2015].

VGG: This CNN was introduced in [Simonyan 2014]. Instead of using a single convolution per depth level such as AlexNet, this architecture utilizes convolution sequences. In addition, VGG has convolutional filters with small size than in AlexNet (size 3×3). Figure 2.17 presents the architecture of the VGG.

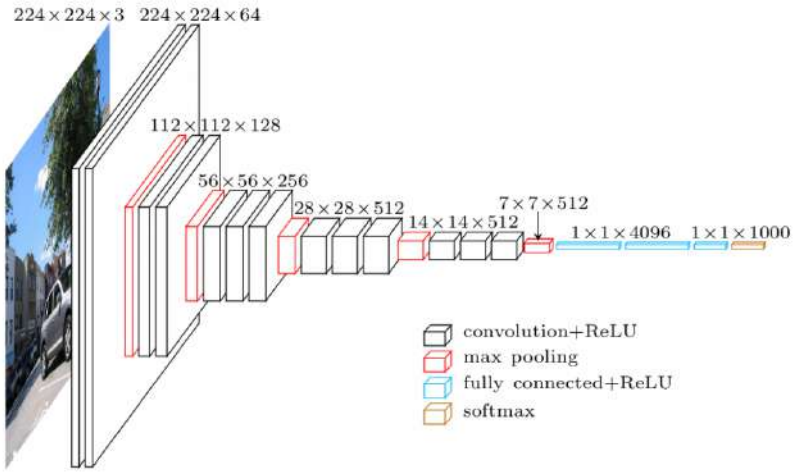


Figure 2.17: Illustration of the VGG architecture [Simonyan 2014].

ResNet: This CNN was presented in [He 2016]. It allows the learning of very deep networks (more than 150 layers). The difficulty in learning such deep networks is particularly related to the retropropagation of the gradient. The deeper the network is, the lower the gradient is for updating the weights of the lowest layers (the first layers). The idea developed in ResNet is the use of residual connections allowing better optimization of very deep networks. A residual connection makes it possible to pass the input in two convolution filters but also to pass this input directly to the following layers. This is done by summing the result of the two convolution layers and the input, as shown in Figure 2.18.

With this architecture, the authors demonstrate the interest of learning very

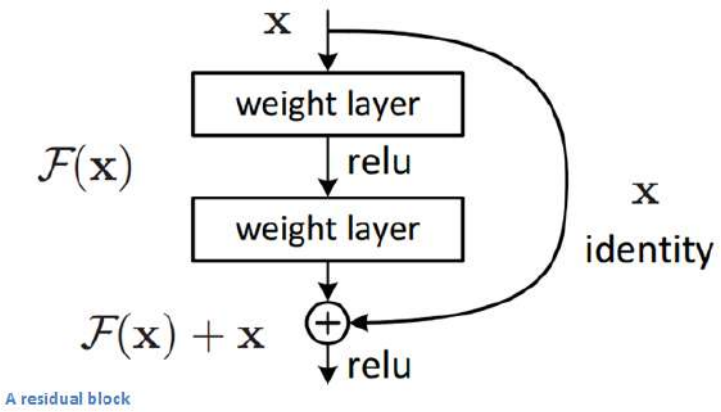


Figure 2.18: Residual connection. [He 2016].

deep networks and propose a way to learn them effectively. Figure 2.19 depicts the architecture of ResNet.

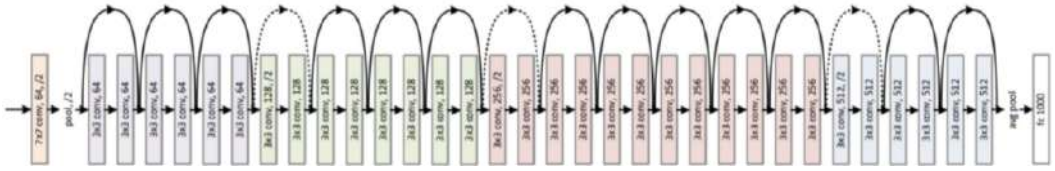


Figure 2.19: Illustration of ResNet architecture. Source [He 2016].

2.4 Convolutional neural networks for object detection

One of the most uses of CNNs is to localize and detect objects in images. This last problem, is a more challenging one, and requires to localize in an image several objects that can vary on size and pose. Some of the datasets challenges and competitions have contributed to the development of approaches and other CNNs architectures able to solve the object detection challenges.

As mentioned in section 2.2.1, initial approaches use well known sliding window and strategy in combination with CNNs-extracted features and a final classifier. However, Sermanet *et al.* provided in their approach [Sermanet 2013] that end-to-end trained CNN models can also be designed to generate an algorithm to object localization, detection and recognition. They utlized CNN special characteristics of location invariance and weights sharing to create an efficient sliding window approach. They also suggest an original approach that learns to predict object boundaries to create bounding boxes, all in the same CNN architecture.

Accordingly to section 2.2.2, the recent advancement of region proposal methods (e.g, [Uijlings 2013]) have suggested new CNNs architectures. An important one,

was put forward by Girshick *et al.* [Girshick 2014b], as what is known Region-based CNNs (R-CNNs). It starts with the *selective search* strategy that outputs 2000 proposals. Next, it uses a pre-trained AlexNet classification model to extract a 4096 feature vector for each of the regions. Finally, it classifies each region with the SVM and with the results they fine-tune the CNN for detection.

Further works on object detection with CNNs have focused mainly on reducing the computations of R-CNNs, which has been achieved successfully by sharing the convolutions across proposals [Girshick 2015a], [Ren 2015c], [Redmon 2016a]. Differently, work done by Shaoqing *et al.* in [Ren 2015c], directly proposes a fully convolutional network able to produce the region proposals by adding two convolutional layers to the network.

Some recent research is dedicated to unifying the two-staged approach from [Ren 2015c] into one stage, avoiding to resample features. Single Shot MultiBox Detector (SSD) [Liu 2016] uses the strategy of *anchors* from [Ren 2015c] proposal network and applies them to several feature maps of different resolution in the convolution network. This allows the detector to consider image regions at different sizes and different resolutions for detection objects at multiple scales.

Table 2.2 provides an overview of detection methods by CNN and their results on Pascal VOC 2007 [Everingham 2010].

Table 2.1: Summary of detection methods by CNN and results on Pascal VOC 2007 [Everingham 2010].

Method	frame per second	mAP
DSOD [Shen 2017]	17.4	77.7%
MobileNet-SSD [Howard 2017]	93	75.4%
R-CNN OHEM [Shrivastava 2016]	–	78.9%
Yolo v2 [Redmon 2016b]	67	76.8%
SSD [Liu 2016]	59	74.3%
Yolo [Redmon 2016a]	45	63.4%
Faster R-CNN [Ren 2015c]	17	73.2%
Fast R-CNN [Girshick 2015a]	10	70.0%
R-CNN [Girshick 2014b]	0.1	62%

2.5 Transfer learning

Several traditional automatic learning methods operate under a common assumption: The learning and test data come from the same feature space and share the same sample distribution. If the distribution is different, it will be necessary to resume learning from scratch while collecting new data from the target domain and handling the tasks differently. However, in many real-world applications, the collection of new data is costly and sometimes difficult [Pan 2010b].

Transfer learning aims to solve this situation by developing methods to transfer knowledge and skills learned in one or more source tasks in order to use it to improve a target task with the consideration of similarity links between these tasks. Figure 2.20 shows the difference between traditional learning and transfer learning. The traditional learning starts the learning process from scratch of new task independently from other tasks. Whereas, the transfer learning techniques use the knowledge acquired in previous tasks when learning a new one.

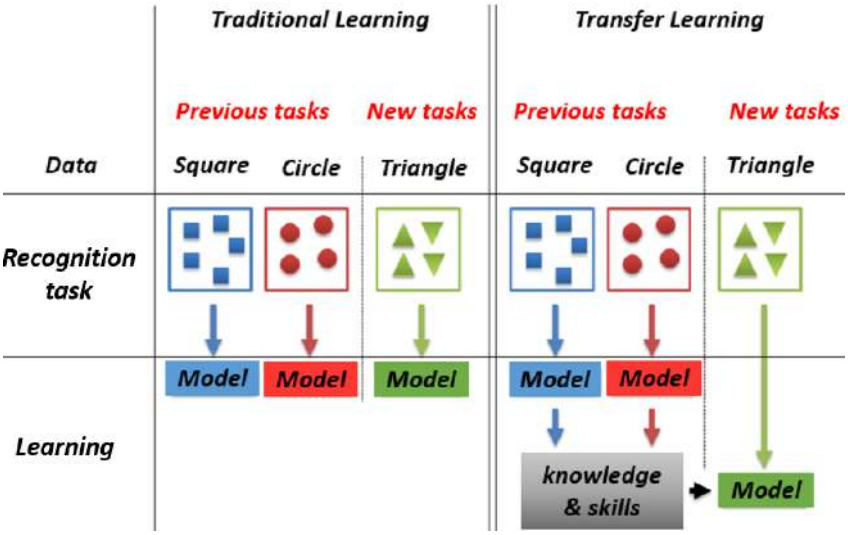


Figure 2.20: Differences between traditional learning and transfer learning [Pan 2010a]

The transfer process begins with a need to learn a target task in a target domain while knowing that a set of source tasks in a source domain is available and that there is a set of relationships created at the basis of similarity between the source and target problems. Defining the target task, available source tasks that can be beneficial to improve the learning of the target task and the existing similarities between the source tasks and the target ones are the answer to the question: "Where is knowledge transfer ?".

Once the source problems, the target problem and the similarity links are specified, the next step is to decide what to transfer ? How to transfer ? And when to transfer ?. The question "What to transfer ?" determines the type of knowledge to be transferred from the source domain to the target one. The question "How to transfer ?" determines the nature of the transfer, that is to say whether the transferred knowledge will be used as it is or whether it must undergo transformations to adapt to the new conditions. It also defines how to use this knowledge during the learning phase of the new task. The question "When to transfer ?" must assess the situation in which the transfer may be advantageous. This question seeks to avoid any case of negative transfer by determining the amount of transfer from the defined sources. If the source tasks are not similar to the target and/or if there is

a sufficient amount of data for target task learning, transfer can result in negative effects instead of improving the learning process [Aytar 2014].

2.5.1 Motivation of transfer learning

The aim of transfer learning is to improve the learning of a target task by bringing back the knowledge learned about other source tasks. The transfer learning has several advantages, such as avoiding the manual effort required to annotate a large amount of data. Figure 2.21 describes three levels of performance improvement of transfer learning (higher start, upper slope, higher asymptote) by comparison with the learning method of the target task without transfer.

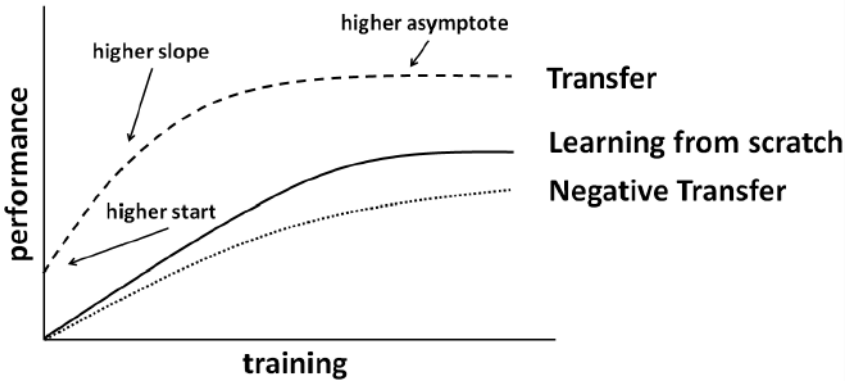


Figure 2.21: Transfer learning advantages [Tommasi 2013, Aytar 2014]

2.5.2 Different types of transfer learning

According to [Pan 2010a], there are three types of transfer learning based on the different relationships between the tasks and the source and target domains. Figure 2.22 summarizes the different types of transfer learning.

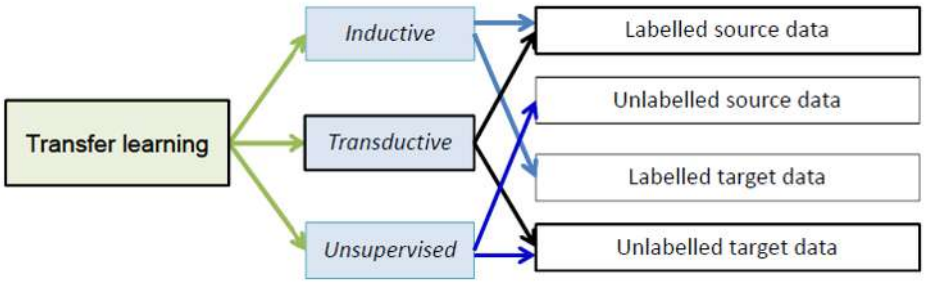


Figure 2.22: Different types of transfer learning. Translated from a presentation of Houda MAAMATOU [Maâmatou 2016a], Italy.

Inductive transfer learning: For this type of transfer, the source and target domains can be similar or not, but the tasks are different. Some annotated target

data are needed to produce a predicative model to be used in the target domain. There are two cases:

- Inductive transfer learning with annotated source data: There interest in transferring source knowledge is to achieve a high performance in the realization of the target task. This type of transfer is similar to the multi-task learning configuration, with the difference that multi-task learning learns both source and target tasks simultaneously [Pan 2010a].

- An inductive transfer learning without annotated source data: This is a configuration similar to a self-learning case as presented by Raina *et al.* [Raina 2007]. This is a situation where the source and target label spaces are different and the knowledge of the source domain cannot be used directly [Pan 2010a].

Transductive transfer learning: The transductive type deals with a target domain without any labelled data and assumes that the distribution of the source domain is different from the target one, though they are actually related. Two cases arise depending on the situation of the source and target domains:

- The characteristic spaces between the source and target domains are different.

- The characteristic spaces between the source and target domains are similar, but the distributions of marginal probabilities are different. In this type of transfer, we find the adaptation of a face detector to specific photos for a good manipulation of the new domain conditions [Jain 2011], we find also the transfer for text classification [Daume III 2006] and the adaptation of a generic pedestrian detector to a new scene [Wang 2011] [Maâmatou 2016d].

Unsupervised transfer learning: As for inductive transfer learning, the source and target tasks are different and the domains may be similar or not. However in this type of learning there is no data labelled either in the source domain or in the target one. The target task is often an unsupervised problem such as grouping, dimension reduction, or density estimation. As an example, we mention the work of Dai *et al.*, [Dai 2008] which presents an unsupervised transfer approach for grouping a set of data in the target domain by exploiting a large quantity of unlabeled data available in the source domain but learning a common feature space across domains.

In our work, we are interested on the transductive transfer learning type. The latter allows avoiding data labelling in each scene and offers improving object detection in different sequences. These were the reasons that motivate us to suggest an original formalization of transductive transfer learning in order to specialize a generic deep detector to a target domain.

The details of the transductive transfer learning methods are described in the following sections.

2.6 Categorization of transductive transfer learning methods

The existing transfer learning methods are categorized according to the knowledge transferred. These transfer learning categories will be described in the next subsections.

2.6.1 Transfer of example

These methods focus on the transfer of source examples that can be reused by solving the target task. Nevertheless, the source data may not be usable in their raw forms and may not all be useful, but some examples can reinforce the target learning process following a ponderation function. Despite the use of source and target examples, the transfer of examples learns only the target task. There are several methods of transfer of examples that are described for artificial intelligence and computer vision applications.

Haung *et al.* [Huang 2006] proposed a Kernel-Mean Matching (KMM) algorithm for the direct learning of the ratio of the source distribution to the target distribution by matching the two averages of the source and target data by producing a Hilbert kernel space. The main benefit of using the KMM is the ability to avoid the density estimation for both domains which can be difficult if the dataset is reduced.

Dai *et al.* [Dai 2007] had a "TrAdaBoost" extension of the basic doping algorithm "Adaboost". It reduces the weight of instances that are poorly predicted in order to reduce as much as possible their undesirable effect on the learning process. TrAdaBoost allows the construction of a good quality classification model while using data of different distributions and quantities: a reduced amount of labeled data from a target distribution that is generally insufficient for learning a good classifier and a large amount of data from another source distribution. At each iteration, if an instance is badly predicted then the algorithm reduces its learning weight to attenuate its effect at subsequent iterations. In this way, examples that are not similar to the new data affect the learning process from one iteration to another. However, older instances that are consistent with recent data help the algorithm better train the classifier.

Jiang and Zhai [Jiang 2007] proposed a heuristic method to minimize the difference in the conditional probabilities of the source and target domains by removing samples that would disrupt target learning. Duan [Jiang 2007] put forward a "Domain Transfer SVM (DTSVM)" approach for a video classification task. The DTSVM minimized the SVM structural risk function and the maximum average divergence. This was the criterion that identified the difference between the distribution of source and target samples by learning an optimized kernel function.

Sugiyama *et al.* [Sugiyama 2008] put forward an algorithm known as the Kullback-Leibler Importance Estimation Procedure (KLIEP) to directly estimate the ratio of source density according to target density based on the minimization of the Kullback-Leibler divergence. The KLIEP can be integrated into cross-validation

to automatically perform model selection in two steps: (1) estimating the weights of the source domain data, and (2) learning models with weighted data.

Wang *et al.* [Wang 2010] introduced a system of object classification at a maximum margin based on the assumption that an object model had to accurately respond to examples from similar source categories and had to respond negatively to non-similar ones. Lim *et al.* [Lim 2011] illustrated the performance improvement by borrowing and pondering a set of samples from several categories of objects visually close when learning a target object detector (sofa). Figure 2.23 depicts the principle of the suggested model. They looked for the right samples to transfer by associating a weight with each sample and for the right transformations to apply for each sample in order to increase the transfer flexibility. The transformed examples in the blue (or red) rectangles are similar in front view (or profile) of the sofa. Barred images will not be transferred for learning because they have low weights.

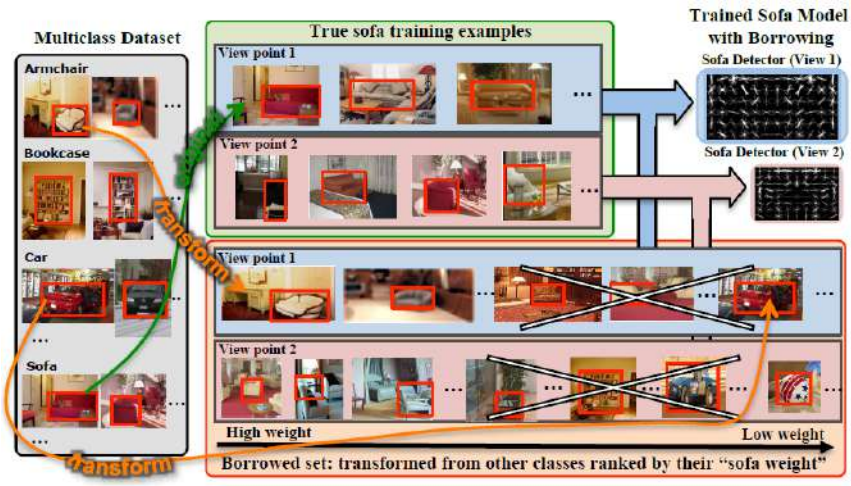


Figure 2.23: Illustrated transfer of examples: learning of a sofa detector by transfer samples from other visually related classes [Lim 2011].

2.6.2 Model transfer

Model transfer consists to transfer the parameters of the model trained on the source domain. It is based on the assumption that the linked task models must share a certain number of parameters as well as the distribution of hyperparameters.

Fei-Fei [Fei-Fei 2006] introduced a Bayesian transfer approach that uses the parameters of the source classifier and learnt the target model by updating the model parameters utilizing one or more examples of the target category.

Zweig and Weinshall [Zweig 2007] proposed a transfer learning approach that combines several object classifiers of different hierarchical levels into a single classifier using a configuration based on object category models. The aim of this method is to consider various aspects of an object.

Yang *et al.* [Yang 2007] presented an adaptive SVM approach for adapting classifiers to new domains. It is a variant of the transfer learning with a maximal margin, which takes advantage of the visual knowledge of source data or other forms of prior knowledge.

Gao *et al.* [Gao 2008] chose to combine a set of models instead of using a single model to transfer useful knowledge to the target domain. They proposed to: (i) approximate the model weights based on various structures in the target domain, (ii) provide an estimate based on the neighborhood of the graphs, and (iii) provide a prediction step to propagate labels from the nearest samples. Prediction is useful if the learning models are unable to provide an accurate response for some samples.

Stark *et al.* [Stark 2009] suggested a model based on a probabilistic form that allows the transfer of knowledge on three different levels: the shape and appearance of the parts, the local symmetry between the parts, and of the topology part. The model enables a partial or complete transfer of knowledge by transferring the parameters of the model.

Aytar and Zisserman [Aytar 2011] suggested transferring a model from a first category to a target category: Adapt a motorcycle model to a bike model and a horse model to a donkey model. In order to avoid learning the target model from scratch, they introduced the model of the pre-trained source category as a regulator of the cost function when learning the target category.

Gao *et al.* [Gao 2012] relied on the assumption that good detectors had to share some statistical properties. They took low-level statistics of the probability distributions of a set of random variables through the parameters of a source model. They proposed to strength these statistics by learning the model of the target task.

2.6.3 Feature transfer

This type of transfer is positioned between the example transfer and the model transfer. The goal is to find a good representation of the features that simplifies learning the solution of the target problem. A feature representation minimizes the domain divergence and the classification or regression model error.

In particular, when the problem of the target has very few examples and generalization is difficult, this type of transfer makes it possible to better guide the learning by limiting the space of research of the characteristics into the most significant characteristics. The transfer of the representation of the characteristics can be considered as a step of transition from a low level of characteristics to an average level. Raina *et al.* [Raina 2007] suggested applying a scattered coding method. It was a non-supervised feature construction method for learning high-level characteristics. The disadvantage of this latter was that it was based on high-level bias vectors. This bias were learned for the source domain and, which might not be adapted for the target domain.

Fink *et al.* [Fink 2005] presented a learning method for object classifier using a single sample. To do this, they gave a high weight to the relevant dimensions of the characteristics for classification using the available examples of related classes.

Quattoni [Quattoni 2008] proposed to learn a sparse prototype image representation from unlabeled data and related visual category data. This approach could exploit any function of the arbitrary kernel. The method was based on the joint approximation on the space of prototypes to find a subset of discriminating prototypes.

Saenko [Saenko 2010] introduced a method that adapted specific visual domains to new image conditions by applying transformations that would minimize the domain-induced changes in the distribution of features. Yao [Yao 2011] proposed to represent the image by the assigned responses of a set of classifiers that were trained in a supervised context. The content of each image was described using a set of basic actions. Figure 2.24 presents the Yao approach [Yao 2011] for the description of human actions.

Figure 2.24: Description of human actions: An action is represented as a weighted sum of a subset of attributes and basic action parts [Yao 2011]

In [Xue 2008], Xue *et al.* have put forward an algorithm called Topic Probabilistic Latent Semantic Analysis PLSA (TPLSA) to manage the problem of cross-domain text classification by allowing the transfer of knowledge acquired from documents from one domain to another one. The algorithm extended the PLSA algorithm to integrate labelled and unlabelled data from different but linked domains into a joint probability model.

Pan *et al.* [Pan 2008] exploited the method of incorporating Maximum Mean Discrepancy Embedding (MMDE), designated to reduce the dimension of the characteristics, to learn a space of small dimension to reduce the difference between both source and target distributions. However, this method suffers from a high degree of complexity. An improvement of the MMDE was suggested by Pan *et al.* [Pan 2011], which was an efficient extraction of features known as the transfer component analysis. A similar idea was proposed by Wang *et al.* [Wang 2008], which was summarized as a Discriminative Transfer Analysis (TDA). TDA was an algorithm that ran iteratively to find the best subspace for the target data. It applied clustering methods

in order to generate pseudo-class labels for unlabeled target data. Then, it applied dimension reduction methods to target data and annotated source data.

Douze *et al.* [Douze 2011] showed that image classification based on a high-level (attribute-based) representation enhances the image search task. Thus, they demonstrated that the combination of attributes would increased performance. Song *et al.*, [Song 2011] proposed to exploit, in an iterative way, the outputs of a task as high-level characteristics of another task in order to improve object classification and detection under new conditions.

2.7 Transfer learning applications for object detection

The literature has presented a lot of transfer learning applications for image classification and recognition in a target domain [Tommasi 2013],[Pan 2011]. Nevertheless, in this section we are interested in the works that deal with transfer learning for object detection. Particularly, object detection must take into consideration a significant number of challenges. We cite for example the alignment of learning images for detector training, considering a set of a priori samples to establish the correspondence between the source and target models. Thus, the different sizes of the interest object and various points of view between the source and target domains are taking into account.

Zhang *et al.* [Zhang 2008] put forward a method for adapting the classifier by combining the objective function of the source domain with that of the target domain. They approximated the learning term by a second-order Taylor expansion to reduce the amount of information needed for adaptation to pedestrian detection.

Aytar and Zisserman [Aytar 2011] suggested applying transfer between categories of similar objects. Figure 2.25 represents the principle of the method and illustrates the detection results with the obtained bike detector. They presented a modification of the objective learning function which retained the convexity and optimization of SVM methods and brought the benefit of learning a target model with a few samples. They presented two applications of the proposed method: an application for transfer between categories (e.g. transfer of a motorcycle detector to a bicycle detector and a transfer of a horse detector to a donkey one), and a second transfer application from a higher class to a subordinate class (transfer of a generic animal detector to a specific category such as horse, sheep and/or a cow).

Kuzborskij *et al.* [Kuzborskij 2013] extended the Projective Model Transfer SVM (PMT-SVM) method developed in [Aytar 2011] to apply the transfer of a multi-class problem to another multi-class one in an image classification framework. Donahue *et al.* [Donahue 2013] developed an extension of the PMT-SVM [Aytar 2011] for multi-class classification with data from several points of view and for object detection in videos. The elaborate extension integrated a Laplacian regularization which combined samples traditionally labeled with constraints coded on the unlabeled samples of the target domain.

Lim *et al.* [Lim 2011] and Gao *et al.* [Gao 2012] presented transfer learning

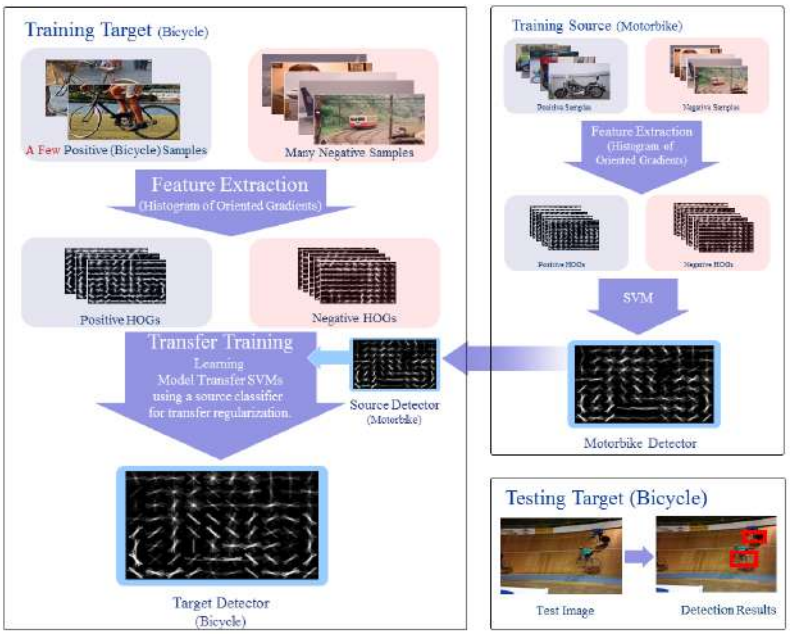


Figure 2.25: Transfer method of Aytar and Zisserman [Aytar 2011]. Learning a bike detector based on few bike samples and a motorcycle source detector.

applications between different object categories to improve the learning of a target category. Wang and Wang [Wang 2011] introduced a transfer learning method to specialize a generic pedestrian detector toward a target scene (more details are available in section 2 - chapter 3). Tang *et al.* [Tang 2012] used binary variables to weight examples that would be added or excluded from the learning set to adapt a bicycle, dog or car detector to static images in a detector of the same object in a video. Wang *et al.* [Wang 2012b] proposed a non-parametric method that utilized a vocabulary tree as a binary vector to encode a visual example for fitting a pedestrian detector to a video.

Table 2.2 summarizes different transductive transfer learning approaches applied on object detection.

2.8 Conclusion

In this chapter, we have started with a general view on two families of object detection methods. After that, we have presented the deep neural network. Then a discussion about transfer learning has been provided. In the fifth section, we have drawn up a state-of-the-art on the categorization of transfer learning methods In the last section, we have given an overview on some transfer learning applications for the object detection category.

In the next chapter, we will present the proposed formalization of transfer learning based on the theory of a sequential Monte Carlo filter so as to automatically generate a specialized deep detector for multi-object detection.

Table 2.2: Summarization of different transfer learning approaches applied on object detection applications.

Method	Description	Drawbacks
Ye *et al.* [Ye 2017]	- Pedestrian detection - Scene-specific pedestrian detection without any annotated training sample - Can work in any day/night conditions - Robust to occlusions, low resolution, and appearance variation - Self-learning approach	- Limited for pedestrian detection - Model performances sensitive to applications - Not label sufficient positive sample for training - Need many iterations for the convergence of the specialization process
Maamatou [Maâmatou 2016b]	- Pedestrian detection - Transductive transfer learning approach based on Monte Carlo filter - Transfer source and target samples - Automatic specialization without human intervention - Achieve good performance	- Based on hard-thresholding rules - Limited for single object detection - Selection samples based on background subtraction algorithm
Mao *et al.* [Mao 2015]	- Pedestrian detection - Transfer only target samples - Selection of positive samples based on tracklet algorithm - Iterative training process - Effectiveness to adapt classifier to specific scenes without human annotations	- Risk of drifting during iterations - Sensitive to applications - Classical features to associate tracklets - Based on hard-thresholding rules
Xudong *et al.* [Xie 2015]	- Vehicle detection - Transfer source and target samples - Transfer learning for CNN classifier	- Manual annotation of target samples - Model performances sensitive to object structure and point view of objects - Not totally automatic and needs some manual annotations

Chapter 3

SMC Faster R-CNN: Toward a Scene-Specialized Multi-Object Detector

Contents

In this chapter, we present the proposed formalism of transfer learning based on the theory of a Sequential Monte Carlo (SMC) filter to automatically specialize a scene-specific Faster R-CNN detector. The suggested framework uses different strategies based on the SMC filter steps to approximate iteratively the target distribution as a set of samples in order to specialize the Faster R-CNN detector towards a target scene. Moreover, we put forward a likelihood function that combines spatio-temporal information extracted from the target video sequence and the confidence-score given by the output layer of the Faster R-CNN, to favor the selection of target samples associated with the right labels.

This work was published at Computer Vision and Image Understanding "CVIU" journal. We present the best specialization framework on several public datasets. Compared with the state-of-the-art specialization frameworks, the proposed framework presents encouraging results for both single and multi-object detections.

This chapter is organized as follows. Section 3.1 presents an introduction to our work. Section 3.2 presents our contributions. After that, a detailed description of our approach is provided in section 3.3. The experiments and results are described in section 3.4. Section 3.5 provides a discussion about the advantages of our work over the state-of-the-art specialization frameworks. Finally, the chapter conclusion is given in section 3.6.

3.1 Introduction

Most state-of-the-art researches have been recently made to iteratively develop a scene-specific detector, whose training process is aided by generic detectors for automatically collecting training samples from target scenes without manually labelling them [Benfold 2011][Wang 2014b][Htike 2014][Maâmatou 2016c]. An ideal framework can apply a generic detector on some frames in a target scene, score each detection using some heuristics and then include the most confident positive and negative detections to the original dataset for retraining [Wang 2012a][Rosenberg 2005][Levin 2003]. Rosenberg *et al.* [Rosenberg 2005] opted for a self-training framework based on background subtraction to label scene samples. Only the samples with high confidence scores were added in a new training dataset from one iteration to another. Contrarily, there was a risk of introducing a wrong labelled examples in the training dataset, which may degrade the framework performance over iterations. In addition, Wang *et al.* [Wang 2014b] utilized different contextual cues such as visual appearances of objects, motion of pedestrian, model of road, size and location to select positive and negative samples from the target scene and to add the last ones in the training dataset for retraining. This approach proved to be sensitive to the risk of drifting and it can be applied only onto a particular classifier.

Moreover, some solutions collected the training source dataset with new samples extracted from the target scene, which increased the time of training and the size of the dataset during iterations [Aytar 2011][Quanz 2012]. Others were limited only to the use of samples extracted from the target domain [All 2011][Mao 2015], which caused the loss of useful samples stored in the source dataset. Htike *et al.* [Htike 2014] presented an approach that used only target samples labeled by a background subtraction algorithm and verified by the tracklet method to train a specific detector. In the same vein, Mao and Yin [Mao 2015] used tracklet chains to automatically label target information. They associated the proposal samples predicted by an appearance-object detector into tracklets and they propagated labels to uncertain tracklets based on a comparison between their features and those of labeled tracklets. This framework used many manual parameters and several thresholding rules for every target scene, which can affect the specialization performance.

Other solutions were proposed in [Li 2015b][Maâmatou 2016c][Mhalla 2016b], which collected new samples from the target scene and the source dataset. Maamatou *et al.* [Maâmatou 2016c] suggested a transfer learning method based on the

SMC filter to iteratively build a new specialized dataset that was used to train a new specialized pedestrian detector. This produced dataset consisted of both source and target samples that were utilized to estimate the unknown target distribution. Our proposed framework is inspired from this latter.

Addressing the transfer learning with deep learning has gained a growing attention. Some deep models have been investigated in the unsupervised and transfer learning challenge [Zeng 2014][Li 2015b]. Transfer learning using deep models has been turned out to be effective in some challenges [Mesnil 2012][Goodfellow 2012] like traffic-object detection [Zeng 2014][Li 2015b], emotion recognition [Ng 2015] and sentiment analysis [Glorot 2011]. In order to take advantage of these types of detectors, several transfer learning methods have been proposed to specialize a CNN detector by fine-tuning an ImageNet-pre-trained model with a small target dataset. Li *et al.* [Li 2015b] suggested adapting a generic CNN vehicle detector to a target scene by appropriating the shared filters between source and target data and updating the non-shared filters. In contrary to [Li 2015b][Oquab 2014], which needed several manual labeling of data in the target scene, Zeng *et al.* [Zeng 2014] proposed to use Wang's approach [Wang 2014b] to select target samples and utilized these latter as an input to their CNN deep model to re-weight samples from target and source domains without manually labeling data from the target scene.

Accordingly, we propose a new formalization of transfer learning based on SMC filter [Smith 2013] so as to automatically generate a specialized Faster R-CNN detector [Ren 2015c] for multi-traffic object detection, enhancing performances better than the generic one.

A global synoptic of our framework is illustrated in Figure 3.1.(a). We have a generic Faster R-CNN detector which is fine-tuned by a source labelled dataset with labeled information given in the form of traffic-object annotations. Given a target video sequence where labeled information is not available, an iterative process estimates both the set of target objects and the parameters of the specialized Faster R-CNN detector. This latter is automatically and iteratively trained and is called until a stopping criterion is reached. Then a final specialized Faster R-CNN detector is produced.

3.2 Contributions

The main contribution of this chapter consists in putting forward a new transfer learning framework based on the formalism and the theory of the SMC filter for deep detector specialization. The aim of our formalization is to automatically label the target data, to favor the selection of the target samples associated with the right label and to fine-tune a scene specialized Faster R-CNN detector.

Although the use of the SMC filter for transfer learning is obviously not new, our work extends the SMC framework for deep detector and for multi-traffic object detection. Moreover, we propose new strategies for transfer learning inspired from the three steps of the SMC filter :

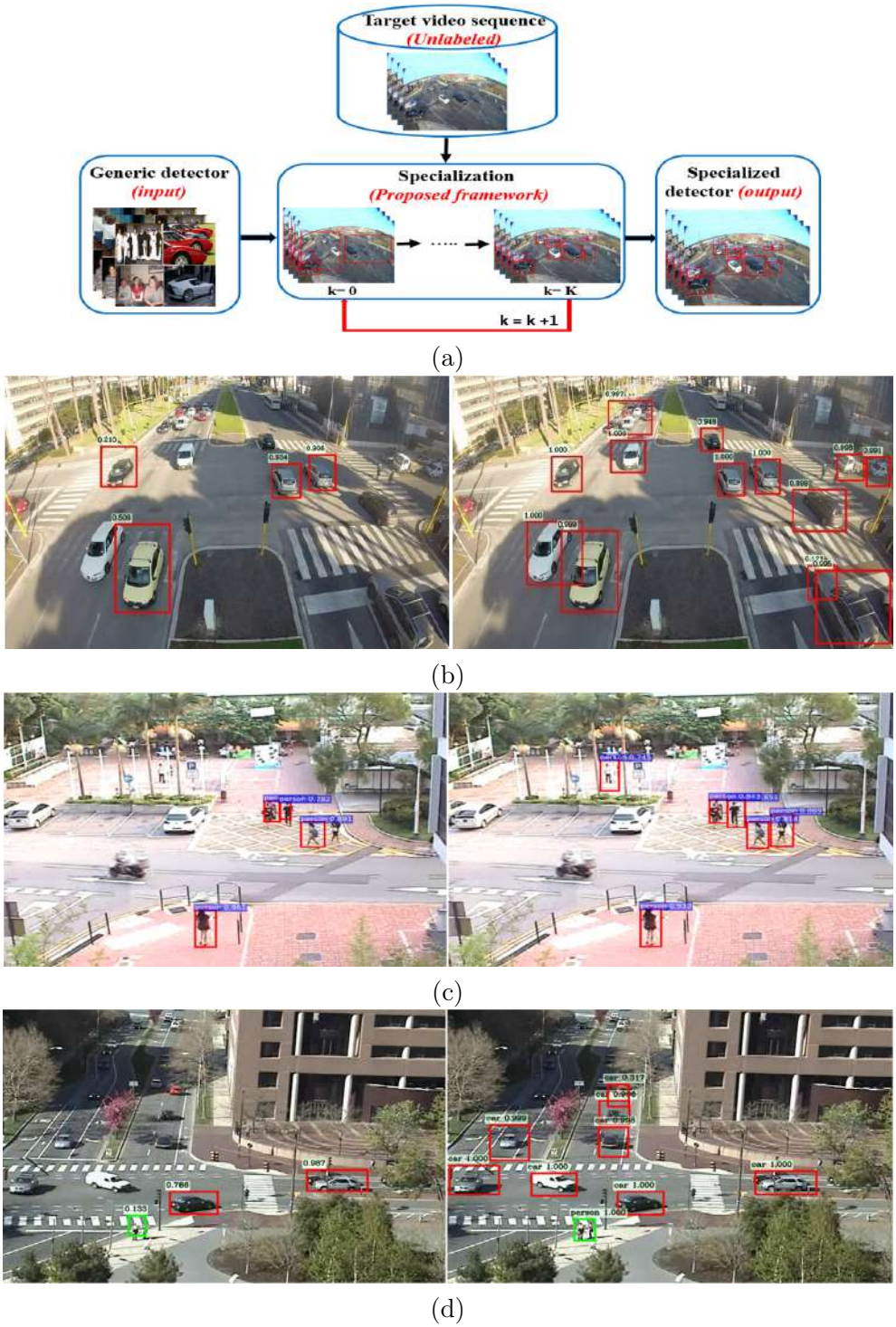


Figure 3.1: (a) General synoptic of the proposed framework. The input of the framework is a generic Faster R-CNN detector fine-tuned on a generic dataset, then given a target video sequence without any label information, an iterative process automatically estimates both the set of target objects and the parameters to specialize the Faster R-CNN deep detector; (b) and (c) improvement of specialized scene-specific detector over generic detector for single-class and (d) multi-class object detection (left images are generic Faster R-CNN detections and right images are specialized Faster R-CNN ones).

(1) Strategy of bounding box proposals: In order to use target samples for training a scene-specialized detector, the first strategy of the algorithm is to propose bounding boxes of traffic-object candidates by adapting the architecture of the Faster R-CNN deep network for only traffic-object detection. This strategy gives a set of suggestions composed by traffic proposals predicted by the output layers of the Faster R-CNN.

(2) Strategy of verification: We suggest a verification strategy to correctly select unlabeled samples from a target scene. This strategy utilizes a combination between the confidence-scores returned by the output layer of the Faster R-CNN and the visual context cues extracted from the target video sequence, in order to favor the selection of positive samples from a target scene and to reduce the risk of introducing wrong labelled examples in the training dataset.

(3) Strategy of sampling: We suggest a sampling strategy that collects useful samples from target datasets according to their weights importance, reflecting the likelihood that they belong to the target distribution. The main role of this strategy is to build the specialized dataset with samples produced by the strategy of verification. To do this, we use the Importance Sampling (IR) algorithm inspired from the theory of the SMC filter [Doucet 2001]. This algorithm transforms the weight on a number of repetitions, through repeating the samples associated to a high weight by numerous ones and repeating the samples associated to a low weight by few ones. This strategy makes the suggested framework applicable to specialize any detector and avoids the distortion of the specialized dataset, while selecting training samples according to the importance of their weights without modifying the training function.

Another contribution is to make a comparative evaluation of the proposed framework to the state-of-the-art specialization frameworks on several public datasets and with new more challenging annotations.

In the following section, we provides a detailed description of our specialization framework.

3.3 Proposed specialization framework

In this section, we present the proposed framework for specializing the Faster R-CNN model to a target scene based on SMC filter steps. Figure 5.4 shows the block diagram representation corresponding to one iteration of our suggested SMC Faster R-CNN. First, a generic Faster R-CNN network $(\mathcal{R}_0, \mathcal{F}_0)$ is fine-tuned on a generic dataset (eg: PASCAL VOC). Given the videos taken by a stationary camera in target scenes, at a first iteration $(k = 1)$, the generic detector $(\mathcal{R}_0, \mathcal{F}_0)$ is applied in the prediction step by using the strategy of bounding box proposals to suggest a set of traffic-object proposals in each individual image. Then an update step based on the likelihood function is used to favor the selection of the positive samples from a target scene by associating weight to each proposal sample returned by the prediction step. By utilizing the sampling strategy, the sampling step determines which samples

should be included in the specialized dataset according to their weights. A new specialized detector $(\mathcal{R}_k, \mathcal{F}_k)$ is trained by using the training strategy in the fine-tuning step. This specialized one will become the input of the prediction step in the next iteration. The scene-specific detector is automatically and iteratively trained and is called until reaching a stopping criterion, for example a fixed number of iterations. When the number of iterations is reached, a final specialized detector $(\mathcal{R}_K, \mathcal{F}_K)$ will be generated.

In what follows, we first describe the specialization of the Faster R-CNN model based on the theory of the SMC filter.

3.3.1 Faster R-CNN specialization based on SMC filter

Given a source dataset, from which a generic Faster R-CNN detector can be trained from this source dataset, and a video sequence of a target scene, then a specialized Faster R-CNN detector will be generated. This latter is the output of the distribution approximation provided by the formalism of the SMC filter and the fine-tuning step. To do this, let us define:

- $\mathcal{I}_t \doteq \{\mathtt{I}^{(i)}\}_{i=1}^{I_i}$ is a set of unlabelled images extracted uniformly from a video sequence of a target scene.

- $\mathcal{D}_k \doteq \{\mathbf{x}_k^{(n)}\}_{n=1}^{N_k}$ is a specialized dataset at iteration k, where $\mathbf{x}_k^{(n)}$ is a target object sample to be detected in each target image of the set $\{\mathtt{I}^{(i)}\}_{i=1}^{I}$. This sample is defined by: $\mathbf{x}_k^{(n)} \doteq \{\mathbf{p}_k^{(n)}, y_k^{(n)}, s_k^{(n)}\}$ where $\mathbf{p}_k^{(n)} \doteq \{u_k^{(n)}, v_k^{(n)}, w_k^{(n)}, h_k^{(n)}\}$ is the position of an object, with $(u_k^{(n)}, v_k^{(n)})$ being the upper left coordinates of the object bounding box and $(w_k^{(n)}, h_k^{(n)})$ being the width and the height of the object bounding box, $y_k^{(n)}$ is the object class label and $s_k^{(n)}$ is an associated score.

- $\{\mathbf{x}^{(n)}\}_{n=1}^{N} = \Theta(\{\mathtt{I}^{(i)}\}_{i=1}^{I_i}; \mathcal{R}, \mathcal{F})$ is a function that applies the Faster R-CNN detector using the RPN network model $\mathcal{R}$ for the localization task and the Fast R-CNN network model $\mathcal{F}$ for detection. For both localization and detection, a set of candidate objects with associated scores is provided.

- $\{\tilde{\mathcal{R}}, \tilde{\mathcal{F}}\} = f(\{\mathtt{I}^{(i)}\}_{i=1}^{I_i}, \{\mathbf{x}^{(n)}\}_{n=1}^{N}; \mathcal{R}, \mathcal{F})$ is a fine-tuning function that returns the new parameters $\tilde{\mathcal{R}}$ and $\tilde{\mathcal{F}}$ of the Faster R-CNN network. The fine-tuning is performed from the Faster R-CNN network with initial $\mathcal{R}$ parameters for the RPN and initial $\mathcal{F}$ parameters for the Fast R-CNN, utilizing a training dataset given by the set of images $\{\mathtt{I}^{(i)}\}_{i=1}^{I_i}$ and the associated objects $\{\mathbf{x}^{(n)}\}_{n=1}^{N}$

We define $\mathbf{x}_k$ to be a hidden random state vector associated to a joint distribution between labels and features of dataset samples at an iteration k and $\mathbf{z}_k$ a random measure vector associated to information extracted from the target scene (i.e. visual spatio-temporal information). Based on our assumption, the target distribution can

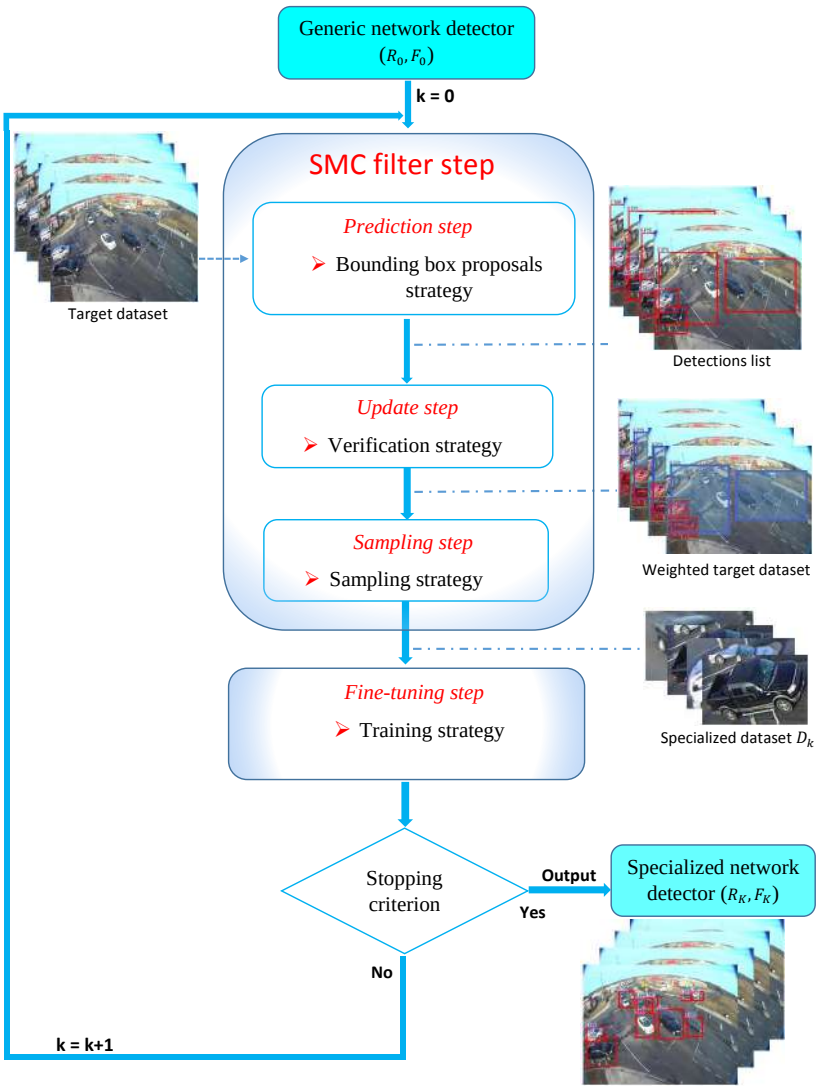


Figure 3.2: Block diagram of proposed approach: At the first iteration, our generic detector $(\mathcal{R}_0, \mathcal{F}_0)$ which is fine-tuned by the source dataset is utilized in the first prediction step by using bounding box proposals strategy to produce a list of traffic-object bounding boxes from the target scene, and then an update step based on the likelihood function is used to favor the selection of positive samples from a target scene. The sampling step determines which samples will be included in the specialized dataset by using the sampling strategy. A new specialized detector $(\mathcal{R}_k, \mathcal{F}_k)$ is fine-tuned by utilized training strategy in the fine-tuning step, which will become the input of the prediction step in the next iteration $k = k + 1$. A final specialized detector $(\mathcal{R}_K, \mathcal{F}_K)$ is called when a predefined number of iterations is reached. The red rectangles in the output image of update step mean that samples have a high weights attributed by our suggested likelihood function and a blue ones mean that samples have a low weights.

be approximated by iteratively applying equation (3.1):

$$p(\mathbf{x}_k|\mathbf{z}_{0:k}) = C.p(\mathbf{z}_k|\mathbf{x}_k) \int_{\mathbf{x}_{k-1}} p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{0:k-1}) d\mathbf{x}_{k-1} \tag{3.1}$$

where C is a normalisation factor: $C = 1/p(\mathbf{Z}_k|\mathbf{Z}_{0:k})$.

The SMC filter estimates the probability distribution $p(\mathbf{x}_k|\mathbf{z}_k)$ by a set of N particles (samples in this case), according to equation (3.2):

$$p(\mathbf{x}_k|\mathbf{z}_k) \approx \sum_{n=1}^{N} \pi_k^{(n)} \delta_{\mathbf{x}_k^{(n)}}(\mathbf{x}_k) \tag{3.2}$$

- δ represents the Dirac function (3.3):

$$\delta_{\mathbf{x}_k^{(n)}}(\mathbf{x}_k) = \begin{cases} 1 & \text{if} \quad \mathbf{x}_k = \mathbf{x}_k^{(n)} \\ 0 & \quad otherwise \end{cases} \tag{3.3}$$

- $\pi_k^{(n)} \in [0, 1]$ is the weight associated to sample n at iteration k and N is the number of target samples (3.4):

$$\pi_k^n = \frac{\pi_{k-1}^n \; p(\mathbf{z}_k|\mathbf{x}_k = \mathbf{x}_k^n)}{\sum_{n=1}^{N} \pi_{k-1}^n \; p(\mathbf{z}_k|\mathbf{x}_k = \mathbf{x}_k^n)} \tag{3.4}$$

  It is important to note that the sum of the weights of all the samples is equal to (3.5):

$$\sum_{n=1}^{N} \pi_k^{(n)} = 1 \tag{3.5}$$

All notations mentioned above are introduced in [Smith 2013].

Therefore, the formalism of the SMC filter is used to approximate the unknown joint distribution of traffic objects by a set of samples that are initially unknown. We suppose that the iterative process selects relevant samples for the specialized dataset from one iteration to another, leading to converge to the right target distribution, and making the resulting Faster R-CNN detector more and more efficient.

The resolution of equation (3.1) is divided into three steps: prediction, update and sampling. These steps are similar to the popular particle filter framework, widely used to solve the tracking problems in computer vision [Mei 2011][Smal 2007]. The details of the three main steps are described in the following subsections.

3.3.1.1 Prediction step

The prediction step consists in applying the Chapman-Kolmogorov equation (3.6):

$$p(\mathbf{x}_k|\mathbf{z}_{0:k-1}) = \int_{\mathbf{x}_{k-1}} p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{0:k-1}) d\mathbf{x}_{k-1} \tag{3.6}$$

Equation (3.6) uses the system dynamics term $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ between two iterations in order to suggest a specialized dataset $\mathcal{D}_k \doteq \{\mathbf{x}_k^{(n)}\}_{n=1}^{N_k}$ producing the approximation (3.7):

$$p(\mathbf{x}_k|\mathbf{z}_{0:k-1}) \approx \{\tilde{\mathbf{x}}_k^{(n)}\}_{n=1}^{\tilde{N}_k} \tag{3.7}$$

We suggest to extract the proposal set $\{\tilde{\mathbf{x}}_k^{(n)}\}_{n=1}^{\tilde{N}_k}$ from the set of proposals produced by the Faster R-CNN fine-tuned by $\{\mathbf{x}_{k-1}^{(n)}\}_{n=1}^{N_{k-1}}$ (the target set at iteration $k-1$), as follows (3.8):

$$\{\tilde{\mathbf{x}}_k^{(n)}\}_{n=1}^{\tilde{N}_k} = \Theta(\{\mathtt{I}^{(i)}\}_{i=1}^{I_i}; \mathcal{R}_{k-1}, \mathcal{F}_{k-1}) \tag{3.8}$$

with a first iteration ($k=1$) that uses an initial generic network $(\mathcal{R}_0, \mathcal{F}_0)$.

3.3.1.2 Update step

This step defines the likelihood term (3.9) by utilizing a likelihood function. This latter assigns a weight $\tilde{\pi}$ to each proposal sample $\{\tilde{\mathbf{x}}_k^{(n)}\}_{n=1}^{\tilde{N}_k}$ returned by the Faster R-CNN at the prediction step.

$$p(\mathbf{z}_k|\mathbf{x}_k = \tilde{\mathbf{x}}_k^n) \propto \tilde{\pi}_k^n \tag{3.9}$$

The likelihood function employs visual contextual cues extracted from the target video sequence and the confidence scores given by the output layer of the Faster R-CNN, to attribute a weight for each sample. More details about the likelihood function are given in section 3.3.2. The update step gives as an output a set of weighted target samples, which will be referred to as "the weighted target dataset" hereafter (3.10):

$$\{(\tilde{\mathbf{x}}_k^{(n)}, \tilde{\pi}_k^{(n)})\}_{n=1}^{\tilde{N}_k} \tag{3.10}$$

where $\{\tilde{\mathbf{x}}_k^{(n)}, \tilde{\pi}_k^{(n)}\}$ represents a target sample with its associated weight and $\tilde{N}_k$ is the number of weighted samples.

3.3.1.3 Sampling step

The aim of this last recursive-filter step is to build a new specialized dataset by deciding, according to the strategy of sampling (defined in the contribution), which samples will be included in the produced dataset $\mathcal{D}_k = \{\mathbf{x}_k^{(n)}\}_{n=1}^{N_k}$ at the iteration k from the weighted dataset $\{\tilde{\mathbf{x}}_k^{(n)}, \tilde{\pi}_k^{(n)}\}_{n=1}^{\tilde{N}_k}$. A sampling strategy is applied in order to generate a new unweighted dataset which has the same number of samples as the weighted one. To do this, we apply the IR algorithm, according to equation (3.11):

$$\mathcal{D}_k = \{\mathbf{x}_k^{(n)}\}_{n=1}^{N_k} = IR\left(\{\tilde{\mathbf{x}}_k^{(n)}, \tilde{\pi}_k^{(n)}\}_{n=1}^{\tilde{N}_k}\right) \tag{3.11}$$

This step generates a new set $\mathcal{D}_k$ by drawing samples according to the weight $\tilde{\pi}_k^{(n)}$

Furthermore, we can apply the sampling strategy to select confidence images instead of the confidence samples from target dataset. The ideas consists to apply

the IR algorithm to select the confidence images relevant to the weights of their samples. To this, we calculate a weight of each image by 3.12:

$$\tilde{\gamma}_k^{(i)} = \frac{\sum_{n=1}^{N} \tilde{\pi}_k^{(n)}}{N} \tag{3.12}$$

where $\tilde{\gamma}_k^{(i)}$ presents the weight of an image, $\tilde{\pi}_k^{(n)}$ is the weight of sample and N presents the number of samples in that image.

According to equation 3.13, we use the IR algorithm to transform the weight on a number of repetitions, by the repeating the images associated to a high weight using numerous ones and by repeating the images associated to a low weight by few ones. This solution permits to generate a new set $\mathcal{D}_k$ of images by drawing images according to their weights $\tilde{\gamma}_k^{(i)}$.

$$\mathcal{D}_k = IR\left(\{\tilde{\mathrm{I}}_k^{(i)}, \tilde{\gamma}_k^{(i)}\}_{i=1}^{\tilde{I}_k}\right) \tag{3.13}$$

3.3.2 Likelihood function

In order to choose the correct proposal, we put forward a likelihood function based on the verification strategy, which assigns a weight $\pi_k^{(n)}$ for each sample $\tilde{\mathbf{x}}_k^{(n)}$ returned by the prediction step. Our specifically designed likelihood function not only incorporates the confidence scores given by the output layer of the Faster R-CNN but also adds a spatial-temporal cues, to prioritize the selection of the correct samples and to reduce the risk of including wrong proposal samples in the specialized dataset.

Summarising the tests carried out on different databases, it is noticed that the generic Faster R-CNN is robust to generate true positive samples with a high score, and its selection of these ones will start to fail when the score of samples is lower than the score threshold α_k. For this reason, we keep the samples which have a confidence score greater than or equal to α_k and we propose an observation function f_L to assign a weight to each proposal sample that has a score lower than α_k, according to (3.14):

$$\pi_k^{(n)} = \begin{cases} s_k^{(n)} & \text{if} \quad s_k^{(n)} \geq \alpha_k \\ f_L(\tilde{\mathbf{x}}_k^{(n)}) & \text{if} \quad s_k^{(n)} < \alpha_k \end{cases} \tag{3.14}$$

Accordingly, we choose a dynamic threshold through iterations to avoid the problem of integrating negative samples into the specialized dataset. We are not limited to a fixed predefined threshold because the choice will be dynamic and will be related to the following equation (3.15):

$$\alpha_k = \begin{cases} \dfrac{\tilde{s}_k}{\tilde{s}_{k-1}}\, \alpha_{k-1} & \text{if} \quad k \neq 0 \\ \alpha_0 & \text{if} \quad k = 0 \end{cases} \tag{3.15}$$

where α_0 is the initial value of the score threshold (fixed to 0.5 for our experiments) and $\tilde{s}_k$ is the mean value of $s_k^{(n)}$ at iteration k.

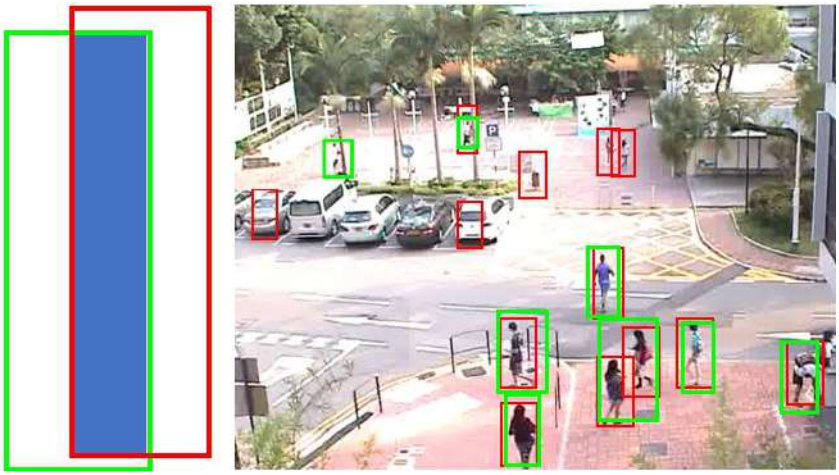


Figure 3.3: The red rectangle presents the area in pixels of the considered RoI, the green rectangle is the foreground area, and the rectangle filled in blue is the area of intersection.

Lower than α_k, the deep detector will start to fail and it will become unable to correctly select positive samples from a specific scene. To solve this problem, we propose an observation function f_L in order to favor the selection of positive samples using the information extracted from the target scene. This function is based on the visual spatio-temporal cue "Background extraction overlap score", to attribute a weight for each sample.

In a traffic scene, it is rare for a traffic object to stay fixed for a long time, and a good detection occurs on a foreground blob; whereas, false positive background detections give some RoIs that appear over time at the same location and with the same size.

To assign a weight for each sample, we calculate an overlap_score λ_o (equation 3.16) that compares the RoI associated to one sample with the output of a binary foreground extraction algorithm.

$$\lambda_o \doteq \frac{2(RoI\_AR \times FG\_AR)}{RoI\_AR + FG\_AR} \tag{3.16}$$

where RoI_AR is the area in pixels of the considered RoI and FG_AR is the foreground area at the RoI position (see Figure 3.3).

The background subtraction algorithm used in the proposed observation function is adopted from [Zivkovic 2006] and was called the "$BackgroundSubtractorMOG2$" algorithm. This latter is a Gaussian mixture-based background/Foreground segmentation algorithm.

One important property of this algorithm is that it chooses the appropriate number of Gaussian distribution for each pixel. It provides better adaptability to illumination changes. In our work, to ameliorate the result generated by the background subtraction algorithm mentioned above, we put forward some improvements

such that:

- We apply several morphological filtering operations like erosion and dilation to the result of this algorithm so as to remove unwanted noise.
- We remove the blobs which have a surface area less than 100 pixels.

The observation function (Algorithm 1) will assign a high weight to a positive proposition if it has an overlap_score λ_o that exceeds a fixed threshold α_p, which is determined empirically.

Algorithm 1 Observation function

Set $\{\tilde{\mathbf{x}}_k^{(n)}\}_{n=1}^{\tilde{N}_k}$ with associated RoI position $\{\mathbf{p}_k^{(i)}\}_{i=1}^{\tilde{N}_k}$ into the target video-sequence
Target video sequence $\mathcal{I}_t$
α_p: overlap threshold
Set $\{\tilde{\pi}_k^{(i)}\}_{i=1}^{\tilde{N}_k}$ of weights associated to samples

---

for $i = 1$ to $\tilde{N}_k$ **do**
$\tilde{\pi}_k^{(i)} \leftarrow 0$
`/* Visual contextual cue computation */`
$\lambda_o = \dfrac{2(RoI\_AR \times FG\_AR)}{RoI\_AR + FG\_AR}$
`/* Weight assignment */`
if $(\lambda_o \geqslant \alpha_p)$ **then**
$\tilde{\pi}_k^{(i)} \leftarrow \lambda_o$

Considering the likelihood function, the favoring of sample associated to the right label becomes efficient and easier.

3.3.3 Fine-tuning step

In the proposed framework, the aim of the fine-tuning step is to specialize the RPN and the Fast R-CNN deep networks to a specific scene. Accordingly, we use the target detection boxes included in the specialized dataset $\mathcal{D}_k$ and the RPN fine-tuning process mentioned in [Ren 2015c].

To do this, we use a sliding window approach to generate k bounding boxes for each position on the feature map produced by the last convolutional layer, where each bounding box is centered on the sliding window and is associated with an aspect ratio and a scale (see Figure 3.4). The intersection-over-Union (IoU) overlap between each box of the specialized dataset $\mathcal{D}_k$ and the bounding boxes is then computed. A bounding box is designated as a positive training example if it has an IoU overlap greater than a predefined threshold with any $\mathcal{D}_k$ box, or if it is the bounding box that has the highest IoU with a $\mathcal{D}_k$ box. A proposal is designated as a negative example to a non-positive bounding box if its maximum IoU ratio with

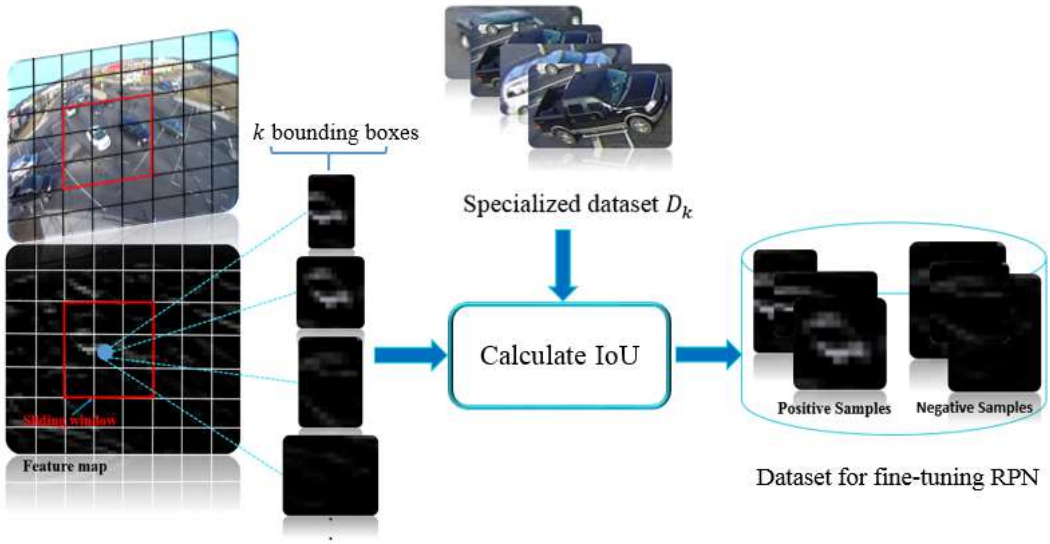


Figure 3.4: Description of training strategy for the RPN fully-convolutional network

all boxes of the specialized dataset $\mathcal{D}_k$ is less than another predefined threshold. The bounding boxes that are neither positive nor negative do not contribute to the training.

Note that, the RPN fine-tuning process mentioned above does not consider that there might exist multiple copies (maximum twice) of the target detection box in the specialized dataset $\mathcal{D}_k$ because the main objective of using the IR algorithm proposed in the sampling strategy is not to increase the size of the database with samples which have high weights but to decrease the risk of distorting the specialized dataset $\mathcal{D}_k$ with wrong labelled examples because it is possible that the weighted target dataset contains wrong samples classified as traffic objects because their $\lambda_o >= \alpha_p$.

After training the RPN, these proposals are used to train the Fast R-CNN. Figure 3.4 illustrates the training strategy of the RPN fully-convolutional network.

Therefore, a new specialized RPN network and the Fast R-CNN one are generated being fine-tuning with the specialized dataset. These networks will become the input of the prediction step in the next iteration and will generate new object proposals (bounding boxes) in the target scene.

$$\{\mathcal{R}_k, \mathcal{F}_k\} = f(\mathcal{I}_t, \mathcal{D}_k; \mathcal{R}_{k-1}, \mathcal{F}_{k-1}) \tag{3.17}$$

The suggested SMC Faster R-CNN framework is summarized in Algorithm 2.

Algorithm 2 SMC Faster R-CNN

Generic network $(\mathcal{R}_0, \mathcal{F}_0)$
Number of iterations: K
Number of target samples $\tilde{N}_k$
Unweighted target dataset: $\tilde{\mathcal{W}}_k$
Target video sequence $\mathcal{I}_t$
Specialized network $(\mathcal{R}_K, \mathcal{F}_K)$
Specialized dataset $\mathcal{D}_K$

---

for `k=1,...,K` **do**

`/* Prediction step */`
$\{\tilde{\mathbf{x}}_k^{(n)}\}_{n=1}^{\tilde{N}_k} = \Theta(\{\mathtt{I}^{(i)}\}_{i=1}^{I_i}; \mathcal{R}_{k-1}, \mathcal{F}_{k-1})$
`/* Update step */`
$\tilde{\mathcal{W}}_k = \{\tilde{\mathbf{x}}_k^{(n)}, \tilde{\pi}_k^{(n)}\}_{n=1}^{\tilde{N}_k}$
`/* Sampling step */`
for $n = 1$ to $\tilde{N}_k$ **do**
Draw a sample: $\{\tilde{\mathbf{x}}_k^{(n)}\}_{n=1}^{\tilde{N}_k}$, according to the weight $\tilde{\pi}_k^{(n)}$
`/* Fine-tuning step */`
$\{\mathcal{R}_k, \mathcal{F}_k\} = f(\mathcal{I}_t, \mathcal{D}_k; \mathcal{R}_{k-1}, \mathcal{F}_{k-1})$
$\{\mathcal{R}_K, \mathcal{F}_K\} = (\mathcal{R}_k, \mathcal{F}_k)$

3.4 Experimental results

This section presents the experiments that have been achieved in order to compare the SMC Faster R-CNN with the relevant frameworks on several public and private datasets for single and multi-traffic object detection.

3.4.1 Implementation details

We describe the implementation details of the SMC Faster R-CNN algorithm. We use the pre-trained VGG16 model [Simonyan 2014] to initialize the Faster R-CNN network, which is used in most recent state-of-the-art approaches [Girshick 2015a][Girshick 2014b].

Both RPN and Fast R-CNN are fine-tuned end-to-end by back-propagation and stochastic gradient descent [LeCun 1989] with a weight decay of 0.0005 and a momentum of 0.9. We use the alternating training algorithm [Ren 2015c] for Faster R-CNN training from one iteration to another. The Faster R-CNN is fine-tuned on a NVIDIA GeForce GTX TITAN X GPU with a 12GB memory.

Following multiple experiments, we chose 9 as the number of bounding boxes (3 aspect ratios [2:1, 1:1, 1:2] and 3 scales [128^2, 256^2, 512^2]) generated on each position of the sliding windows. We also chose 0.7 as the threshold of the IoU to select the positive samples and 0.3 for the negatives to build the training dataset.

The parameter K (number of iterations of the SMC process) is fixed to $K = 2$.

Figure 3.7 shows that the specialization converges after two iterations for both car and pedestrian detection applied on the MIT Traffic dataset (introduced in the next section).

3.4.2 Datasets

The PASCAL VOC 2007 dataset [Everingham 2010] was utilized to learn the generic Faster R-CNN. This dataset consists of about 5,011 trainval images and 4,952 test ones over 20 object categories. In our experiments, we use only 713 annotated cars, 2,008 pedestrian, 186 buses and 245 for motorbikes, to fine-tune the generic Faster R-CNN.

The evaluation is achieved on three target datasets (two public ones and a private one):

- **CUHK Square dataset** [Wang 2012a]: This is a video sequence of road traffic which lasts 60 minutes. 352 images are utilized for specialization, uniformly extracted from the first half of the video. 100 images are used for the test, extracted from the latest 30 minutes. Annotations were provided by Wang [Wang 2012a] for pedestrian detection (called CUHK_WP after). However, we notice that some annotation errors are made in the public ground truth and we suggest a new annotation (called CUHK_MP after) (see Figure 3.5.a).

- **MIT Traffic dataset** [Wang 2009]: This is a 90-minute video. We use 420 images from the first 45 minutes for specialization. 100 images are uniformly sampled from the last 45 minutes for the test. Annotations are available for pedestrians [Wang 2009] (called MIT_WP) and cars [Li 2015b] (called MIT_LV). Since some annotation errors are present, we propose new annotations (called MIT_MV) (see Figure 3.5.b).

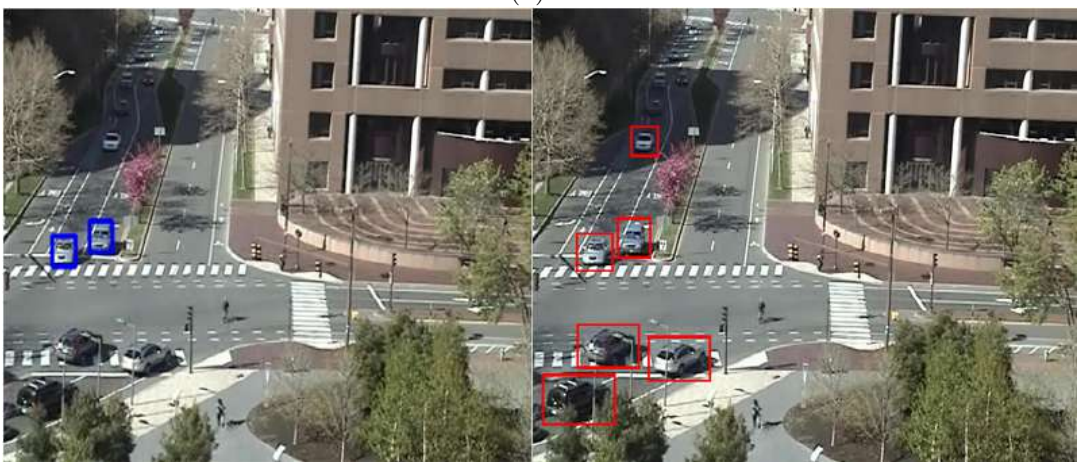
- **Logiroad Traffic dataset:** This is a private video sequence of road traffic which lasts 20 minutes. We utilize 600 images for specialization, extracted uniformly from the first 15 minutes of the video. 100 images are used for the test, extracted from the latest 5 minutes. Annotations are available for vehicles (called Logiroad_MV).

3.4.3 Descriptions of experiments

Evaluation is performed in terms of recall False Positives Per Image (FPPI) curves. The PASCAL 50 percent overlap criterion [Everingham 2010] was utilized to give a score for the detection bounding boxes. The SMC Faster R-CNN framework is compared with several state-of-the-art frameworks:

- Generic Faster R-CNN: It is a detector fine-tuned on the generic dataset. This is the baseline for our comparison.

(a)



(b)

Figure 3.5: Some annotations provided by Wang [Wang 2014b] ground truth on CUHK dataset ((a) left image), Li [Li 2015b] ground truth on MIT Traffic dataset ((b) left image) and our annotations ((a), (b) right images). There are several missing objects in the baseline annotations: This is why we propose an updated version.

- Maamatou (2016) [Maâmatou 2016c]: An SMC framework was applied to specialize a generic HOG-SVM classifier to a particular video sequence for traffic object detection.

- Xudong Li (2015) [Li 2015b]: A deep learning domain adaptation framework was proposed for vehicle detection with manually annotated data from the target scene. Unlike other methods, the latter was not totally automatic and requires some manual annotations.

- Mao (2015) [Mao 2015]: A framework was suggested to automatically train scene-specific pedestrian detectors based on tracklets.

- Htike (2014) [Htike 2014]: A non-iterative domain adaptation framework was used to adapt a pedestrian detector to video scenes.

- Zeng (2014) [Zeng 2014]: A deep learning domain adaptation framework was proposed to automatically select training samples from target scenes without manual labelling for pedestrian detection.

- Wang (2014) [Wang 2014b]: A specific-scene detector was trained on only relevant samples collected from both source and target datasets.

- Nair (2004) [Nair 2004]: An iterative self-training framework for detector adaptation was opted for using a background subtraction algorithm.

3.4.4 Results and analysis for single-traffic object

Given each dataset and its annotation, we present the ROC curves (Figure 3.6) of the generic Faster R-CNN, the SMC Faster R-CNN and the available state-of-the-art frameworks. The ROC curves present the comparison between the true detection rate and the false positive detection rate per image. Furthermore, we give two comparative synthetic tables: one for pedestrian detection (cf. Table 3.1) and the other for vehicle detection (cf. Table 3.2). In addition, on the last line of both tables, the improvement between the generic Faster R-CNN and the SMC Faster R-CNN is given.

- **Comparison with generic detector:** Figure 3.6 shows that the specialized Faster R-CNN detector significantly outperforms the generic one on all public and private datasets with several annotations. The median improvement is 51%.

- **Comparison with state-of-the-art:** According to the ROC curves at the top of Figure 3.6, for the CUHK pedestrian detection, the SMC Faster R-CNN outperforms all other state-of-the-art frameworks. Besides, the detection rate achieved with our proposed annotations on CUHK_MP is nearly 90% for 0.5 FPPI. However, despite of the wrong annotations given by Wang (left curve in the top of Figure 3.6), the SMC Faster R-CNN also exceeds the six other

specialized detectors of Nair (2004), Wang (2014), Zeng (2014), Htike (2014), Mao (2015) and Maamatou (2016) respectively by 24%, 45%, 53%, 49%, 58% and 62%.

For the MIT pedestrian detection MIT_WP in Table 3.1), the specialized deep detector proposed by Zeng (2014) exceeds the SMC Faster R-CNN detector for an 0.5 FPPI, which is less than 0.9.

Despite the wrong annotations given by Li *et al.* [Li 2015b], Figure 3.6 (right curve in the middle) shows that for the MIT car detection (MIT_LV), the proposed SMC Faster R-CNN clearly outperforms the specialized CNN detector proposed by Li (2015) which trained with manual data labeling from the target scene. According to Table 3.2), for the MIT and Logiroad car detection with the proposed annotations, the SMC Faster R-CNN is ranked first and exceeds the specialized detector suggested by Maamatou (2016).

One can notice that the generic Faster R-CNN, fine-tuned on the PASCAL VOC 2007 dataset, has a poor detection rate resulting in a limitation of the size of the specialized dataset.

Table 3.1: Comparison of detection rate for pedestrian with state of the art (at 0.5 FPPI)

Approach \ Dataset	CUHK_WP	CUHK_MP	MIT_WP
Nair [Nair 2004]	0.24	–	0.35
Wang [Wang 2014b]	0.45	–	0.42
Zeng [Zeng 2014]	0.53	–	0.58
Htike [Htike 2014]	0.49	–	–
MAO [Mao 2015]	0.58	–	–
Maamatou [Maâmatou 2016c]	0.62	0.58	0.40
Generic Faster R-CNN [Ren 2015c]	0.60	0.69	0.07
SMC Faster R-CNN	**0.65**	**0.88**	**0.47**
Improvement / Generic (%)	8%	28%	571%

Table 3.2: Comparison of detection rate for car with state of the art (at 1 FPPI)

Approach \ Dataset	MIT_LV	MIT_MV	Logiroad_MV
Li [Li 2015b]	0.77	–	–
Maamatou [Maâmatou 2016c]	–	0.29	0.47
Generic Faster R-CNN [Ren 2015c]	0.68	0.38	0.40
SMC Faster R-CNN	**0.77**	**0.80**	**0.70**
Improvement / Generic (%)	13%	110%	75%

- **Effect of likelihood function:** To show the effectiveness of our likelihood function, the ROC curves in Figure 3.8 show the comparison between using

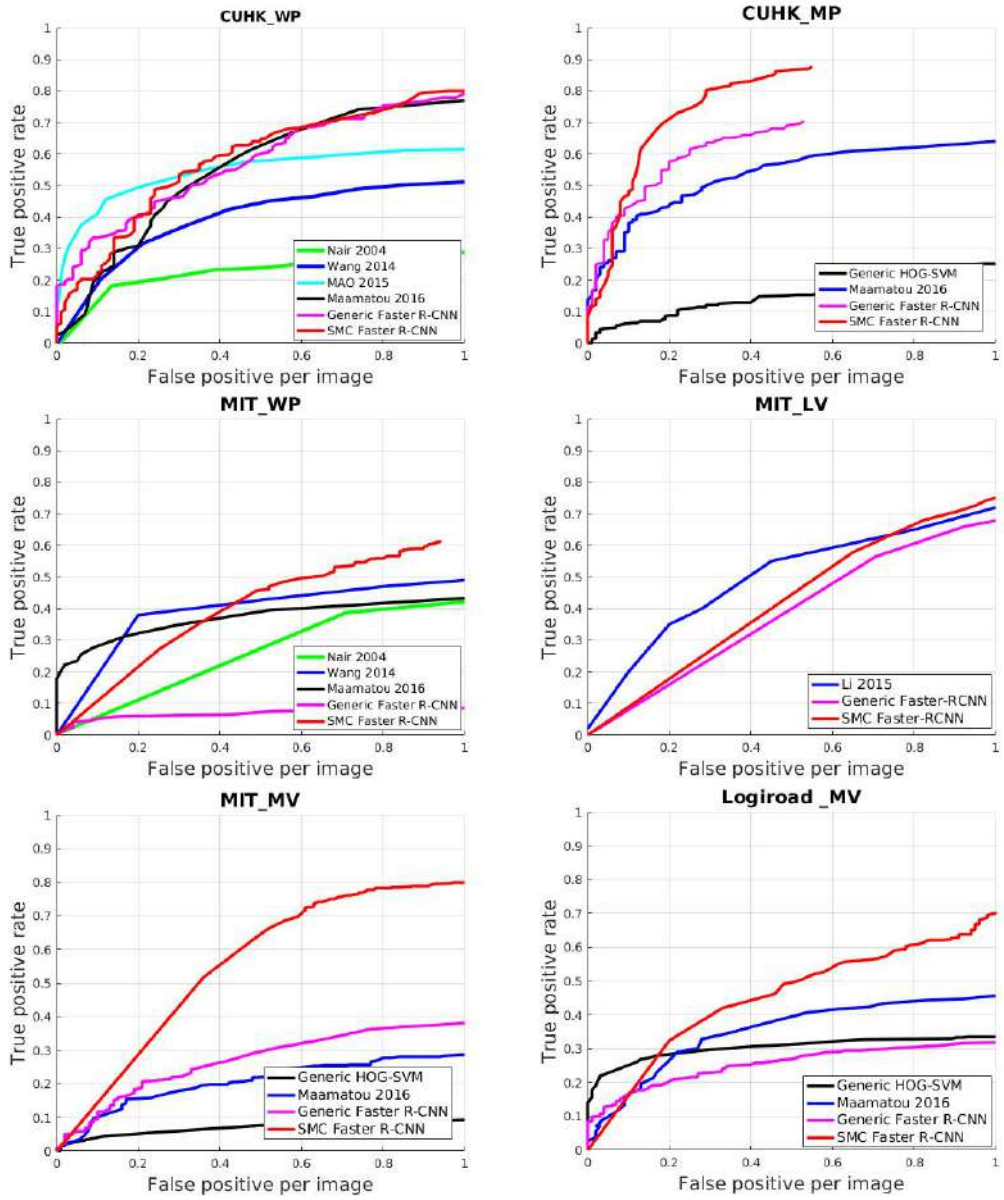


Figure 3.6: ROC curves for several public and private datasets and with different annotations

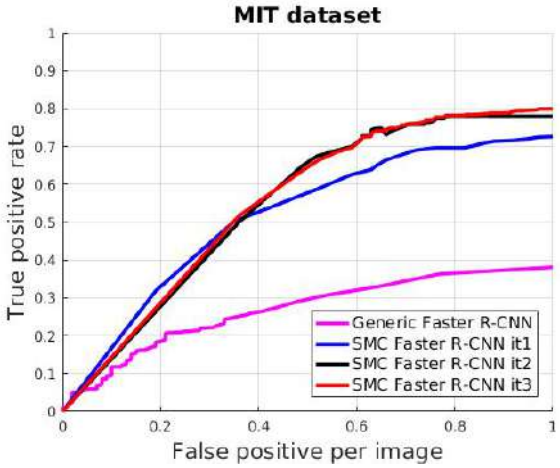


Figure 3.7: ROC curves for convergence of specialization process

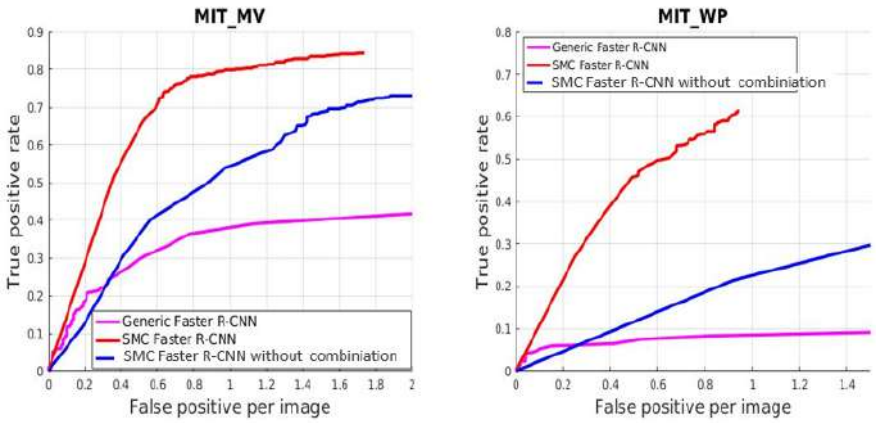


Figure 3.8: Effect of the likelihood function in our specialization framework on MIT Traffic dataset for both pedestrian and car detections.

the likelihood function based only on confidence score predicted by the Faster R-CNN and our proposed one on two datasets.

The red curves in Figure 3.8 present our proposed likelihood function based on the combination between the confidence score and the spatio-temporal cue, and the blue ones indicate the use of the confidence score only, which is given by the output layer of the Faster R-CNN. The results demonstrate that the proposed likelihood function based on using the verification strategy improves the detector performance and accelerates the convergence of the specialization process. Furthermore, we cannot say that this choice is the best because it is possible to ameliorate the suggested framework by proposing other strategies for the SMC steps. For example, we can improve the likelihood function with more complex visual cues like tracking, optical flow or contextual information to enhance the weighting of positive samples.

3.4.5 Results and analysis for multi-traffic object

We evaluate the proposed approach for multi-traffic objects on two datasets, the MIT Traffic dataset and the Logiroad one using two evaluation criteria: namely the ROC curves and the confusion matrix (classical metrics for object detection).

For the MIT Traffic dataset, we select 2 classes {'pedestrian', 'car'} and 4 classes for the Logiroad Traffic dataset {'pedestrian', 'car', 'bus', 'motorbike'}.

The results are reported in Table 3.3. The SMC Faster R-CNN presents a median improvement of 89% related to the generic detector. Moreover, Tables 3.4 and 3.5 provide the associated similarity matrix. We show that some confusion may occur between motorbikes and cars or between buses and cars. Furthermore, these results illustrate that our framework has a robust performance for multi-traffic object detection. This indicates that it is useful to run our specialization algorithm whenever we have a new sequence and we want to automatically generate a much better deep detector than the generic one.

Table 3.3: Detection rate for multi-traffic object detection with SMC Faster R-CNN (at 1 FPPI)

Approach \ Dataset	Logiroad-Car	Logiroad-Per	Logiroad-Moto	MIT-Car	MIT-Per
Generic Faster RCNN	0.28	0.24	0,065	0.32	0.05
SMC Faster RCNN	**0.60**	**0.36**	**0.18**	**0.73**	**0.30**
Improvement/Generic	114%	50%	176%	128%	500%

Table 3.4: Illustration of similarity matrix between traffic object categories on Logiroad Traffic dataset (diagonal row shows the accuracy to recognize traffic objects of its own class)

Actual class \ Predicted class	Pedestrian	Car	Motorbike	Bus
Pedestrian	140/**97%**	12/1.5%	5/14%	0
Car	0	750/**96%**	1/3%	1/2.5
Motorbike	5/3%	12/1.5%	30/**83%**	1/2.5%
Bus	0	7/1%	0	38/**95%**
Total	145	781	36	40

Table 3.5: Illustration of similarity matrix between traffic object categories on MIT Traffic dataset

Actual class \ Predicted class	Pedestrian	Car
Pedestrian	342/**99.7**%	7/1.6%
Car	1/0.3%	420/**98.4**%
Total	343	427

3.5 Discussion

This section provides a discussion about the advantages of our work over the state-of-the-art scene specialization frameworks and the main difference between the SMC framework proposed by Maamatou *et al.* [Maâmatou 2016c] and the suggested one.

Most of the specialization frameworks cited above are based on hard-thresholding rules and are very sensitive to the risk of drifting during iterations, or they are applied only to particular classifiers or few detectors like the HOG-SVM. In fact, several frameworks are limited only for mono-traffic object detection, or they need many iterations for the convergence of the specialization process.

Differently from the existing works, we put forward an iterative process based on the formalism of the SMC filter to specialize the Faster R-CNN deep detector for multi-traffic object detection. Accordingly, our proposed framework allows reducing the risk of drifting by using efficient strategies during iterations and it can be used to specialize any deep detector like the Fast R-CNN [Girshick 2015a] and the R-CNN [Girshick 2014b]. Furthermore, this framework may be applied using several strategies on each step of the SMC filter. Particularly, we cite some advantages of the suggested framework:

- We propose a likelihood function based on an efficient strategy of verification. This latter is used to favor the selection of samples associated to the right label from a target scene, to decrease the risk of drifting the detector over iterations by reducing the introduction of mislabeled examples in the training dataset.

- The suggested framework automatically specializes a generic detector to a target scene. This framework iteratively estimates the unknown target distribution as a specialized dataset by selecting only relevant samples from the target dataset. These samples are selected to re-train a specialized detector that increases the detection accuracy in the target scene. Contrarily, several state-of-the-art frameworks have aimed to collect samples from both source and target datasets to improve accuracy by augmenting the training dataset. These frameworks have led to extend the size of the training dataset and to slightly decrease the performance of the detector during iterations.

- To permit training an accurate specialized detector with the same function as the generic one and avoiding the distortion of the specialized dataset, we suggest a sampling strategy which uses the IR algorithm to select the confidence samples relevant to their weight returned by the likelihood function. This makes our framework applicable to specialize any deep detector, while training the treating samples according to the importance of their weight without modifying the training function, as done by [Wang 2014b] [Wang 2012a].

- We derive a generic transfer learning framework in which many strategies can be integrated in the SMC steps.

Table 3.6 provides a comparison over the SMC framework proposed by Maamatou *et al.*[Maâmatou 2016c] and our suggested one.

Table 3.6: Description of the difference between the work of Maamatou *et al.* [Maâmatou 2016c] and our proposed one.

	Maamatou *et al.* [Maâmatou 2016c]	Our framework
Generic detector	HOG-SVM	Faster R-CNN
Transfer learning	Positive & negative samples	Positive samples
Specialized dataset	Source & target samples	Target samples
Output	Specialized SVM	SMC Faster R-CNN
Specialized process	SMC steps	SMC steps & fine-tuning step
Traffic objects	Pedestrian	Multi-traffic object

The advantages of our specialization framework over the SMC framework [Maâmatou 2016c] are:

- In [Maâmatou 2016c], for each iteration, they selected relevant samples from both source and target domains to create a specialized dataset. In contrast, our proposed framework selects only the relevant samples from target domains according to the importance of their weights to create a specialized dataset. This solution enables a faster learning of detector and leads to an increase in detection accuracy.

- The specialized framework proposed in [Maâmatou 2016c] was very sensitive to the risk of drifting because they used only a background subtraction algorithm to assign weights to the target samples. Indeed, several static objects or those with similar background appearances were classified as negative samples, and mobile background objects were labeled as objects of interest. On the other hand, to avoid the distortion of the specialized dataset with mislabeled samples, we propose a likelihood function based on the verification strategy, which combines the confident-score given by the output layer of the Faster R-CNN network with spatial-temporal cues in order to attribute confidence weights to target samples.

- The work of Maamatou *et al.*[Maâmatou 2016c] was limited for only single-traffic object detection, but our proposed one is extended for multi-traffic objects like cars, pedestrians, buses, motorbikes...

- Differently from the work in [Maâmatou 2016c], we put forward new strategies for transfer learning inspired from the three steps of the SMC filter to specialize the Faster R-CNN deep detector.

- It is important to say that we need only two iterations for the convergence of our specialization process, whereas the framework suggested in [Maâmatou 2016c] required at least 4 iterations for this convergence.

- The proposed approach in [Maâmatou 2016c] was limited to specialize the SVM classifier, in contrary, our framework is applicable to specialize some deep detector like the Fast R-CNN [Girshick 2015a], the Faster R-CNN [Ren 2015c] and the R-CNN [Girshick 2014b].

3.6 Conclusion

In this chapter, we have put forward an efficient framework based on the formalism of the SMC filter to specialize the Faster R-CNN deep detector for multi-traffic object detection. This framework approximates the unknown target distribution by selecting relevant samples from target datasets. These samples are utilized to fine-tune a specialized deep detector in order to decrease the detection rate in the target scene. Given a generic detector and a target video sequence, this framework automatically provides a robust specialized detector. Moreover, the proposed framework allows reducing the risk of drifting by using efficient strategies during iterations and it can be used to specialize any deep detector. The extensive experiments have demonstrated that the suggested framework has produced a specialized detector that performs much better than the generic one for both single and multi-traffic object detections in different scenes. Furthermore, the results show that the framework outperforms the state-of-the-art specialization ones on several challenging datasets.

In the next chapter, we will present a novel tracking framework that proposes to built an intermediate interlaced video-sequence and an associated DCNN detector in order to improve the tracking performance.

Chapter 4

Power of Video Interlacing for Deep-Learning-Based Multi-Object Tracking

Contents

In this chapter, we propose an original framework for Multi-Object Tracking (MOT). This is a novel spatio-temporal-based model and a specialized deep object detector. Our MOT framework models the spatio-temporal variation of objects in *interlaced images*. A specialized interlaced DCNN detector is then trained to detect such new objects and a classical association step produces output targets. Since interlaced objects are built to increase overlap during the association step, the resulting framework improves the MOT performances comparing to the same detector/association algorithm applied on non interlaced images.

The effectiveness of this contribution is demonstrated through experiments on popular tracking-by-detection datasets such as the PETS 2009 and TUD datasets. Experimental results demonstrate that the "power of video-interlacing" outperforms several state-of-the-art tracking frameworks in multiple object context.

This work was submitted at European Conference on Computer Vision (ECCV 2018).

The structure of the chapter is organized as follows. Section 4.1 presents the context of our work and gives the contributions of this chapter. Section 4.2 provides

the related work performed in the field of object tracking. A detailed description of our tracking framework is presented in section 4.3. Section 4.4 describes the experimentation details and provides the experimental results. Finally, a conclusion of this chapter is given in section 4.5.

4.1 Introduction to visual tracking

Visual tracking for multiple objects in video sequences has been extensively studied for several practical applications such as road traffic control, driving assistance, behaviour analysis and video surveillance [Dehghan 2017][Mhalla 2017].

Multi-target tracking aims to find the trajectories of moving objects in a video sequence. This problem is generally formulated as a data association task where a generic detector localizes object bounding boxes in each frame and then a data association algorithm associates corresponding detection boxes across frames.

Visual tracking approaches can be performed offline [Milan 2014][Wang 2014c] by simultaneously exploiting all the images of a processed video, or online [Shu 2012][Naiel 2014][Danelljan 2017] by limiting themselves to past images. The online approaches are selected when the real-time aspect is paramount and produce results that are fairly comparable to offline approaches as detailed in some articles [Bae 2014][Dehghan 2017]. Since recent deep object detectors have high performances, recent tracking approaches have mostly followed tracking-by-detection techniques: it consists in using a detector of the tracked-object classes to estimate the positions of the targets at each frame, following by an association step.

Lately, these techniques have gained significant interest in the research community and they are becoming more and more popular for visual tracking applications. Several approaches based on tracking-by-detection theories have been proposed by research groups in the world to solve the problems of multi-object tracking [Kim 2015][Dorai 2017][Tang 2017]. Among these approaches, we quote in particular the tracklet algorithms [Wang 2016a][Dorai 2017] which take a sequence of frames with their respective detections. Afterward, a tracklet association method associates target objects in a video sequence. Other tracking algorithms have been focused on using appearance models to estimate the tracks in each frame. In most cases, these models are learned online and utilized to estimate an affinity score for any track-detection pair. Some of the most popular work has suggested using more reliable and robust appearance models in order to differentiate objects of similar appearance [Shu 2012][Bae 2014].

Despite these efforts, existing multi-object tracking frameworks still suffer from various challenges such as noisy detections, occlusions and inaccurate detections in crowded scenes. Such issues frequently affect the tracking performance in real world-scenarios.

In recent years, deep learning techniques have achieved the state-of-the-art performance in several computer vision applications such as object detection [Liu 2016][Shen 2017][Mhalla 2017], semantic segmentation

[Kalogerakis 2017][Rota Bulo 2017] and object tracking [Nam 2016][Wang 2016b] [Wang 2017]. Yet, the performances of the existing tracking frameworks based on deep learning [Leal-Taixé 2016][Wang 2016a][Milan 2017] are limited and are not as competitive as the approaches based on hand-crafted features. In this chapter, we propose to build an interlaced representation of an input video sequence that combines several frames in an interlaced one. The resulting interlaced video provides for each frame spatio-temporal information that should be learnt into a DCNN. By detecting moving objects, the network learns to associate several temporal instances of the object in the same interlaced frame. The produced detection is then easier to associate because it is naturally overlapped between successive interlaced frames.

The main contributions of this chapter can be summarized as:

- An original interlaced model combined with a specialized object detector which improves the performances of the tracking-by-detection based MOT algorithms.
- A set of comparative experiments on the PET2009 and TUD benchmarks, which achieves competitive results with current state-of-the-art tracking frameworks.

4.2 Existing work

In this section, we are interested in the related multi-object tracking frameworks proposed to automatically track objects in video sequences.

In the recent years, tracking framework has attracted a lot of research groups in developing state-of-the-art theories and novel applications in several domains like robotics, video surveillance and intelligent transportation systems [Wang 2014c][Szegedy 2015][Wang 2016b][Milan 2017]. Several tracking frameworks have been suggested by research community in the world to solve the problems of multi-object tracking [Yoo 2017][Tang 2017][Danelljan 2017]. Some of them have been based on recursive Bayesian filters such as the Kalman filter [Lee 2004] and the Sequential Monte Carlo one [Vermaak 2005] in order to handle data association problems.

Other recent approaches have been based on matching object hypotheses obtained by detection between two consecutive frames using their characteristics like the size, the representation, the appearance and the position [Kim 2015][Yoo 2017][Valmadre 2017][Wang 2017]. On the other hand, tracking frameworks based on local data association (between two consecutive frames) have had critical limitations in resolving occlusion problems and therefore tend to generate short trajectories. Differently, some multi-object tracking frameworks build a set of trajectories through global or delayed optimization [Wu 2013][Bae 2014][Badie 2014] in order to handle occlusion problems and noisy detection in tracking sequences.

Cox *et al.*[Cox 1996] suggested a classic Multiple Hypothesis Tracking (MHT), in an effort to delay association decisions until they were resolved. However, the number of hypotheses grew exponentially. An improved version of the MHT was proposed by [Han 2004] which incorporated an appearance model to solve this issue. Kim *et al.* [Kim 2015] introduced a revisited version of the standard MHT by including an online appearance model for multiple hypothesis tracking. This new formulation led to prove substantial performance gains over the old versions of the MHT by generating tracking hypotheses at each interlaced image with a prediction training model. The tracking hypotheses would conflict when attempting to assign different identifiers to the same object. The resolution of these conflicts generated global hypotheses with associated scores, and the best hypothesis was chosen to produce the final result.

Other methods used an appearance model or features for tracking. Danelljan *et al.* [Danelljan 2014] put forward an on-line tracking framework based on adaptive color channels. This framework resolved several types of real-word scenarios, but it failed at scaling. Thus, the authors in [Danelljan 2017] proposed solution to this problem. Ma *et al.* [McLaughlin 2015] employed a novel deep learning based characteristics trained on object recognition datasets to enhance the tracking performance.

Chari *et al.* [Chari 2015] put forward pairwise costs to reinforce a min-cost network flow framework, which effectively handled overlapping problems and tracking enhancements. [Dehghan 2015] also followed the network flow framework. McLaughlin *et al.* [McLaughlin 2015] improved this min-cost network flow algorithm so that the tracking problem could be reduced in two steps. Firstly, an initial result was estimated without motion information, and secondly it was then combined with a motion feature to generate a more reliable tracking solution.

Other state-of-the-art tracking frameworks have aimed to associate the detections by introducing a similarity function between detections based on the CNN. The triplet network mentioned in [Wang 2014a][Hoffer 2015] and Siamese network [Chopra 2005] were efficiency techniques to measure the similarity between two objects. The Siamese network utilized a contrastive loss function to train the neural network, which helped the network to have small distances between the pair detections that belonged to the same objects while forcing the object with different identities to have large distances. This network is used to face recognition [Taigman 2014][Sun 2014], single object tracking [Tao 2016] and multi-target tracking [Wang 2016b][Leal-Taixé 2016]. The triplet network, an enhanced version of Siamese network, was more robust to intra-class variations [Hoffer 2015], since it used a new loss function for network training. It was utilized for characteristic learning [Hoffer 2015][Kumar 2016] and object re-identification [Cheng 2016]. Also, generalized versions of the triplet network using higher order relationships were suggested [Zhang 2016][Huang 2016a] and these approaches were useful for fine-grained feature representation learning. Accordingly, Wang *et al.*[Wang 2016a] proposed an original deep model for tracklet association which could join the Siamese CNN network learning and the temporal metrics so as to improve tracklet models.

Here, we propose to enhance the performance of classical data association algorithms and help to recover objects in complex videos including occlusion, strong motion and intersection, by using an interlacing intermediate video representation model as well as a specialized DCNN detector.

4.3 Multi-object detection and tracking using interlaced video

This section presents the principle of the multi-object tracking using interlaced videos.

A global synoptic of the suggested framework is illustrated in Figure 4.1. Given a video input, an interlacing model is applied to create an intermediate set of interlaced images. Then a specialized DCNN detector fine-tuned by interlaced datasets provides objects on each interlaced frame. Targets (object trajectories) are produced by a classical association algorithm from interlaced detections. Finally, a reverse interlacing model is applied to extract final trajectories into the initial video-sequence.

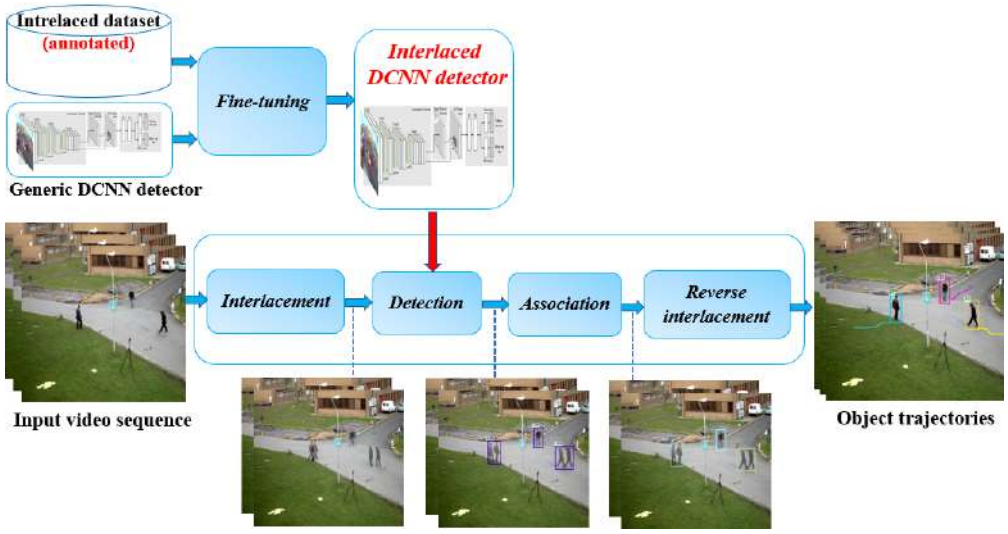


Figure 4.1: General synoptic of the proposed framework: Given a video sequence, the suggested framework uses an interlacing strategy to create interlaced images. A generic deep detector fine-tuned by interlaced datasets provides objects for each interlaced frame. Next, a data association algorithm links detection on consecutive interlaced frames. Finally, estimated trajectories of objects are produced from interlaced ones by an inverse interlacing strategy.

In what follows we describe in details the main steps of the suggested tracking framework. In subsection 4.3.1, we describe the proposed interlacing and inverse interlacing models. Subsection 4.3.2 shows how to train the interlaced DCNN detector.

4.3.1 Interlacing and inverse interlacing models

We present the interlacing and inverse interlacing mathematical models that serve as a basis of this work. Given a set of temporal images $\mathcal{I} \doteq \{\mathtt{I}_k\}_{k=1,...K}$ extracted from the input video sequence, we propose to build an interlaced image set $\tilde{\mathcal{I}} \doteq \{\tilde{\mathtt{I}}_{\tilde{k}}\}_{\tilde{k}=1,...\tilde{K}}$. The tilde ($\tilde{}$) notation is used for variables related to the interlaced video/image. If $\mathtt{I}_k(x,y)$ is the value of the (x,y) pixel (gray level or colour) of an image k, the interlaced set of images are generated by equation (4.1):

$$\tilde{\mathtt{I}}_{\tilde{k}}(x,y) \doteq \sum_{d=0,..,(D-1)} \mathtt{I}_{(kg+ds)}(x,y).\delta(y[D]-d) \tag{4.1}$$

- $\delta(.)$ represents the Kronecker Delta function (4.2):

$$\delta(n) = \begin{cases} 1 & \text{for} \quad n=0 \\ 0 & \quad otherwise \end{cases} \tag{4.2}$$

- $y[D]$ is the modulo operator (y mod D), D is the depth (number of images in one interlaced image), g is a global step (difference between two successive interlaced images), and s is a local step (the gap between the frames which are combined for an interlaced image). Figure 4.2 depicts the interlacing step.

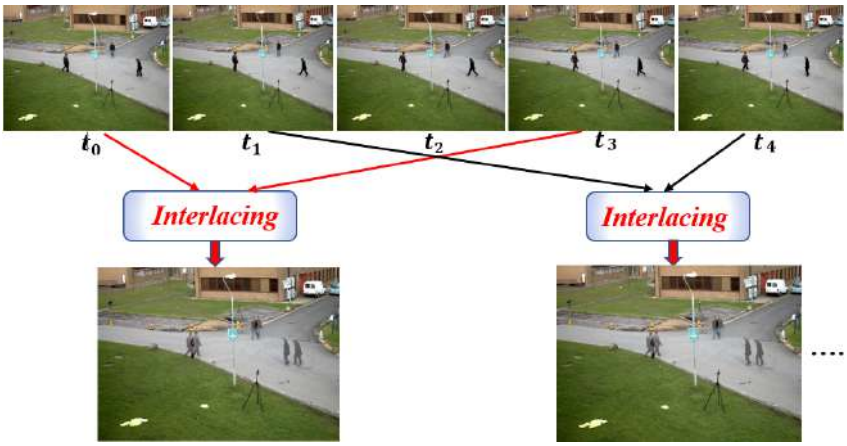


Figure 4.2: Interlacing step. The top images are temporal frames extracted from a video sequence. The bottom images present results from interlaced model and combine several video frames: objects appears several times in interlaced image.

Several strategies can be proposed. The key idea is that good strategies should produce a high overlap between interlaced detections. Figure 4.3 shows examples of interlaced strategies for several sets of parameters (D,s,g).

Since the aim of this work is to detect and track objects, we define the bounding box associated to the object i at frame k by equation (4.3):

$$\mathbf{o}_k^i \doteq (\mathbf{p}_k^{(i,1)T}, \mathbf{p}_k^{(i,2)T}, \mathbf{p}_k^{(i,3)T}, \mathbf{p}_k^{(i,4)T})^T \tag{4.3}$$

where $\mathbf{p}_k^{i,\cdot} \doteq (x_k^{(i,\cdot)}, y_k^{(i,\cdot)})^T$ is the position of the four corners (upper left, upper right, lower right and lower left) of the bounding box. Similarly, let us define the bounding boxes extracted from a tracking process applied on the interlaced video by equation (4.4):

$$\tilde{\mathbf{o}}_{\tilde{k}}^i \doteq (\tilde{\mathbf{p}}_{\tilde{k}}^{(i,1)T}, \tilde{\mathbf{p}}_{\tilde{k}}^{(i,2)T}, \tilde{\mathbf{p}}_{\tilde{k}}^{(i,3)T}, \tilde{\mathbf{p}}_{\tilde{k}}^{(i,4)T})^T \tag{4.4}$$

The object bounding box $\mathbf{o}_k^i$ is associated to the interlaced bounding box $\tilde{\mathbf{o}}_{\tilde{k}}^i$ with equation (4.5):

$$\tilde{k} = \lfloor k/g \rfloor \tag{4.5}$$

Some interlaced strategies may produce interlaced images with a high redundancy; i.e., one original object for a given time can be extracted from several interlaced images. In this case, an average object position can be computed.

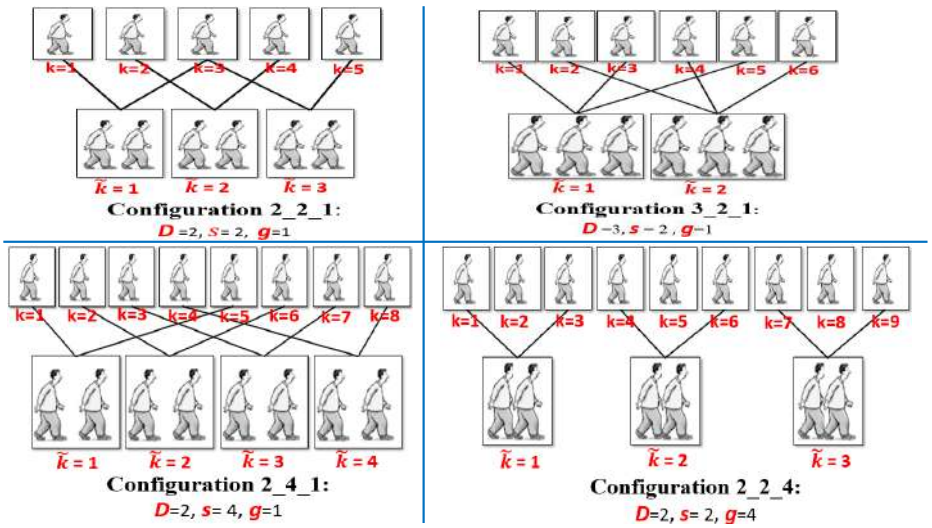


Figure 4.3: Examples of interlaced strategies with four sets of parameters (D, s, g).

We consider a constant object velocity between two interlaced images. For a depth D, an interlaced image encodes objects D times. An estimation of the object bounding box $\mathbf{o}_k^i$ in the original image k can be extracted by interpolation between the interlaced bounding box $\tilde{\mathbf{o}}_{\tilde{k}}^i$ and the interpolated interlaced bounding box $\tilde{\mathbf{o}}_{\tilde{k}+\alpha_k}^i$. Figure 4.4 illustrates the estimation of $\tilde{\mathbf{o}}_{\tilde{k}+\alpha_k}^i$. The latter box is computed by equation (4.6):

$$\tilde{\mathbf{o}}_{\tilde{k}+\alpha}^i = \tilde{\mathbf{o}}_{\tilde{k}}^i + \alpha \Delta_{\tilde{\mathbf{o}}_{\tilde{k}}^i} \tag{4.6}$$

with:

$$\alpha \doteq \frac{s(D-1)}{g} \tag{4.7}$$

and

$$\Delta_{\tilde{\mathbf{o}}_{\tilde{k}}^i} = \tilde{\mathbf{o}}_{\tilde{k}+1}^i - \tilde{\mathbf{o}}_{\tilde{k}}^i \tag{4.8}$$

where $\Delta_{\tilde{\mathbf{o}}_{\tilde{k}}^i}$ is the displacement of interlaced bounding boxes between two successive interlaced frames.

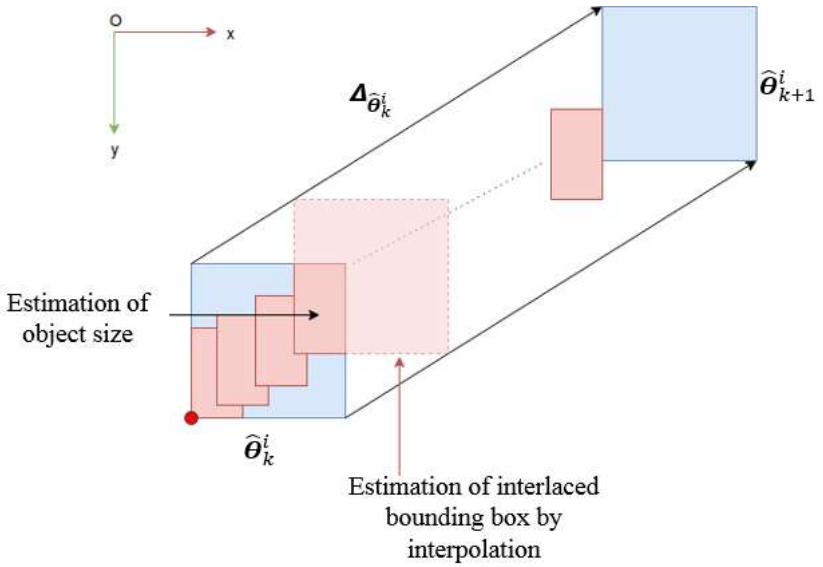


Figure 4.4: Estimation of bounding boxes by interpolation strategy. The interpolation makes it possible to determine the size of an object that is assumed to be constant over all the interlacing strategies.

The next step consists in extracting the object bounding box $\mathbf{o}^i_{g\tilde{k}+D}$ from the intersection between the interpolated object $\tilde{\mathbf{o}}^i_{\tilde{k}+\alpha}$ and $\tilde{\mathbf{o}}^i_{\tilde{k}}$ $(k = g\tilde{k})$, according to equation 4.9:

$$\mathbf{o}^i_{g\tilde{k}+D} = \tilde{\mathbf{o}}^i_{\tilde{k}} \cap \tilde{\mathbf{o}}^i_{\tilde{k}+\alpha} \tag{4.9}$$

$\cap$ is the intersection operator between the two detections $\mathbf{o}^i_{k_\cap} = \mathbf{o}^i_{k_1} \cap \mathbf{o}^i_{k_2}$ defined by (4.10):

$$\begin{cases} x^{(i,1)}_{k_\cap} = \max(x^{(i,1)}_{k_1}, x^{(i,1)}_{k_2}) \\ x^{(i,2)}_{k_\cap} = \min(x^{(i,2)}_{k_1}, x^{(i,2)}_{k_2}) \\ y^{(i,3)}_{k_\cap} = \max(y^{(i,3)}_{k_1}, y^{(i,3)}_{k_2}) \\ y^{(i,4)}_{k_\cap} = \min(y^{(i,4)}_{k_1}, y^{(i,4)}_{k_2}) \end{cases} \tag{4.10}$$

with $x^{(i,4)}_{k_\cap} = x^{(i,1)}_{k_\cap}$, $x^{(i,3)}_{k_\cap} = x^{(i,2)}_{k_\cap}$, $y^{(i,2)}_{k_\cap} = y^{(i,1)}_{k_\cap}$ and $y^{(i,4)}_{k_\cap} = y^{(i,3)}_{k_\cap}$.

In the same way, the object $\mathbf{o}^i_{g\tilde{k}}$ is extracted from the intersection between $\tilde{\mathbf{o}}^i_{\tilde{k}-\alpha}$ and $\tilde{\mathbf{o}}^i_{\tilde{k}}$, according to equation (4.11):

$$\mathbf{o}^i_{g\tilde{k}} = \tilde{\mathbf{o}}^i_{\tilde{k}} \cap \tilde{\mathbf{o}}^i_{\tilde{k}-\alpha} \tag{4.11}$$

Object ROIs for $k \in \{g\tilde{k}, g\tilde{k}+1, .., g\tilde{k}+D\}$ $(\tilde{k} = \lfloor k/g \rfloor)$ are estimated using linear interpolation :

$$\mathbf{o}^i_k = \mathbf{o}^i_{g\tilde{k}} + \beta \Delta_{\mathbf{o}^i_{g\tilde{k}}} \tag{4.12}$$

with:

$$\begin{cases} \beta = \dfrac{k - g\tilde{k}}{Ds} \\ \\ and \\ \\ \Delta_{\mathbf{o}^i_{g\tilde{k}}} = \mathbf{o}^i_{g\tilde{k}+D} - \mathbf{o}^i_{g\tilde{k}} \end{cases} \tag{4.13}$$

4.3.2 Interlaced deep detector

The interlaced deep detector is a DCNN detector which allows to detect interlaced objects in the interlaced images. To do this, we specialize the Faster R-CNN detector [Ren 2015a] with a specific interlaced dataset. The specialization of the Faster R-CNN is done by adapting its network parameters by fine-tuning with the interlaced dataset generated by the interlacing step (see Figure 4.1). The specialized Faster R-CNN deep detector consists of two stages. The first one is called Region Proposal Network (RPN) which is specialized to propose interlaced object bounding boxes. The second stage, which is Fast R-CNN [Girshick 2015b], extracts features using a RoIPool layer from each interlaced object box and performs classification and bounding-box regression. A fine-tuning of the Faster R-CNN is applied using annotated interlaced images built from several public datasets.

Given an annotated video dataset in which the trajectory of each object is labelized by a set of bounding boxes, we generate a new interlaced video dataset with annotated bounding boxes. Each object provides D "views" in an interlaced image. The interlaced bounding box is defined as the smallest bounding box that includes all bounding boxes of object "views". Figure 4.5 illustrates this step.

Figure 4.5: Building an annotated interlaced video: Two images extracted from an interlaced video ($D = 2$) with object bounding boxes in black and interlaced bounding boxes in color.

4.4 Experimentation

This section presents the various tests performed to evaluate the performance of our multi-object tracking framework.

4.4.1 Evaluation datasets

Our framework has been evaluated on several varied public datasets: PETS 2009 [Ferryman 2009] and TUD [Andriluka 2008] sequences. These datasets are mainly differentiated in terms of number of tracking objects and fields of views. Figure 4.6 shows example images of the evaluated datasets.

- S2L1, S2L2 and S2L3 are video sequences extracted from the PETS2009 dataset: This is pedestrian tracking captured by a static video-surveillance camera with a sparse crowd for S2L1, medium density crowd for S2L2 and dense crowd for S2L3.
- TUD-Stadtmitte sequence is a video captured by a static camera at about a 2-meter height. This sequence shows walking people on the street.
- TUD-Crossing sequence is a road crossing from a side view.
- TUD-Campus sequence is a short video scene with side-view pedestrians.

We demonstrate the evaluation results on testing sequences in order to verify the effectiveness of our framework.

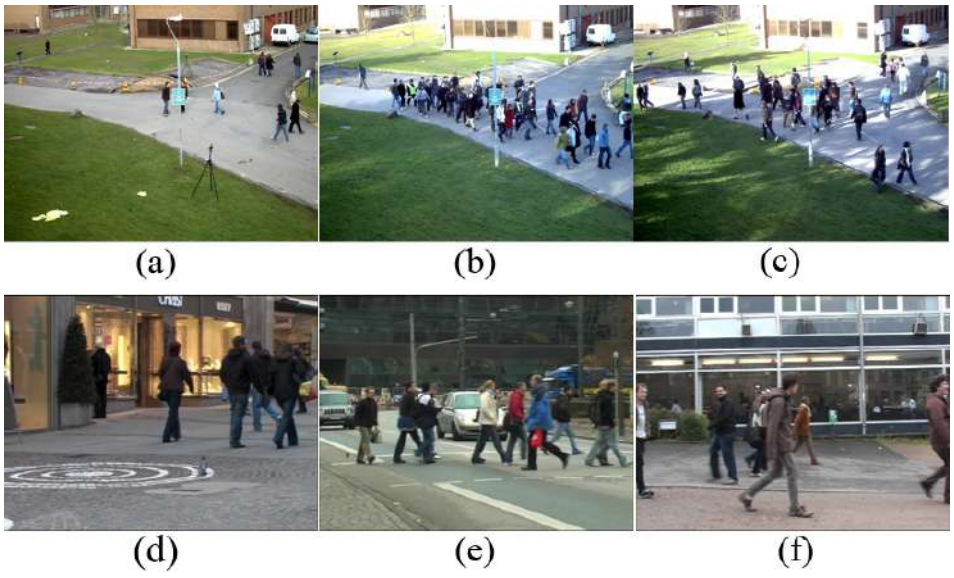


Figure 4.6: Image examples of evaluated datasets: (a), (b), (c): Images from the PETS 2009 dataset. (d), (e), (f): Images from the TUD dataset.

4.4.2 Implementation Details

In what follows, we will describe the implementation details of the proposed tracking framework.

As mentioned in section 4.3.2, the detection step is achieved by a Faster R-CNN deep detector [Ren 2015a]. This network is initialized with a pre-trained VGG16 model [Simonyan 2014]. The fine-tuning of the Faster R-CNN is applied using annotated interlaced images built from several public datasets from ETH [Ess 2008], PETS2009 [Ferryman 2009] and TUD [Andriluka 2008].

The association between detected interlaced bounding boxes is performed by the revisited version of MHT [Kim 2015]: This association method is ranked third on 2D MOT 2017 challenge. This new formulation of the standard MHT was put forward by including both an online appearance and spatio-temporal models for multiple hypothesis tracking. It has shown a substantial performance gain over the old versions of MHT. The suggested framework has been tested with this association algorithm and we only use the spatio-temporal model. However, our framework is totally generic and it can be tested with other association algorithms.

The resulting proposed MOT framework, called "FRCNNVI-MHT", is composed by: 1) a video-interlacing model, 2) a Faster R-CNN deep interlaced detector, 3) a MHT-DAM association algorithm, and 4) a video-inverse interlacing model. Default parameters for video-interlacing are set to $(D = 2, s = 2, g = 1)$ resulting in a cross validation approach (see Table 4.1).

4.4.3 Evaluation metrics

Performance evaluation is achieved using CLEARMOT metrics defined for visual multi-target tracking and detailed in [Bernardin 2008]. The following metrics are taken into account: the MOT Accuracy (MOTA), the multiple MOT Precision (MOTP), the number of identity changes (IDS), the number of False Positives (FP) and the number of Missing Positions (MS). The MOTP considers only the localization precision of individuals without taking into account identity changes. The MOTA is a score which takes into consideration false negatives, false positives and identity switches of output trajectories. The MOTA metric is considered as the most important metric to evaluate the quality of the tracking. In addition to these metrics, the number of Mostly Lost (ML) targets, Fragmentation (FM) and Mostly Tracked (MT) targets are also reported.

4.4.4 Description of experiments

The proposed tracking framework (FRCNNVI-MHT) is compared with several state-of-the-art MOT frameworks:

- FRCNN-MHT: It is a Faster R-CNN detector and an revisited MHT (MHT-DAM) association method (using only the spatio-temporal model) without interlacing strategy. This is the baseline for our comparison. It is important

to note that the same sequences are used to fine-tune the baseline FRCNN-MHT and the proposed FRCNNVI-MHT.

- DO (2017) [Dorai 2017]: A tracking method was proposed for multi-object tracking with a tracklet association algorithm.
- NF (2016) [Chari 2015]: A tracking framework suggested a pairwise cost to enforce tracklets, which effectively handled overlapping problems and tracking enhancements. The detection set is provided by the MOT challenge.
- MHT-DAM (2015) [Kim 2015]: This is the same association algorithm that we use in FRCNNVI-MHT and FRCNN-MHT but the detections are given by the MOT Challenge and an appearance model is combined with the spatio-temporal one.
- Milan (2013) [Milan 2013]: A tracking framework was suggested to formulate the tracking problem by first selecting tracklets and then connecting them using a learned a conditional random field. The detection set is provided by the MOT challenge.

4.4.5 Results and analysis

This section presents the experiments realized in order to show the performances of the proposed video-interlacing strategy for the MOT. After testing several sets of parameters for the video-interlacing model, we compare the MOT framework with the baseline method and state-of-the-art tracking frameworks. The last experiment indicates that video-interlacing can be used to reduce the computation time using a set of parameters to produce an image skipping strategy.

Since the main objective of the suggested video interlacing strategy is to increase MOT performances, a first experiment is proposed to compare, for several sets of interlaced parameters, the benefit of our contribution. Table 4.1 presents the MOTA evaluation metric for several selected configurations. Results show that if some configurations improve the MOTA comparing with the baseline method (last column), others provide weak performance. Best results are obtained with the configuration: $D = 2$, $s = 2$, and $g = 1$. This configuration will be set by default.

Table 4.1: MOTA comparison for several interlacing strategies on several sequences of TUD public dataset. The best configuration is $D = 2, s = 2, g = 1$

Sequence	(D=2,s=2,g=1)	(2,2,2)	(2,8,1)	(4,2,1)	(2,2,4)	(1,1,1)
TUD-Stadtmitte	92.8%	**93.5%**	69.3%	85.5%	87.5%	87.5%
TUD-Campus	**88.1%**	64%	45.4%	70.8%	–%	79%
TUD-Crossing	**84.8%**	69.2%	33.1%	65.5%	–%	84.4%

Figure 4.7: Comparison between proposed interlaced MOT framework (FRCNNVI-MHT for top row images) and baseline one (FRCNN-MHT for bottom row images) for two challenging situations: occlusion and pedestrian crossing.

Table 4.2 shows the performances of the proposed MOT framework compared with the baseline "Faster R-CNN & MHT-DAM" and state-of-the-art tracking frameworks. For a fair comparison, the ground truth annotations and the evaluation script provided by [Milan 2013] are used. The MOTA increases for all tested datasets. The red value on the last line of each row of Table 4.2 represents the improvement of our interlacing framework over the baseline one (without interlacing). The median improvement is 4.8% in all evaluation datasets. Figure 4.7 illustrates a comparison between our proposed framework (FRCNNVI-MHT) and the baseline one (FRCNN-MHT) on two challenging situations: occlusion and pedestrian crossing.

Compared to the state-of-the-art tracking frameworks (PETS2009 and TUD sequences), the Faster R-CNN combined with the MHT-DAM association algorithm [Kim 2015] is very competitive. Our framework has significant improvements and enhances the result of the MHT-DAM algorithm by about 12.5% on average as a MOTA evaluation metric (31.4% for TUD-Stadtmitte sequence). However, for sequence S2L3, which represents a dense crowd, the Faster R-CNN gives poor results compared to the detection set provided by the MOT challenge (MHT-DAM).

Figure 4.8 illustrates four interesting interlacing strategies that produce frame skipping; i.e., for ($D = 2$, $s = 6$, $g = 3$), odd images are never processed, resulting in a reduced computation time and for ($D = 2, s = 10, g = 5$), 25% of the images are processed.

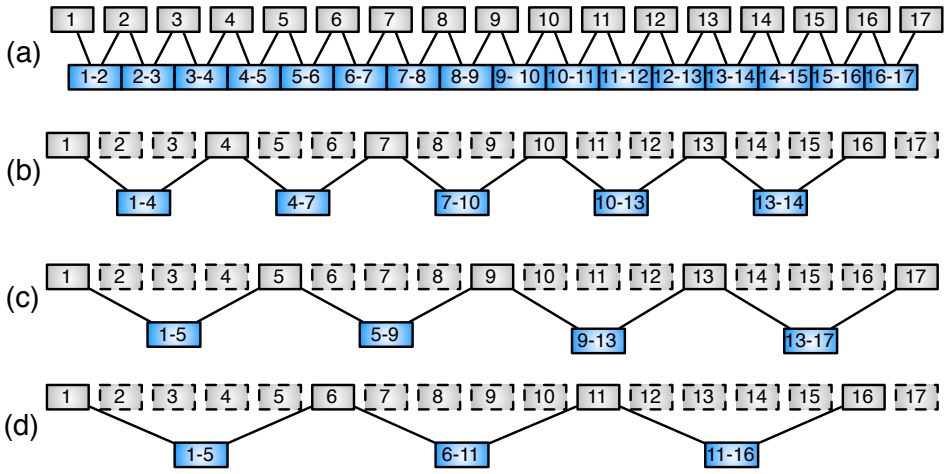


Figure 4.8: Frame skipping strategy with four different sets of parameters for interlaced model: (a) $D = 2, s = 2, g = 1$, (b) $D = 2, s = 6, g = 3$, (c) $D = 2, s = 8, g = 4$ and $D = 2, s = 10, g = 5$. The top row with grey images represents the original video sequence, and the bottom row with blue images represents the interlaced video sequence. These images are never processed (skipped images)

Table 4.2: Table summarizing results of our framework on PETS and TUD sequences. Bold indicates best value for each column for each dataset. Abbreviations are as follows: MT - Mostly Tracked, ML - Mostly Lost, FP - False Positives, FN - False Negatives, IDs - ID swaps, FM - Fragmentation.

PETS and TUD benchmarks							
Sequence	Method	MOTA	MOTP	MT	ML	FM	IDs
S2L1	MHT-DAM [Kim 2015]	83.5%	**75.8%**	18	0	35	25
	DO [Dorai 2017]	86%	69%	18	0	–	–
	NF [Chari 2015]	85.5%	72.9%	18	0	74	56
	Milan [Milan 2013]	90.3%	74%	18	0	15	22
	FRCNN-MHT	87.4%	73.2%	19	0	71	15
	FRCNNVI-MHT	**91.0 %/3.6%**	73.7%	18	0	55	**11**
S2L2	MHT-DAM [Kim 2015]	50.2%	71.3%	7	2	207	197
	DO [Dorai 2017]	**68%**	54%	–	–	–	–
	NF [Chari 2015]	50.4%	60.6%	6	3	379	244
	Milan [Milan 2013]	58.1%	59.8%	11	1	153	167
	FRCNN-MHT	57.8%	**72.2%**	20	0	305	268
	FRCNNVI-MHT	63.9%/ **6.1%**	71.8%	24	0	280	232
S2L3	MHT-DAM [Kim 2015]	35.6%	**73.5%**	8	23	34	45
	NF [Chari 2015]	**40.3%**	61.2%	12	17	50	44
	Milan [Milan 2013]	39.8%	65%	8	19	22	27
	FRCNN-MHT	28.6%	70.7%	9	6	188	204
	FRCNNVI-MHT	32.7%/**4.1%**	70.9%	6	18	147	138
TUD-Stadtmitte	MHT-DAM [Kim 2015]	61.4%	75.4%	4	0	13	19
	NF [Chari 2015]	51.6%	61.6%	2	0	22	15
	Milan [Milan 2013]	56.2 %	61.6 %	4	0	13	15
	FRCNN-MHT	87.5%	**87.3%**	9	0	6	7
	FRCNNVI-MHT	**92.8%/5.3%**	86.9%	8	0	4	3
TUD-Campus	MHT-DAM [Kim 2015]	–%	–%	–	–	–	–
	FRCNN-MHT	79%	**79.5%**	5	0	6	2
	FRCNNVI-MHT	**88.1%/9.1%**	79.3%	7	0	4	0
TUD-Crossing	MHT-DAM [Kim 2015]	–%	– %	–	–	–	–
	FRCNN-MHT	84.4%	**78.5%**	12	0	15	7
	FRCNNVI-MHT	**84.8%/0.4%**	78.3%	13	0	16	4

Figure 4.9 presents examples of the output of the interlaced specialized object detector for the four interlacing strategies mentioned above.

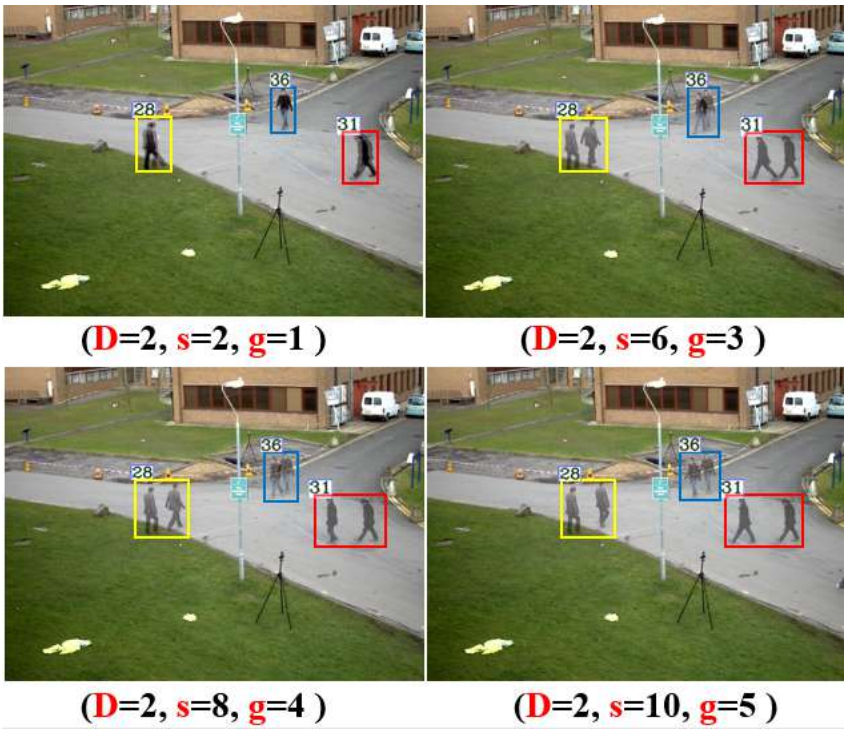


Figure 4.9: Output examples of the interlaced specialized object detector for four interlacing strategies.

The results are provided in Table 4.3. The MOTA for the interlacing strategy (FRCNNVI-MHT) is represented in black while the MOTA for the baseline method (FRCNN-MHT) applied with frame skipping is represented in blue. Please notice that the MHT-DAM association fails from two skipped images. The main reason is that our implementation of the MHT-DAM uses only a spatio-temporal model for tracking, which will fail when there is no overlap between detection. The proposed interlacing strategy produces overlapped detection, which improves the performances of the association.

Table 4.3: MOTA evaluation metric for several interlacing strategies selected to produce frame skipping on TUD dataset. The performance results using the interlaced strategy are presented in black color, and without the interlaced strategy (but with frame skipping) are in blue.

		Interlacing configurations			
Sequence	Method	(D=2,s=2,g=1)	(2, 6, 3)	(2, 8, 4)	(2, 10, 5)
TUD-Stadtmitte	FRCNNVI	**92.8%** |**87.5%**	92.7% | –%	79.8% | –%	74% | –%
TUD-Campus	FRCNNVI	**88.1%** |**79%**	61.7% | –%	30.8% | –%	20.4% |–%
TUD-Crossing	FRCNNVI	**84.8%** |**84.4%**	36.6% |–%	30.3% | –%	21.7% |–%

As mentioned in Table 4.3, for TUD-Stadmitte, the MOTA performance for four skipped frames decreases by about 20% compared to the non skipped frames strategy. Since video-interlacing and reverse video-interlacing are very low CPU time consuming related to the detector, skipping frame strategies provide an efficient way to decrease the computation time of the MOT while maintaining competitive performances.

However, the FRCNN-MHT applied with frame skipping gives bad results (as shown by –% in Table 4.3) due to the limitation of the Faster R-CNN detector to detect objects on crowded sequences (see Figure 4.10).

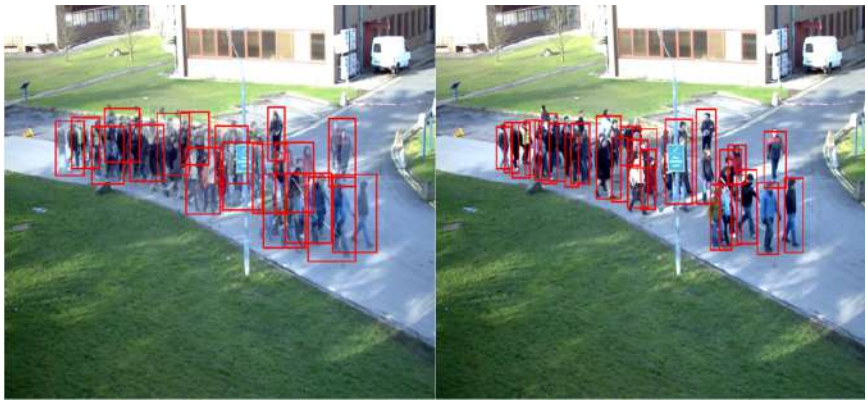


Figure 4.10: Examples of FRCNN-MHT failures (with interlacing strategy (D=2, s=6, g=3)). The reason of that is due to wrong detection provided by specialized Faster R-CNN detector: The left image shows the detection results provided by a specialized interlaced detector and the right one for the Faster R-CNN. In dense crowd, the interlaced video mixes pedestrians resulting to a limitation of the method.

4.5 Conclusion

In this chapter, we have presented a new MOT framework which proposes to built an intermediate interlaced video-sequence and an associated DCNN detector. The resulting MOT algorithm improves the tracking performance. Our suggested tracking framework is generic and can be used with other association algorithms. Moreover, we have demonstrated that some interlacing strategies can be proposed to skip frames and reduce complexity during tracking, while maintaining a good performance.

The proposed framework implies that annotated video sequences have to be available to train the specialized interlaced DCNN.

In the next chapter, we will present the application of our frameworks in the context of traffic surveillance. We will propose an embedded traffic surveillance system which is based on an extension of the SMC Faster R-CNN framework.

Chapter 5

An Embedded Computer-Vision System for Multi-Object Detection in Traffic Surveillance

Contents

In this chapter, we put forward an embedded system for traffic surveillance based on an extension of the SMC framework presented in chapter 3. This applicative contribution consists to analyse traffic and particularly focuses on the problem of detecting and categorizing traffic objects in both day and night conditions. Moreover, it includes a robust detector produced by an original specialization framework. The experiments demonstrate that the proposed system presents encouraging results for multi-traffic object detection and outperforms the state-of-the-art frameworks on several public traffic datasets.

This work was accepted at IEEE Transactions on Intelligent Transportation Systems journal.

This chapter is organized as follows. Section 5.1 presents an introduction and provides the contributions of this chapter. Section 5.2 reviews the existing work

performed in the field of traffic surveillance system and object detection. A discussion about the advantages of our work over the state-of-the-art approaches is given in section 5.3, followed by a detailed description of our approach in section 5.4. The experiments and results are described in section 5.5. The implementation of an embedded system for traffic object detection is discussed in section 5.6. Finally, a conclusion is given in section 5.7.

5.1 Introduction

Intelligent traffic systems for traffic surveillance and monitoring have become a topic of great interest to some cities in the world. Generally, the existing traffic surveillance systems are made up of costly equipments with complicated operation procedures and have difficulties with congestion, occlusion and lighting night/day and day/night transitions. Figure 5.1 illustrates major challenges on traffic applications.

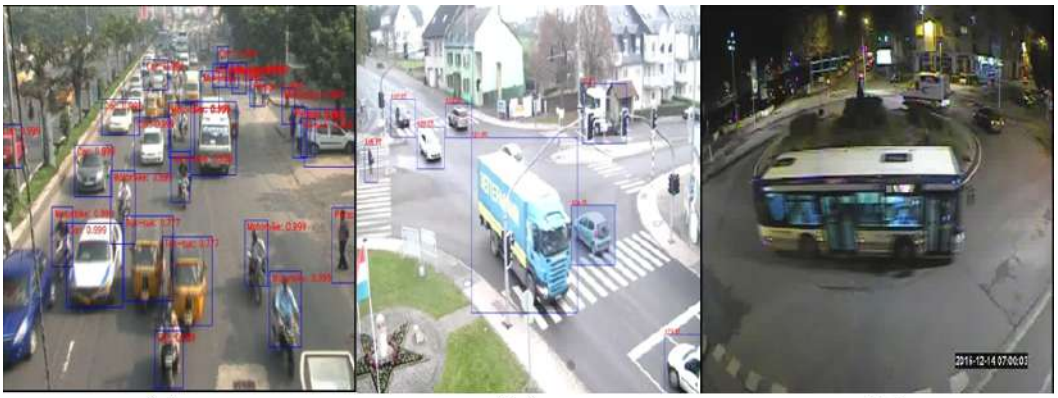
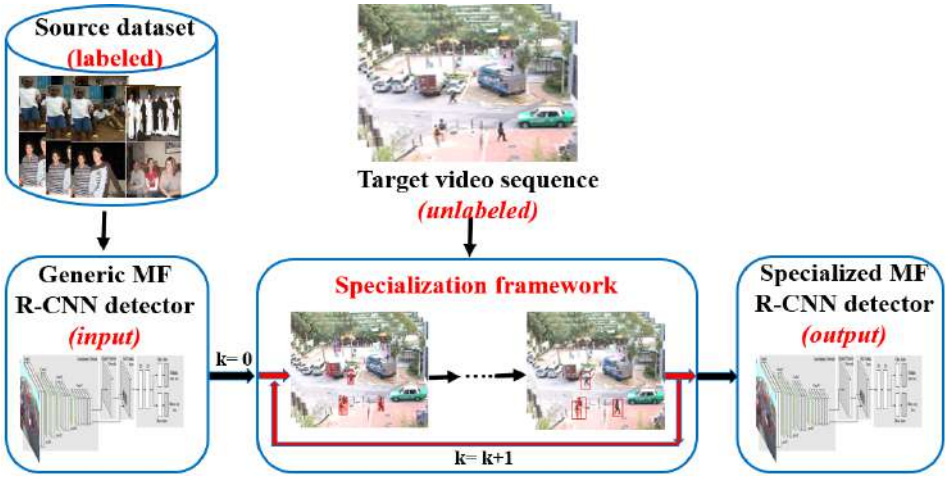


Figure 5.1: Main challenges on traffic applications. (a) presents an example of congestion, (b) an example of occlusion and (c) an example of day/night transitions.

Over the past decade, there has been a significant effort dedicated to the development of traffic surveillance systems, which is intended to raise safety by monitoring the on-road environment. Moreover, numerous sensing modalities have become available for traffic surveillance, including radar, lidar and cameras [Ohn-Bar 2015][Matti 2017]. Concurrently, the computing power has increased dramatically. Besides, we have seen the emergence of computing platforms geared towards parallelization, such as graphical processing units and multi-core processing. Such hardware advances allow the computer vision approaches for traffic surveillance to follow up on real-time implementation. However, the performance of the existing systems depends much on their traffic object detector and it is notable that a traffic surveillance system becomes more reliable if it has a robust detector.

Recently, traffic object detection has been a topic of great interest to researchers [Abdulrahim 2016][Redmon 2016b] and a significant progress has been achieved in



(a)

(b)

(c)

Figure 5.2: (a) General synoptic of specialization framework. Given a generic detector trained by the source labelled dataset and a target video sequence as input, the proposed framework estimates automatically the set of target objects and the parameters of the specialized deep detector. Finally, after a predefined number of iterations, a final specialized detector is generated. (b) and (c) present the improvement of the specialized deep detector in both day and night conditions: left images show the detection results for the generic detector and the right ones for the specialized detector in several datasets.

the recent years [Li 2015b][Maâmatou 2016c][Mhalla 2016b]. Despite the research advances in computer vision, the performance of learning-based traffic object detectors is often limited and decreases significantly when tested on a specific scene due to the large variations between the source training dataset and the target scene. A solution of these problems is to exploit transfer learning approaches. These latter help to build a scene-specialized detector that provides a superior performance than a generic one. Consequently, various transfer learning methods were proposed to develop scene-specific detectors, whose iterative training process is aided by generic detectors to select training samples automatically from target scenes without labelling them manually [Htike 2014][Mao 2015][Maâmatou 2016c][Mhalla 2016a].

This chapter provides an embedded system for traffic surveillance which integrates an extension of our specialization framework by new likelihood function based on tracking algorithm, and a new DCNN detector adopted from the Faster R-CNN deep model for multi-traffic object detection, so as to enhance the detection rate in different scenarios.

A global synoptic of the specialization framework is given in Figure 5.2.(a). A generic deep detector is fine-tuned by a source labelled dataset with labeled information given in the form of traffic-object annotations. Given a target video sequence where labeled information is not available, an iterative process will estimate both the set of target objects and the parameters of the specialized deep detector. This latter is automatically and iteratively trained until a stopping criterion is reached. A final specialized deep detector is then generated, which has performed better than the generic one (see Figure 5.2.(b) and (c)).

The main contributions of this chapter are summarized below:

- A new deep detector for multi-traffic object detection based on the Faster R-CNN deep model
- A new likelihood function based on a tracklet that is used to correctly select unlabeled samples from a specific scene
- An embedded system implemented on an NVIDIA Jetson platforms for traffic surveillance

5.2 Existing work related to video surveillance system and object detection for Intelligent Transportation Systems

This section describes the related work in video surveillance system and in traffic object detection.

The problem of detecting and tracking traffic objects is part of the field of traffic surveillance, which is a subfield of Intelligent Transport Systems (ITS). ITS is known as one of the keys that helps to create the future of the urban world

[Abdulrahim 2016][Sivaraman 2013]. It really attracts a lot of research groups in developing state-of-the-art theories and novel applications.

Several systems have been proposed by research groups in the world to solve the problems of traffic object detection and tracking in a traffic surveillance systems. Some of them have been described in the work of Sivaraman *et al.* [Sivaraman 2013], Neelima *et al.* [Neelima 2012] and Abdulrahim *et al.* [Abdulrahim 2016]. These systems use motion detection to recognize traffic objects as moving blobs and to track those blobs for a number of subsequent frames. Approximately, most of the systems belong to three categories, based on features [Han 2006][Cheng 2006], classifiers [Wu 2007][Huang 2005] and models [Wei Zheng 2013][Jin 2007].

The first category focuses on extracting the specific features by a classical descriptor to represent the traffic object for classification (e.g., HOG [Dalal 2005], Sobel edges [Gao 2010], SIFT [Bay 2008]). Han *et al.* [Wu 2007] took into account the special property of the image patch of a traffic object and proposed to extend HOG features that would incorporate the spatial locality in the standard HOG features [Dalal 2005]. Cheng *et al.*[Cheng 2006] put forward the boosted Gabor features whose parameters were learnt from some samples to give a good response for a traffic object candidate. Zheng *et al.* [Wei Zheng 2013] considered relatively consistent structural components of a traffic object and suggested image strip features that represented various kinds of basic local elements of the traffic object such as bumpers, pillars and wheels.

The second category concentrates on designing the classifier. Wu *et al.* [Wu 2007] proposed the cluster-boosted classifier which was automatically constructed for a multi-view traffic object detection. This method employed an unsupervised clustering to divide the sample space. In the same vein, the similar classifier model was purported by Huang *et al.* [Huang 2005] which needed a predefined knowledge of the intra-class sub-categorization.

The last category focuses on designing the descriptor and the classifier by using DCNN models to detect traffic objects. Li *et al.* [Li 2015a] suggested a generic DCNN traffic-object detector which detected pedestrians with different spatial scales by using a large-size sub-network and a small-size one into a unified architecture. Tian *et al.* [Tian 2015] proposed a novel deep model trained by multiple tasks and datasets to give a robust detector for traffic object detection.

However, the performance of these above methods depends a lot on their training dataset and drops significantly when it is applied to a new scene due to the large variations between the source training dataset and the samples from the target scene. This problem can be resolved by transfer learning which is known as cross-domain adaptation. This latter helps to specialize a generic detector into a specific scene. As mentioned in the chapter 3, transfer learning has become an important topic in computer vision research, mainly in image detection and recognition. Thanks to their superior performance and computation efficiency in object detection, we are incorporate this recent approach in the system presented in this chapter.

Over the past decades, several methods have been suggested for transfer learning in traffic object detection [Mhalla 2016b][Maâmatou 2016c][Ye 2017]. Please refer

to section 3.1 in chapter 3 for more details description of the related transfer learning frameworks.

Addressing this problem with deep learning has recently attracted growing attention by researchers. Several deep models have been suggested in the unsupervised and transfer learning models [Guyon 2011], owing to its robust performance in various tasks like metric learning [Hu 2015] and face recognition [Huang 2012][Taigman 2014]. Moving in this direction, we put forward an extension of the SMC specialization framework based on a new likelihood function, which is developed to automatically generate a robust specialized deep detector for multi-traffic object detection in both day and night conditions. The suggested framework proposes some advantages and several improvements over the related specialization frameworks. In the next section, we discuss the differences and the advantages of the proposed framework related to existing ones.

5.3 Framework proposition

In this section, we present a discussion about the advantages and the differences of our work over the related approaches like the Faster R-CNN deep detector [Ren 2015b] and the SMC specialization approach [Mhalla 2016b]. The proposed deep detector is developed based on the popular Faster R-CNN deep model [Ren 2015b] given its superior performance and computation efficiency in detecting general objects. Table 5.1 presents various differences between proposed deep detector "MF R-CNN" and baseline Faster R-CNN.

Table 5.1: Description of the difference between the Faster R-CNN deep neural network architecture [Ren 2015b] and the MF R-CNN one.

Architecture / Specification	Faster R-CNN [Ren 2015b]	MF R-CNN
Number of pooling layers	4	3
Types of pooling layer	Max-Pooling	Stochastic
Objects	General objects	Traffic objects

We put forward several improvements over the architecture of the Faster R-CNN [Ren 2015b] such that:

- We remove the fourth MAX-Pooling layer in the Faster R-CNN model for training and testing the network to produce larger feature maps for small-size object proposals.
- We replace the Max-pooling layers by Stochastic pooling ones. These latter preserve much more information than the other pooling strategies and provide flexibility in choosing the output image size.

In what follows, we provide a discussion about the advantages of our framework over the state-of-the-art scene specialization frameworks.

Most of the specialization frameworks in the literature have been based on hard-thresholding rules and have been sensitive to the risk of drifting during iterations, or they have been applied only to specific classifiers like the HOG-SVM.

Differently from the existing work and the SMC frameworks [Mhalla 2016b], we propose an iterative framework to specialize any deep detector for traffic object detection. Accordingly, the suggested framework proposes some advantages over the related ones [Maâmatou 2016a][Mhalla 2016b][Maâmatou 2016a][Mhalla 2017], we cite in particular:

- We extend the likelihood function suggested in the SMC framework by utilizing an efficient tracking algorithm based on tracklets, so as to assign a weight for each proposal sample. The tracklet method is used to decrease the risk of detector drifting during iterations by reducing the possibility of introducing wrong labelled examples in the training dataset.
- We can apply our framework to specialize any deep detector for both mobile and stationary cameras. Whereas, the related specialization frameworks can be used only with stationary cameras.
- We show that the proposed framework is able to specialize a traffic object detector in both day and night conditions. In contrast, this is impossible with the related SMC Faster R-CNN framework because it uses a likelihood function based on a background-subtraction spatial-temporal cue to favor the selection of the positive samples from a specific scene; this one does not work with the nighttime conditions.

Table 5.2 provides a comparison over the SMC Faster R-CNN framework proposed in chapter 3 and our suggested specialization one.

Table 5.2: Description of difference between the SMC Faster R-CNN framework and proposed approach

Approach / Specifications	SMC Faster R-CNN	Our approach
Generic detector	Faster R-CNN [Ren 2015b]	MF R-CNN
Likelihood function	Background subtraction	Tracklets
Daytime conditions	Day	Day and night

5.4 Proposed Approach

The suggested specialization framework is an improvement of the SMC Faster R-CNN one (mentioned in chapter 3), which is able to specialize the proposed deep detector toward each traffic scene with a precise classification and in both day and night conditions. The specialized MF R-CNN framework suggests a new architecture of the Faster R-CNN and a novel likelihood function based on a tracklet tracking algorithm.

5.4.1 Architecture of proposed detector

Figure 5.3 illustrates the architecture of the MF R-CNN in details. The MF R-CNN architecture is a single, unified network for traffic-object detection which is inspired from the Faster R-CNN one [Ren 2015b]. According to this latter, the proposed detector is composed of two modules. The first module is an RPN that provides a set of rectangular object proposals from an input image. The second module is the Fast R-CNN deep model [Girshick 2014b] which takes as inputs this set of object proposals and then uses them for classification. For more details, an RPN is a fully-convolutional network that is constructed by adding two additional convolutional layers: one that encodes each convolutional map position into a short feature vector and another that outputs, at each convolutional map position, an objectness score and regress bounds for the region proposals relative to various scales and aspect ratios at that location. The RPN shares the rest of convolutional layers with the Fast R-CNN network.

According to Figure 5.3, the MF R-CNN passes the input image into several convolutional layers and stochastic pooling ones (shared layers) to extract a feature map. Then the RPN fully-convolutional network, inspired from [Ren 2015b], is learned specifically to localize traffic objects into the feature map produced by the last convolutional layer of the pre-trained VGG16 model [Simonyan 2014].

After that, the RoI pooling layer (inspired from the Fast R-CNN network) is utilized to pool the feature maps of each input object proposal which is fed into a sequence of fully connected layers, into a fixed-length feature vector. The network finishes with two output layers that produce two output vectors per object proposal. Specially, one layer outputs classification scores over a T traffic-object class plus a "background" class. The other one is the bounding box regressor which outputs refined bounding-box positions for each T class.

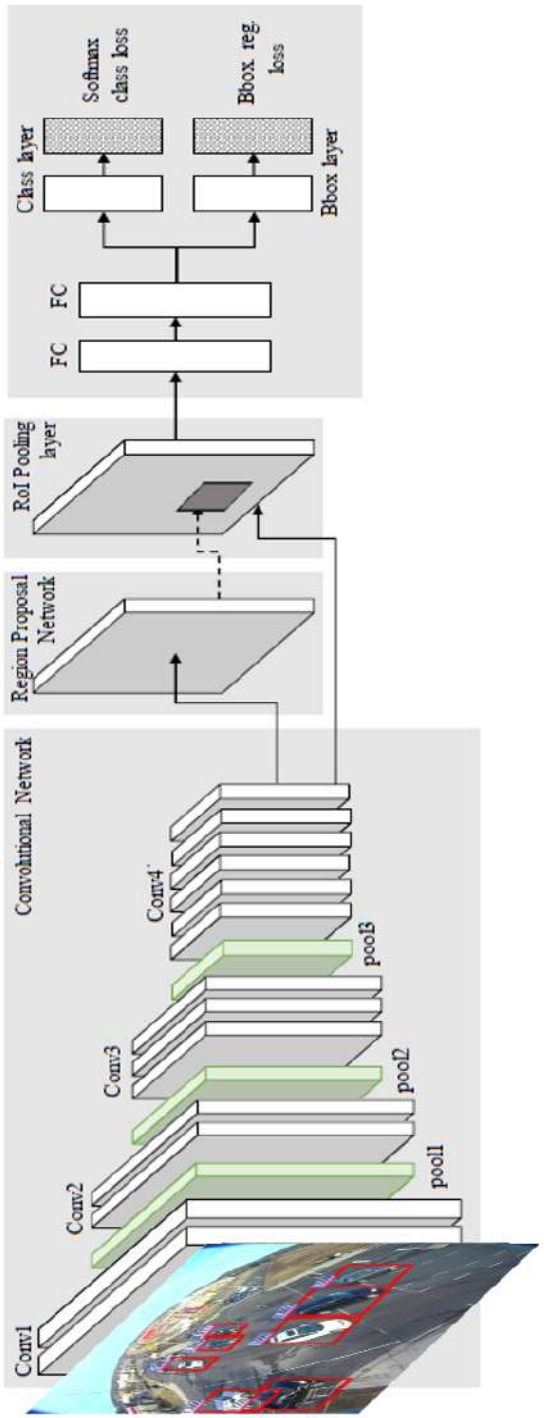


Figure 5.3: Architecture of the suggested MF R-CNN deep detector. This latter contains a sequence of convolutional layers and stochastic pooling ones to extract a feature from the whole input image. After that, a RoI pooling layer pools the produced feature maps for each region proposal generated by the RPN into a fixed-length feature vector. Then a sequence of fully connected layers finished with two output layers are performed to generate detection results: One outputs classification scores over j traffic-object classes plus a "background" class and the other outputs refined bounding-box positions for each j class.

5.4.2 Specialization of the MF R-CNN

This section presents the specialization of the MF R-CNN deep detector towards a target scene.

The diagram in Figure 5.4, demonstrates the specialization steps based on the SMC Faster R-CNN framework. First, a generic detector is trained on a generic dataset. Given the videos taken by a stationary or mobile cameras in specific scenes, at a first iteration ($k = 0$), the generic detector is applied in a prediction step to detect traffic object candidates in each individual image, which may include a lot of positive and negative detections. Then a likelihood function is applied based on a tracklet tracking algorithm in the update step, which is used to associate a weight to each proposal sample from a specific scene. Then a sampling step determines which samples should be included in the specialized dataset by using an IR algorithm (detailed in chapter 3) inspired from the theory of the Monte Carlo filter [Doucet 2001]. The IR algorithm transforms each weight produced by the likelihood function in the previous step on a number of repetitions, by repeating the samples associated to a high weight by numerous ones and replacing the samples linked to a low weight by few ones. At the training step, a new specialized detector is fine-tuned by the specialized dataset, and it will become the input of the prediction step in the next iteration. The scene-specific detector is automatically and iteratively trained and is called until a stopping criterion is reached. Please refers to chapter 3 for more details about the SMC steps for specialization.

In what follows, we will describe the new proposed likelihood function based on the tracklet tracking algorithm.

5.4.3 Likelihood function

In order to choose the correct proposal, we put forward a likelihood function based on a tracking method, which assigns a weight $\pi_k^{(n)}$ for each sample $\tilde{\mathbf{x}}_k^{(n)}$ returned by the prediction step. The aim of this function is to favor the selection of the correct samples and reduce the risk of including wrong proposal samples in the specialized dataset. The output of this function is a set of weighted target samples that approximates the posterior probability function, according to equation (5.1):

$$\pi_k^{(n)} = f_L(\tilde{\mathbf{x}}_k^{(n)}) \tag{5.1}$$

where f_L is the likelihood function and $\pi_k^{(n)}$ is the weight assigned to each sample $\tilde{\mathbf{x}}_k^{(n)}$.

The likelihood function is based on a tracking method called "tracklet", to assign weights to target samples according to their importance. The tracklet tracking algorithm used in the proposed likelihood function is inspired from [Dorai 2016].

Based on the set of samples produced by the prediction step, we are able to track/link them into tracklets (a tracklet is a chain of samples belonging to the same object over time) by adopting the multi-object tracking algorithm in [Dorai 2016]. This association-based tracking provides tracklets to favor the selection of positive

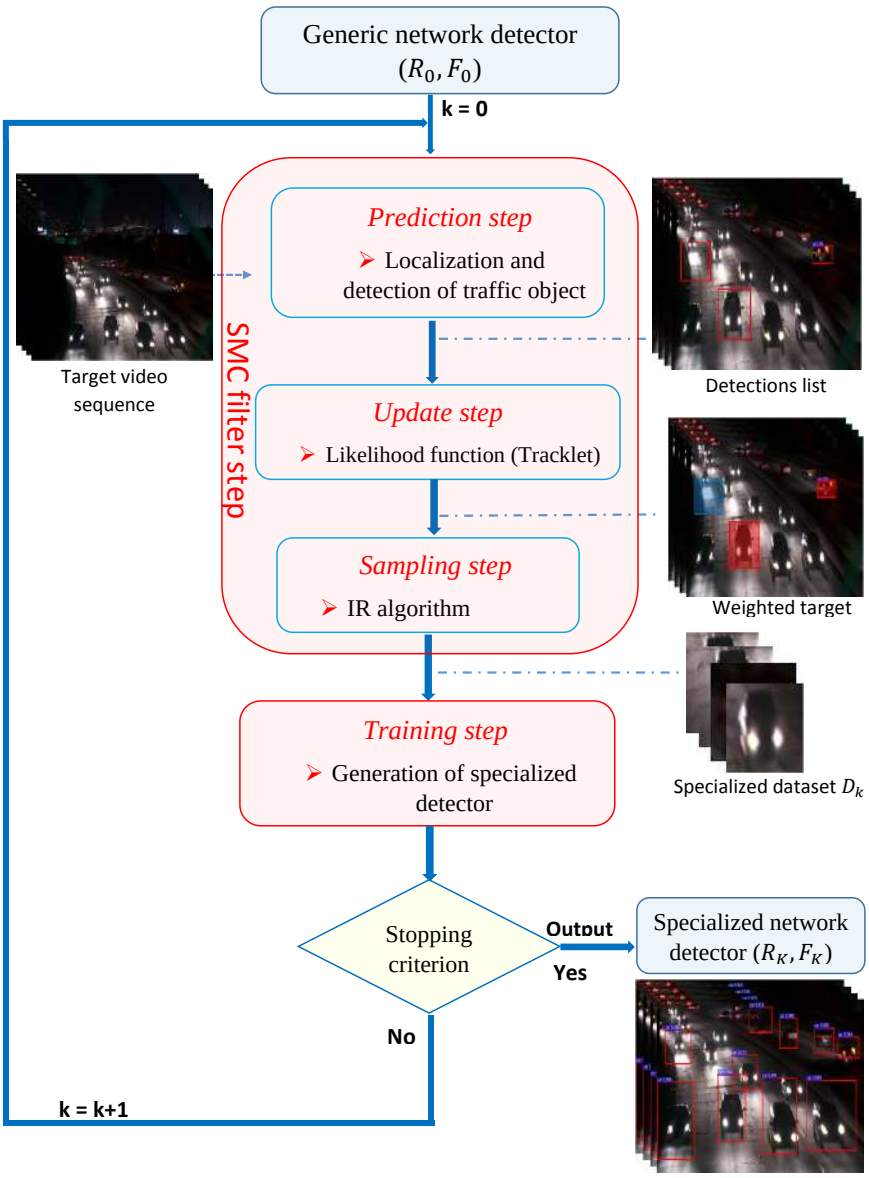


Figure 5.4: Block diagram of proposed approach: At the first iteration, our generic detector $(\mathcal{R}_0, \mathcal{F}_0)$ which is fine-tuned by the source dataset is utilized in the first prediction step to produce a list of detections from the target scene, and then a likelihood function based on tracklets in the update step is used to favor the selection of positive samples from a specific scene. The sampling step determines which samples will be included in the specialized dataset by using an IR algorithm. A new specialized detector $(\mathcal{R}_k, \mathcal{F}_k)$ is fine-tuned by the specialized dataset in the training step, which will become the input of the prediction step in the next iteration $k = k + 1$. A final specialized detector $(\mathcal{R}_K, \mathcal{F}_K)$ is generated when a number of iterations is reached. The red rectangles in the output image of update step mean that samples are selected by our suggested likelihood function and the blue ones mean that samples are removed.

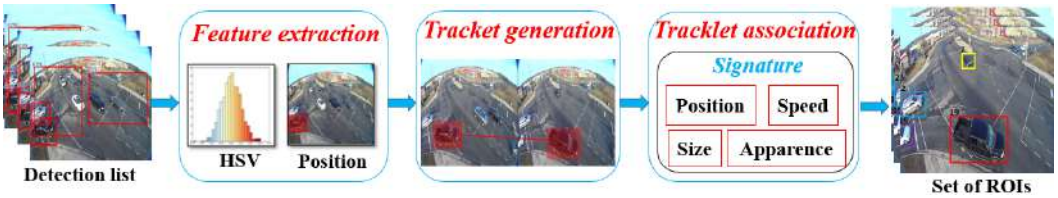


Figure 5.5: Description of tracklet steps. Given a set of samples, a feature extraction block allows to define the characteristics of each sample. The latter is characterized by a position and an appearance information determined by a color histogram (HSV). After that, a tracklet generation block is used to construct initial tracklets by association of samples. The association between the samples is done according to calculate IoU overlap and appearance similarity between samples in successive frames. Next, after initial tracklet constructions, we associate the tracklets having similar signatures. A signature contains the characteristics of appearance, position, speed and size. The output of the association step is a set of RoI associated to target objects.

samples. The tracklet tracking method is divided into three main steps: extraction of features, tracklet generation and tracklet association. The details of the three main steps are described in the following three points.

5.4.3.1 Feature extraction

According to Figure 5.5, each target sample produced by the prediction step is passed through the extraction feature block to define the characteristics of each sample. The latter is characterized by a position produced by the output layer (bounding box regressor layer) of our MF R-CNN detector and the characteristic vector that contains appearance information determined by a color histogram (HSV).

5.4.3.2 Tracklet generation

After feature extraction step, initial tracklets (object trajectories) are constructed by association of samples. The association between the samples is done according to the IoU overlap and the appearance similarity between the successive frames. Subsequently, the IoU overlap is calculated by comparing the bounding boxes of samples in successive frames. After that, we compare the appearance similarity between the overlapped samples in successive frames. Accordingly, the appearance similarity is provided by calculating the distance between the two HSV histogram vectors associated to the overlapped samples.

5.4.3.3 Tracklet association

After initial tracklet constructions, we associate the tracklets having similar signatures. In addition, a signature contains the characteristics of speed determined by a

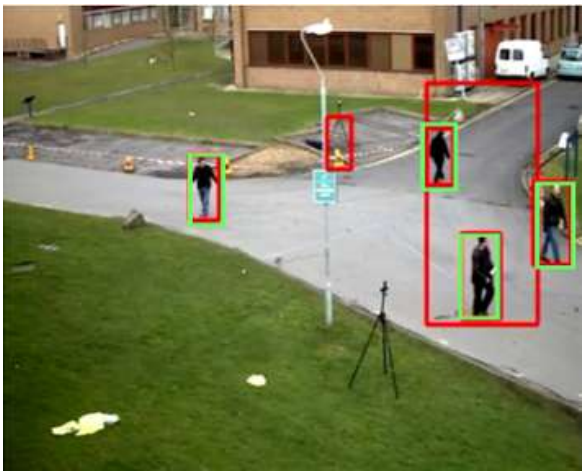


Figure 5.6: The red rectangle presents the area in pixels of the considered RoI, the green rectangle is the area generated by the tracklet tracking algorithm.

Kalman filter [Lee 2004], appearance information determined by a color histogram (HSV), position and size of each tracklet.

The output of the association step is a set of RoIs associated to target objects (as depicted in Figure 5.5), each RoI is defined by: $\{a_k^{(n)}, b_k^{(n)}, c_k^{(n)}, d_k^{(n)}, id\}$, where $(a_k^{(n)}, b_k^{(n)})$ are the upper left coordinates of the RoI, $(c_k^{(n)}, d_k^{(n)})$ are the width and the height of the RoI box and id is the identifier of object. To assign a weight for each sample, we calculate an overlap (equation 5.2) that compares the RoI associated to one sample in the frame n with the outputs of the association step in that frame.

$$\pi_k^{(n)} \doteq \frac{2(RoI\_AR \times RoI\_AR1)}{RoI\_AR + RoI\_AR1} \tag{5.2}$$

where RoI_AR is the area in pixels of the sample $\tilde{\mathbf{x}}_k^{(n)}$ in the frame n and RoI_AR1 is the area of each RoI generated by the tracklet tracking algorithm in that frame (see Figure 5.6).

The likelihood function assigns a high weight to a positive samples if it has a weight value $\pi_k^{(n)}$ that exceeds a fixed threshold α_p, which is determined empirically. Otherwise, it will be associated to zero (as shown in equation 5.3).

$$\pi_k^{(n)} = \begin{cases} \pi_k^{(n)} & \text{if} \quad \pi_k^{(n)} \geq \alpha_p \\ 0 & \text{if} \quad \pi_k^{(n)} < \alpha_p \end{cases} \tag{5.3}$$

Considering the proposed likelihood function, the selection of samples associated to the right label and the removal of the false samples become efficient. The output of this function is a set of weighted target samples.

5.5 Experiments

In this section, we evaluate the effectiveness of the proposed specialized framework with the relevant ones on several public and private datasets including the CUHK [Wang 2012a], Logiroad [Mhalla 2016b] and Traffic Night datasets.

5.5.1 Datasets

The PASCAL VOC 2007 dataset [Everingham 2010] is utilized to train the proposed generic MF R-CNN. In the experiments, we use 713 annotated cars and 2,008 people, to fine-tune the generic MF R-CNN detector.

The evaluation is achieved on three datasets:

- **CUHK Square dataset** [Wang 2012a]: This is a public video sequence of road traffic which lasts 60 minutes. 352 images are used for specialization, uniformly extracted from the first half of the video. 100 images are utilized for the test, extracted from the latest 30 minutes. The annotations are provided by Wang [Wang 2012a] for pedestrian detection (noted CUHK_WP). However, we notice that some strong annotation errors were made in the public ground truth, so we use the annotation provided by Mhalla [Mhalla 2016b] (noted CUHK_MP).

- **Logiroad Traffic dataset** [Mhalla 2016b]: This is a private video sequence of road traffic which lasts 20 minutes. We use 600 images for specialization, extracted uniformly from the first 15 minutes of the video. 100 images are utilized for the test, extracted from the latest 5 minutes. the annotations are available for cars (Logiroad_MV)

- **Traffic Night dataset:** This is a private video sequence of road traffic at nighttime which lasts 4 minutes. We use 300 images for specialization, extracted uniformly from the first 3 minutes of the video. 100 images are utilized for the test, extracted from the last minute. The annotations are available for cars (noted NightRoad_MV)

5.5.2 Implementation details

We use the pre-trained VGG16 deep network [Simonyan 2014] to initialize the MF R-CNN, which has been used in several state-of-the-art detection approaches [Girshick 2015b][Ren 2015b]. The first seven convolutional layers and the three max pooling layers of the VGG16 network are used as shared convolutional layers to produce feature maps from the entire input image. The remaining layers of the VGG16 network are used to initialize the MF R-CNN. The fourth max pooling layer is removed to produce larger feature maps. We change the rest of the max pooling layers by stochastic pooling ones.

Following the Faster R-CNN network [Ren 2015b], the last max pooling layer of the VGG16 network is replaced by the RoI pooling layer to pool the feature maps

of each object proposal into a fixed resolution (7*7). The final fully connected layer and softmax are replaced with two sibling fully-connected layers. The MF R-CNN is trained with a stochastic gradient descent with a momentum of 0.9 and a weight decay of 0.0005. The MF R-CNN is implemented based on the publicly available Caffe platform [Jia 2014]. For training, the first four convolutional layers in the network keep constant parameters initialized from the pre-trained VGG16 model.

For the training parameters and the number of iterations of the specialization process, we utilize the same ones detailed in chapter 3. Please refer to section 3.4.1 in chapter 3 for more implementation details.

5.5.3 Evaluated algorithms

Performance evaluation is done in terms of recall False Positives Per Image (FPPI) curves. The PASCAL 50 percent overlap criteria [Everingham 2010] are used to give a score for the detection bounding boxes. The specialization framework is compared with several state-of-the-art ones:

- Wang (2014) [Wang 2014b]: A specific-scene detector was trained on only relevant samples chosen from both source and target datasets.
- Htike (2014) [Htike 2014]: A non-iterative specialization framework was used to specialize a pedestrian detector to video scenes.
- Mao (2015) [Mao 2015]: A specialization framework was proposed to automatically train scene-specific pedestrian detector based on tracklets.
- Faster R-CNN (2015) [Ren 2015b]: A generic detector used a deep convolutional network for both localization and detection of general objects.
- Maamatou (2016) [Maâmatou 2016c]: A specialized framework was applied to specialize a generic HOG-SVM classifier to a specific video sequence for detecting traffic objects.
- SMC Faster R-CNN (2016) [Mhalla 2016b]: A specialization framework was based on the SMC filter to specialize a generic Faster R-CNN detector.
- Generic MF R-CNN: It is the proposed detector which is fine-tuned on the generic dataset. This is the baseline for our comparison.

5.5.4 Results and analysis

Given each annotation dataset, we present the ROC curves of the generic MF R-CNN, the specialized MF R-CNN and the available state-of-the-art algorithms. For comparison, two synthetic tables are given: Table 5.3 for pedestrian detection and Table 5.4 for car detection. The true detection rate is compared to the constant FPPI in several methods related to several datasets and annotations. Furthermore, the two last lines of both tables give the improvement between the generic MF R-CNN and the generic Faster R-CNN, and the second one gives the improvement

between the generic MF R-CNN and the specialized one. The latter has a better detection rate than the generic detector in all the achieved experiments. The median improvement is **40%** in all traffic datasets.

- **Comparison between generic Faster R-CNN and generic MF R-CNN:**

  Table 5.4 and the ROC curves in Figure 5.8 show that the suggested generic MF R-CNN detector outperforms the generic Faster R-CNN on all public and private datasets with several annotations. The median improvement is **20%**.

- **Comparison with state-of-the-art specialization frameworks:**

  According to Table 5.4, for the CUHK pedestrian detection, the specialized MF R-CNN outperforms all the other state-of-the-art specialization frameworks. Besides, the detection rate achieved with Mhalla [Mhalla 2016b] annotations on CUHK_MP is nearly 90% for 0.5 FPPI. However, despite the wrong annotations given by Wang (CUHK_WP in Table 5.4), the specialized MF R-CNN also exceeds the six other specialized detectors of Wang (2014), Htike (2014), Mao (2015), Maamatou (2016) and Mhalla (2016) respectively by 45%, 49%, 58%, 62% and 65%. For the Logiroad car detection (Table 5.4) with the annotations given by [Mhalla 2016b], the specialized MF R-CNN is ranked first and exceeds the SMC Faster R-CNN [Mhalla 2016b] and the specialized detector suggested by Maamatou (2016).

Table 5.3: Comparison of detection rate for pedestrian with state of art (at 0.5 FPPI)

Approach \ Dataset	CUHK_WP	CUHK_MP
Wang [Wang 2014b]	0.45	–
Htike [Htike 2014]	0.49	–
MAO [Mao 2015]	0.58	–
Generic Faster R-CNN [Ren 2015b]	0.60	0.69
Maamatou [Maâmatou 2016c]	0.62	0.58
SMC Faster R-CNN [Mhalla 2016b]	0.65	0.88
Generic MF R-CNN	**0.71**	**0.74**
Specialized MF R-CNN	**0.75**	**0.90**
Improvement/MF R-CNN & Faster R-CN	18%	6%
Improvement/Generic MF R-CNN	6%	22%

Figure 5.7 shows the improvement of the specialized MF R-CNN detector in detecting small-sized traffic objects and in removing the false positive samples compared to the SMC Faster R-CNN one [Mhalla 2016b].

One can notice after multiple experiments that the generic MF R-CNN, fine-tuned on the PASCAL VOC 2007 dataset, has a poor detection rate resulting in a limitation of traffic-object annotations.

Table 5.4: Comparison of detection rate for car with state of art (at 1 FPPI)

Approach \ Dataset	Logiroad_MV
Generic Faster R-CNN [Ren 2015b]	0.40
Maamatou [Maâmatou 2016c]	0.47
SMC Faster R-CNN [Mhalla 2016b]	0.70
Generic MF R-CNN	**0.48**
Specialized MF R-CNN	**0.75**
Improvement /MF R-CNN & Faster R-CNN	20%
Improvement /Generic MF R-CNN	57%

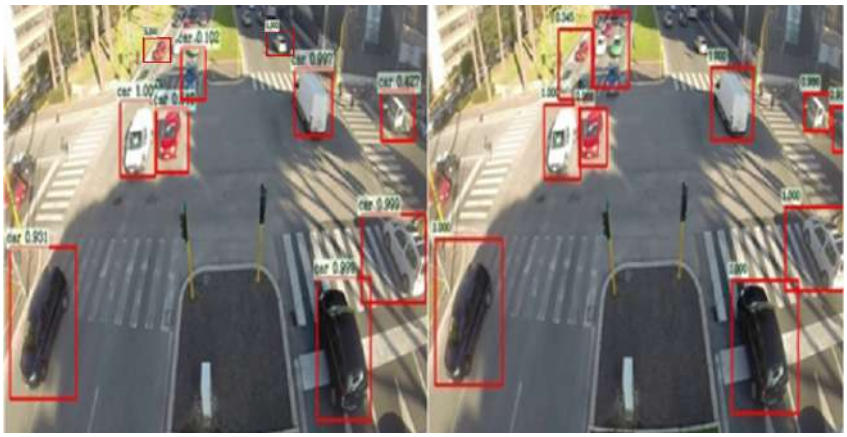


Figure 5.7: Improvement of our proposed specialization framework in detecting small-sized objects and in removing the false positive samples on the Logiroad Traffic dataset: The left image shows the detection results for the specialized MF R-CNN detector and the right one for the SMC Faster R-CNN[Mhalla 2016b].

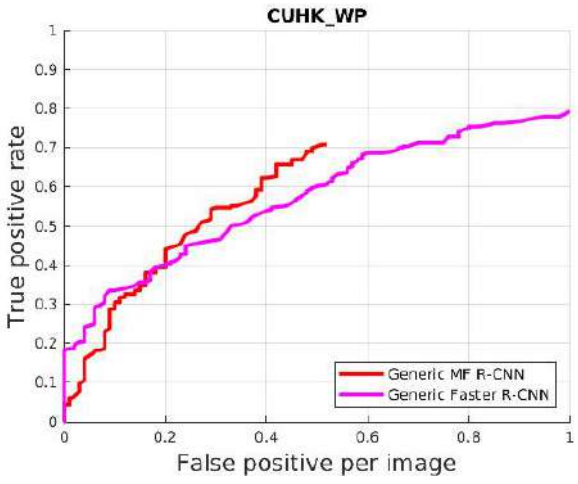
- **Effect of likelihood function:**

  The results demonstrate that the proposed likelihood function based on the tracklet tracking algorithm improves the detector performance and accelerates the convergence of the specialization process. Figure 5.9 presents the efficiency of the suggested likelihood function to correctly select positive samples from a target scene. Furthermore, we cannot say that this choice is the best because it is possible to improve the likelihood function with more spatio-temporal information like contextual information or optical flow.

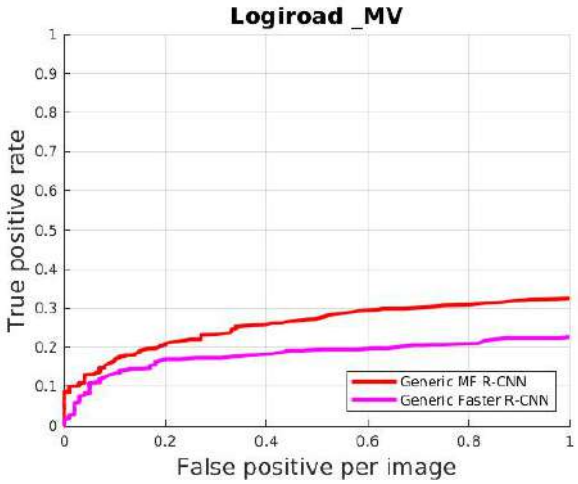
5.5.5 Results and analysis in nighttime conditions

Table 5.5 provides the working of our specialization framework in nighttime conditions with a superior performance on the Traffic Night dataset.

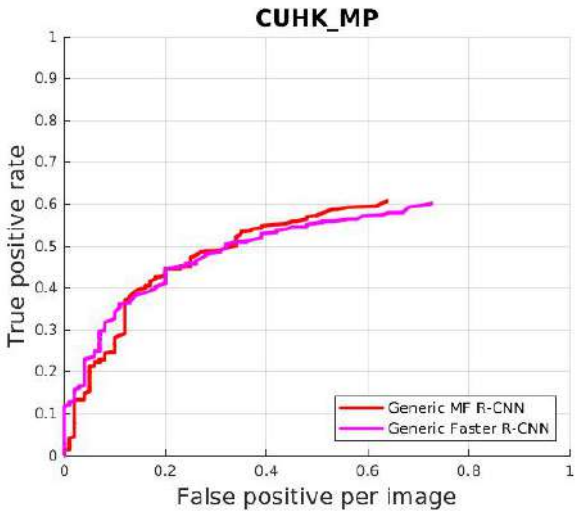
The specialized MF R-CNN outperforms the generic one, and the median improvement is 57%.



(a)



(b)



(c)

Figure 5.8: ROC curves for comparison between generic Faster R-CNN detector (magenta curves) and proposed MF R-CNN one (red curves), performed on (a) CUHK_WP, (b) Logiroad_MV and (c) CUHK_MP datasets.

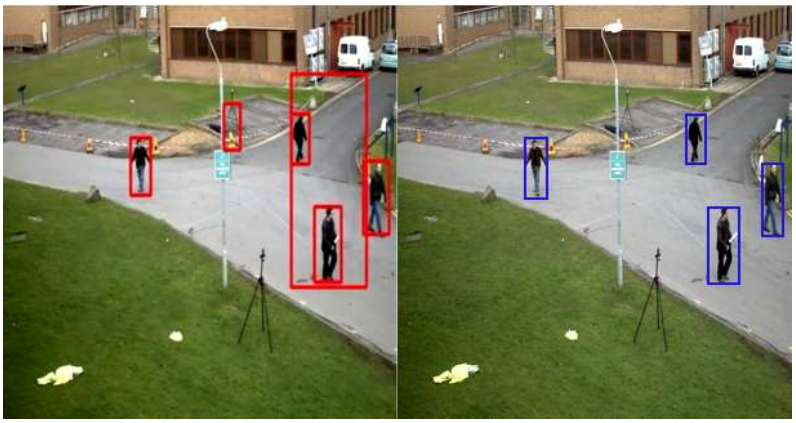


Figure 5.9: Efficiency of proposed likelihood function. The red blobs in the left image present the inputs of the likelihood function and the blue ones in the right image are the outputs.

Table 5.5: Comparison of detection rate for Traffic Night dataset with state of art (at 1 FPPI)

Approach \ Dataset	NightRoad _MV
Generic Faster R-CNN [Ren 2015b]	0.27
SMC Faster R-CNN [Mhalla 2016b]	—
Generic MF R-CNN	**0.38**
Specialized MF R-CNN	**0.60**
Improvement /MF R-CNN & Faster R-CNN	40%
Improvement /Generic MF R-CNN	57%

5.6 Proposed embedded system

This section details the implementation of our proposed system "Traffic system" for multi-traffic object detection.

Due to the performance and computation efficiency of the multi-traffic object detector generated by our specialization framework, we propose to deploy our framework on a small and power efficient device like the NVIDIA Jetson embedded platform with a powerful GPU onboard, in order to provide an embedded system for traffic surveillance. In this section, we analyze the Jetsonâs suitability by benchmarking the run-time of our specialized detector in comparison to a high performance GPU. Exemplary, we port to this platform an embedded traffic system for Traffic analysis and particularly focuses on the problem of detecting and categorizing traffic objects on several traffic scenes.

For the hardware components (Figure 5.10) of the proposed embedded system, we use the recent mobile device, the Tegra TX2 board. The Tegra TX2 device is a technology developed by NVIDIA in the embedded system category. This device delivers the performance required for the latest visual computing applications. It is

built around an NVIDIA Pascal-family GPU and loaded with 8 GB of memory and 59.7 GB/s of memory bandwidth , 64-bit CPUs, and a camera with 5 mega pixels. Table 5.6 provides more technical specification of the hardware components.

Table 5.6: Technical specification details of hardware components

Component	Specification
Camera	5 mega pixel
GPU	NVIDIA Pascal, 256 CUDA cores
CPU	HMP Dual Denver 2/2 MB L2 + Quad ARM A57
Operating system	Ubuntu Linux 16.04 LTS
Memory	8 GB 128 bit LPDDR4 59.7 GB/s
CSI	Up to 6 cameras | 1400 Mpix/s
Connectivity	Connects to 802.11ac Wi-Fi
Networking	1 Gigabit Ethernet
Storage	16 GB eMMC, SDIO, SATA

To run our specialized deep detector on the NVIDIA Jetson TX2 device, we use the Caffe deep learning framework [Jia 2014] compiled for GPU and Python programming language. The specialized MF R-CNN network is able to recognize up to 4 classes of traffic objects including pedestrians, cars, buses and motorbikes.

In this part, we explore the running of the DCNN architectures including : VGG16 [Simonyan 2014], ZF [Zeiler 2014] and CNN_M_1024 [Chatfield 2014] for multi-object detection on the NVIDIA Jetson TX2 embedded platform. In Table 5.7, we summarize the running time of our suggested specialized detector on NVIDIA Jetson TX2 and the performance of different deep architectures explored in our work. According to Table 5.7, to run our embedded system with the NVIDIA Jetson

Table 5.7: Description of running our specialized detector on the NVIDIA Jetson TX2 through different deep architectures

Specification \ Architecture	VGG16	ZF	CNN_1024
Running time	—	**5 fps**	2.5 fps
Performance (mAP%)	85.1	83.3	69.2

TX2, we choose the ZF architecture for the suggested deep detector thanks to its performance and speed on object detection. As mentioned in Table 5.7, the NVIDIA Jetson TX2 embedded platform can not run with the VGG16 deep architecture.

In order to obtain a traffic surveillance system which runs in real-time, we specialize other recent detectors like SSD 300 [Liu 2016] and SSD with MobileNet architecture [Howard 2017]. After that, we deploy these latter on the NVIDIA embedded platform. Table 5.8 summarizes the obtained results.

According to Table 5.8, we illustrate that our proposed embedded system can work on real-time with a specialized MobileNet-SSD detector (**20 fps**).

The resulting product provides an intelligent system for monitoring and securing transport infrastructure. It is composed of several processing blocks, allowing both

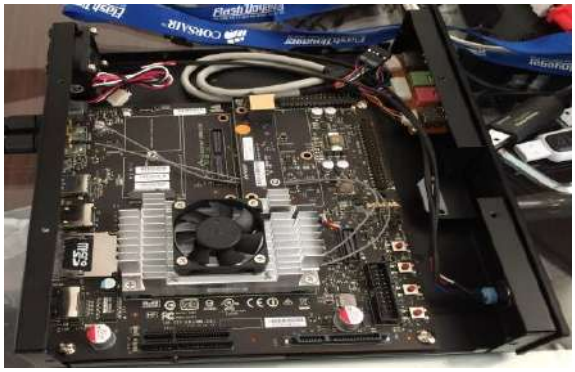


Figure 5.10: Image of hardware components of proposed embedded system

Table 5.8: Description of running specialized detectors on the NVIDIA platform.

Specification \ Detector	MobileNet-SSD	MF R-CNN	SDD 300
Running time	**20 fps**	5 fps	9 fps
Performance (mAP%)	85.5	83.3	80.0

the analysis and automatic interpretation of observed scenes as well as a self-decision system. We mainly target our system toward transport infrastructure (road, highway ...) and intelligent vehicles.

5.7 Conclusion

In this chapter, we have put forward an embedded system for multi-object detection in traffic surveillance, which includes a new architecture of a deep detector adopted from the Faster R-CNN and an extension of the SMC specialization framework for a traffic object detector. Given a generic detector and a target video sequence, this framework automatically provides a specialized traffic-object detector. The extensive experiments have demonstrated that the proposed approach has produced a robust traffic object detector which is superior in detecting traffic objects in different scenes and in both day and night conditions. This detector has surpassed the state-of-the-art performance on several challenging benchmarks.

Chapter 6

Conclusion and perspectives

Conclusion

In this PhD thesis, we have been interested in video sequence analysis; particularly, we have been focused on the problem of detecting and tracking multi-objects in video sequences. This thesis integrates three main contributions.

First, we have presented a transfer learning contribution based on the formalism and the theory of the SMC filter to specialize a DCNN detector to a particular scene, taking into account the advantages of deep learning and transfer learning as well as the need for specialization of a deep detector. Given the significant drop in the performance of a generic detector when applied to a specific scene due to the large variation between the source training dataset and the target scene, we have proposed an automatic specialization framework in order to specialize a generic deep detector.

The suggested framework uses different strategies based on the SMC filter steps to approximate iteratively the target distribution as a specialized dataset composed of samples from the target domain. These samples are selected according to their importance of weights, reflecting the likelihood that they belong to the target distribution. Actually, the specialized dataset is used to fine-tune a DCNN detector to increase the detection performance in the target scene.

Furthermore, the suggested framework uses a likelihood function which utilizes an efficient combination between the information given by the output layer of the DCNN model and spatio-temporal information extracted from the target sequence, to favor the weighting of target samples associated to the right label. This function permits decreasing the risk of introducing wrong labelled examples in the specialized dataset and accelerating the convergence of the specialization process.

Given a generic detector and a target video sequence, the proposed framework automatically provides a robust specialized detector. The experiments have shown that the performance of the specialized detector outperforms the generic one and the state-of-the-art performance on three challenging datasets for multi-object detection. Moreover, the suggested framework is a generic transfer learning framework in which many strategies can be integrated in the SMC steps to increase the detection accuracy. Also, the proposed framework is generic and can be applied with any DCNN detector.

In addition, some improvements can be applied to ameliorate the suggested framework by proposing other strategies for the SMC steps. For example, the likeli-

hood function can be improved with more complex visual cues like the optical flow or the contextual information to enhance the weighting of positive samples.

The second contribution is focused on tracking multi-objects based on exploiting the concept of interlaced videos. The proposed MOT framework is based on an interlacing strategy, which is utilized to combine frames with different interlacing configurations, and an interlaced DCNN detector, which is specialized to detect objects in interlaced images. In particular, by interlacing multiple frames, a DCNN detector is encouraged to exploit an implicit temporal cue, despite processing a single input at a time. Moreover, some interlacing strategies can be proposed to skip frames and reduce complexity during tracking, while maintaining good performance.

The MOT framework consists in improving the tracking performances and to operate under tracking challenges such as occlusion, intersection and congestion. The proposed framework implies that annotated video sequences have to be available to train the specialized interlaced DCNN. Nevertheless, the results demonstrate that the performance of the MOT framework outperforms the baseline one and the state-of-the-art tracking performance on several challenging datasets. The suggested tracking framework is generic and can be utilized with different association algorithms.

The last contribution is considered as an applicative contribution that presents an extension of the SMC specialization framework with a new verification strategy based on a tracking algorithm, which is used to correctly select positive samples from the target scene, and a new DCNN detector adopted from the Faster R-CNN model. The proposed improvements over the related SMC framework is to extend the likelihood function suggested in the SMC framework by utilizing an efficient tracking method based on tracklets, so as to assign a weight for each proposal sample. The tracklet method is used to decrease the risk of detector drifting during iterations by reducing the possibility of introducing wrong labelled examples in the specialized dataset. Furthermore, we put forward a new deep detector inspired from the Faster R-CNN detector which performs better in detecting small-sized objects.

Moreover, we can apply our framework to specialize detectors for both mobile and stationary cameras and in both day and night conditions. On the contrary, the related SMC specialization framework can be used only with stationary camera and only in day time due to the limitation of its likelihood function which is based on a classical background subtraction algorithm.

Extensive experiments have demonstrated that the proposed contribution has produced a robust traffic object detector in different scenes and in both day and night conditions. This specialized detector has surpassed the generic one and the state-of-the-art performance on several challenging benchmarks. Taking into account the performance of the new specialized detector in several traffic scenes, we have deployed the latter in an embedded system for traffic surveillance applications.

Perspectives

Based on the obtained results, the proposed automatic frameworks allow improving the detection and tracking performances in a specific scene without human intervention. These frameworks are generic and can be used with any DCNN detector and with any association algorithms. However, several perspectives can be envisaged.

As mentioned in chapter 4, the suggested MOT framework implies that annotated video sequences have to be available to train the specialized interlaced DCNN. Accordingly, if we want to apply the MOT framework on new videos, we must find a way to generate these annotations automatically.

As a first perspective, we will propose an automatic specialization system using domain adaptation algorithms for the MOT. The main idea consists in utilizing the SMC specialized framework to automatically generate the annotated input tracking video to perform the interlacing dataset for training the interlaced deep detector. A global synoptic of the suggested automatic specialization system for the MOT is illustrated in Figure 6.1. The proposed system includes two parts: The first one aims to automatically generate the input video annotations by exploiting the SMC framework for interlaced detector specialization. The second part of the proposed system is the MOT tracking framework provided in chapter 4.

As a second perspective, we propose to study the parameters of each stage of the suggested specialization framework, a specifically the update and sampling stages, so as to improve the performance of specialization and accelerate its convergence. Accordingly, we will deal with an extension of the framework to improve the likelihood function by using a new strategy of verification based on more complex visual cues like the optical flow or the contextual information. Besides, we will study the possibility of injecting some spatio-temporal information into the Faster R-CNN network in order to enhance the detection performance.

As a third perspective, we will study the interlacing configurations in the tracking framework to reduce the computation time and to maintain a good performance.

In our work, we have studied the scientific contributions of multi-object specialization in the detection and tracking tasks. One more perspective is to perform the study of a set of components of our chain that influence time and accuracy information, so as to improve the stability, precision and speed of our proposed chain.

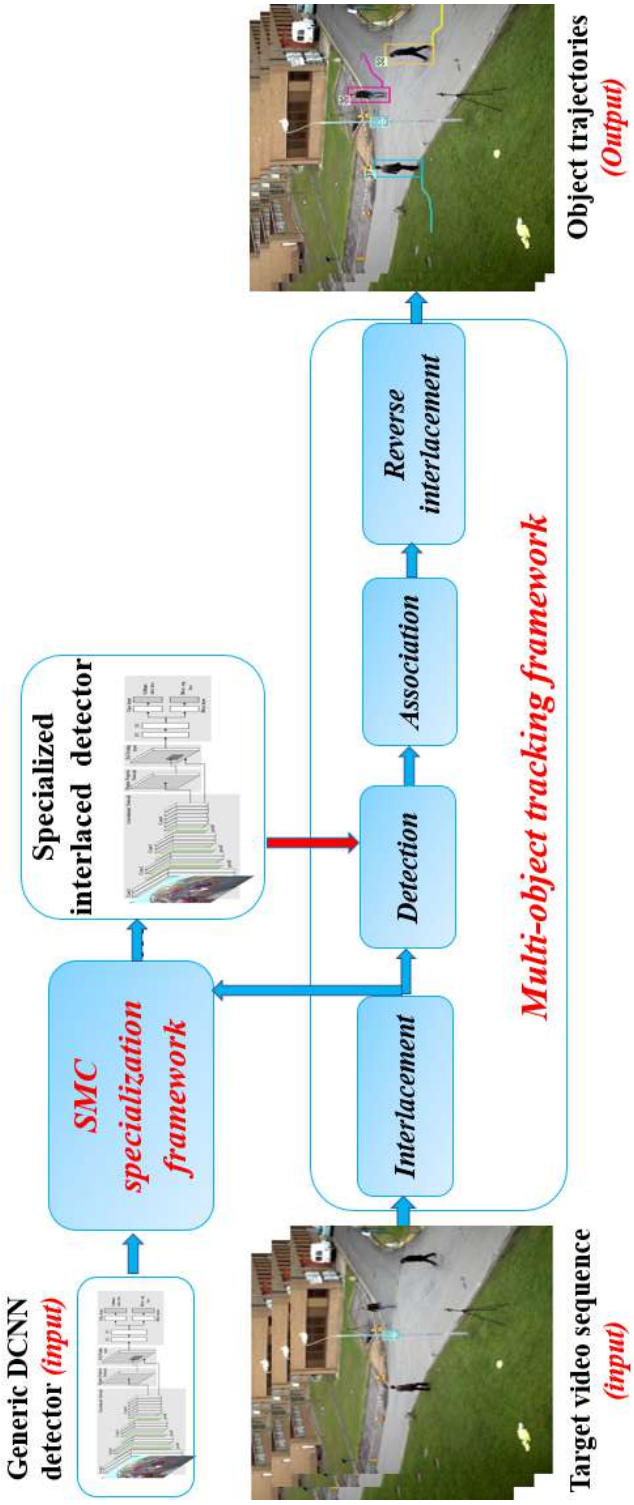


Figure 6.1: Synoptic diagram of suggested automatic specialization system

Another interesting perspective is to extend our work to other types of sensors like 3D cameras, Kinect, thermal imaging camera, Dynamic Vision Sensor (DVS) and lidar. Figure 6.2 shows examples of images captured with various types of sensors on which we will study the possibility of applying our specialization framework for detection and tracking tasks. Also, we will evaluate the specialization frameworks with other types of public datasets such as Kitti [Geiger 2012] and Pascal3D+ [Xiang 2014].

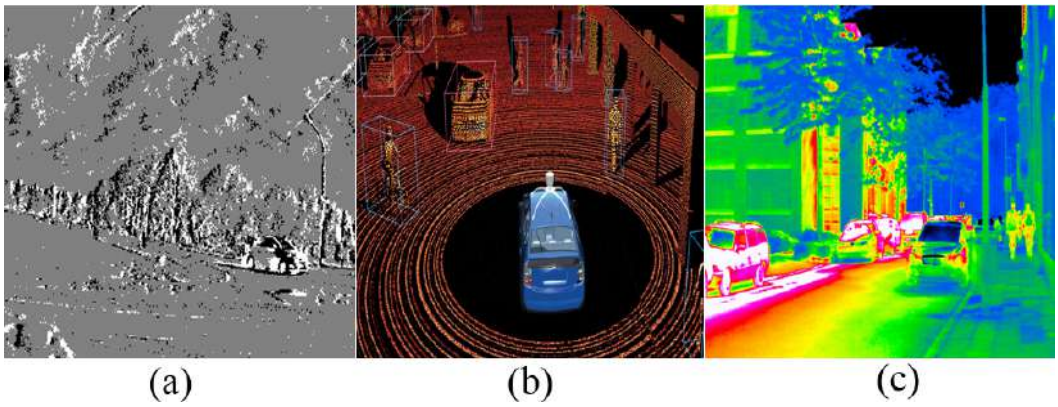


Figure 6.2: Examples of images captured with different types of sensors. (a) Image captured by DVS. (b) Image captured by lidar sensor. (c) Image captured by thermal imaging camera.

As a sixth perspective, we will propose to extend the transfer learning to specialize DCNN semantic segmentation algorithms to a specific scene in order to improve its segmentation performance in that scene. Differently from our previous work, the idea consists in suggesting a new specialization framework inspired from the Monte Carlo filter to automatically specialize a scene-specific segmentation algorithm. The proposed framework will use different strategies based on the SMC filter steps to approximate iteratively the target distribution as a set of binary masks that say whether or not a given pixel is part of an object, to specialize the segmentation algorithm towards a target scene. Moreover, we will put forward a new likelihood function based on the optical flow, which will be utilized to favor the selection of pixels associated to the right label. These pixels are selected according to their importance of weights, reflecting the likelihood that they belong to the target distribution. The weight of each pixel is related to a combination between a the optical flow information extracted from the target sequence and the confidence score of each pixel given by the output layer of the DCNN segmentation algorithm. Figure 6.3 illustrates the semantic segmentation task.

Another perspective of our work is the extension of our frameworks to online specialization, which seems to be an interesting idea since our specialization frameworks will be carried out online for detection and tracking tasks.

As a final applicative perspective, we will address the possibility of deploying

Figure 6.3: Illustration of semantic segmentation task. (a) depicts the input image and (b) is the output one generated by an instance segmentation algorithm.

the generated specialized deep detector or tracker in embedded devices with limited resources such as embedded devices based on CPUs, mobile phone and FPGAs with only several megabyte resources. The idea consists in proposing new acceleration and compression method to reduce the complexity of the DCNN specialized detector network. In our case, our application needs compacted models from pretrained models so the best solution is to use the pruning strategy which can be applied to explore the redundancy in the specialized-model parameters.

Bibliography

[Abadi 2015] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu and Xiaoqiang Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015. Software available from tensorflow.org. (Cited on page 16.)

[Abdulrahim 2016] Khairi Abdulrahim and Rosalina Abdul Salam. *Traffic Surveillance: A Review of Vision Based Vehicle Detection, Recognition and Tracking.* International Journal of Applied Engineering Research, vol. 11, no. 1, pages 713–726, 2016. (Cited on pages 82 and 85.)

[Abu-El-Haija 2016] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan and Sudheendra Vijayanarasimhan. *YouTube-8M: A large-scale video classification benchmark.* arXiv preprint arXiv:1609.08675, 2016. (Cited on page 15.)

[All 2011] Karim All, David Hasler and Frangois Fleuret. *FlowBoostâAppearance learning from sparsely annotated video.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1433–1440, 2011. (Cited on page 40.)

[Andriluka 2008] Mykhaylo Andriluka, Stefan Roth and Bernt Schiele. *People-tracking-by-detection and people-detection-by-tracking.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–8. IEEE, 2008. (Cited on pages 72 and 73.)

[Aytar 2011] Yusuf Aytar and Andrew Zisserman. *Tabula rasa: Model transfer for object category detection.* In 2011 International Conference on Computer Vision, pages 2252–2259. IEEE, 2011. (Cited on pages 33, 35, 36 and 40.)

[Aytar 2014] Yusuf Aytar. *Transfer learning for object category detection.* PhD thesis, University of Oxford, 2014. (Cited on page 29.)

[Badie 2014] Julien Badie and François Bremond. *Global tracker: an online evaluation framework to improve tracking quality.* In Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on, pages 25–30. IEEE, 2014. (Cited on page 65.)

[Bae 2014] Seung-Hwan Bae and Kuk-Jin Yoon. *Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning.* In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1218–1225, 2014. (Cited on pages 64 and 65.)

[Bay 2008] Herbert Bay, Andreas Ess, Tinne Tuytelaars and Luc Van Gool. *Speeded-up robust features (SURF).* Computer vision and image understanding, pages 346–359, 2008. (Cited on page 85.)

[Benfold 2011] Ben Benfold and Ian Reid. *Stable multi-target tracking in real-time surveillance video.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3457–3464, 2011. (Cited on page 40.)

[Bernardin 2008] Keni Bernardin and Rainer Stiefelhagen. *Evaluating multiple object tracking performance: the CLEAR MOT metrics.* EURASIP Journal on Image and Video Processing, vol. 2008, no. 1, page 246309, 2008. (Cited on page 73.)

[Carreira 2010] Joao Carreira and Cristian Sminchisescu. *Constrained parametric min-cuts for automatic object segmentation.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3241–3248. IEEE, 2010. (Cited on page 13.)

[Chari 2015] Visesh Chari, Simon Lacoste-Julien, Ivan Laptev and Josef Sivic. *On pairwise costs for network flow multi-object tracking.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5537–5545, 2015. (Cited on pages 66, 74 and 77.)

[Chatfield 2014] Ken Chatfield, Karen Simonyan, Andrea Vedaldi and Andrew Zisserman. *Return of the devil in the details: Delving deep into convolutional nets.* arXiv preprint arXiv:1405.3531, 2014. (Cited on page 100.)

[Cheng 2006] Hong Cheng, Nanning Zheng and Chong Sun. *Boosted Gabor features applied to vehicle detection.* In ICPR, pages 662–666. IEEE, 2006. (Cited on page 85.)

[Cheng 2016] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang and Nanning Zheng. *Person re-identification by multi-channel parts-based cnn with improved triplet loss function.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1335–1344, 2016. (Cited on page 66.)

[Chopra 2005] Sumit Chopra, Raia Hadsell and Yann LeCun. *Learning a similarity metric discriminatively, with application to face verification.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, pages 539–546. IEEE, 2005. (Cited on page 66.)

[Clevert 2015] Djork-Arné Clevert, Thomas Unterthiner and Sepp Hochreiter. *Fast and accurate deep network learning by exponential linear units (elus).* arXiv preprint arXiv:1511.07289, 2015. (Cited on page 22.)

[Collobert 2002] Ronan Collobert, Samy Bengio and Johnny Mariéthoz. *Torch: a modular machine learning software library.* Technical report, Idiap, 2002. (Cited on page 16.)

[Cordts 2016] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth and Bernt Schiele. *The Cityscapes Dataset for Semantic Urban Scene Understanding.* In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. (Cited on page 15.)

[Cortes 1995] Corinna Cortes and Vladimir Vapnik. *Support-vector networks.* Machine learning, vol. 20, no. 3, pages 273–297, 1995. (Cited on page 17.)

[Cox 1996] Ingemar J. Cox and Sunita L. Hingorani. *An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking.* IEEE Transactions on pattern analysis and machine intelligence, vol. 18, no. 2, pages 138–150, 1996. (Cited on page 66.)

[CS2] *CS231n: Convolutional Neural Networks for Visual Recognition.* `http://cs231n.github.io/`. Accessed: 2017-09-30. (Cited on page 23.)

[Dai 2007] Wenyuan Dai, Qiang Yang, Gui-Rong Xue and Yong Yu. *Boosting for transfer learning.* In International Conference on Machine learning (ICML), pages 193–200. ACM, 2007. (Cited on page 31.)

[Dai 2008] Wenyuan Dai, Qiang Yang, Gui-Rong Xue and Yong Yu. *Self-taught clustering.* In International Conference on Machine learning (ICML), pages 200–207. ACM, 2008. (Cited on page 30.)

[Dalal 2005] Navneet Dalal and Bill Triggs. *Histograms of oriented gradients for human detection.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 886–893, 2005. (Cited on pages 12, 13, 17, 18 and 85.)

[Danelljan 2014] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg and Joost Van de Weijer. *Adaptive color attributes for real-time visual tracking.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1090–1097, 2014. (Cited on page 66.)

[Danelljan 2017] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan and Michael Felsberg. *Discriminative scale space tracking.* IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 8, pages 1561–1575, 2017. (Cited on pages 64, 65 and 66.)

[Daume III 2006] Hal Daume III and Daniel Marcu. *Domain adaptation for statistical classifiers.* Journal of Artificial Intelligence Research (JAIR), vol. 26, pages 101–126, 2006. (Cited on page 30.)

[Dehghan 2015] Afshin Dehghan, Yicong Tian, Philip HS Torr and Mubarak Shah. *Target identity-aware network flow for online multiple target tracking.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1146–1154, 2015. (Cited on page 66.)

[Dehghan 2017] Afshin Dehghan and Mubarak Shah. *Binary Quadratic Programming for Online Tracking of Hundreds of People in Extremely Crowded Scenes.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017. (Cited on page 64.)

[Deng 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei. *Imagenet: A large-scale hierarchical image database.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 248–255, 2009. (Cited on pages 10, 15 and 16.)

[DiCarlo 2007] James J DiCarlo and David D Cox. *Untangling invariant object recognition.* Trends in cognitive sciences, vol. 11, no. 8, pages 333–341, 2007. (Cited on page 15.)

[Donahue 2013] Jeff Donahue, Judy Hoffman, Erik Rodner, Kate Saenko and Trevor Darrell. *Semi-supervised domain adaptation with instance constraints.* In Conference on Computer Vision and Pattern Recognition (CVPR), pages 668–675. IEEE Computer Society, 2013. (Cited on page 35.)

[Dorai 2016] Yosra Dorai, Sami Gazzah, Frederic Chausse and Najoua Essoukri Ben Amara. *Tracking multi-object using tracklet and Faster R-CNN: PhD Forum.* In Proceedings of the 10th International Conference on Distributed Smart Camera, pages 222–223. ACM, 2016. (Cited on page 90.)

[Dorai 2017] Yosra Dorai, Frédéric Chausse, Sami Gazzah and Najoua Essoukri Ben Amara. *Multi Target Tracking by Linking Tracklets with a Convolutional Neural Network.* In VISIGRAPP (6: VISAPP), pages 492–498, 2017. (Cited on pages 64, 74 and 77.)

[Doucet 2001] Arnaud Doucet, Nando De Freitas and Neil Gordon. Sequential monte carlo methods in practice. Springer, 2001. (Cited on pages 43 and 90.)

[Douze 2011] Matthijs Douze, Arnau Ramisa and Cordelia Schmid. *Combining attributes and fisher vectors for efficient image retrieval.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 745–752, 2011. (Cited on page 35.)

[Endres 2010] Ian Endres and Derek Hoiem. *Category independent object proposals.* Computer Vision–ECCV 2010, pages 575–588, 2010. (Cited on page 13.)

[Ess 2008] A. Ess, B. Leibe, K. Schindler, and L. van Gool. *A Mobile Vision System for Robust Multi-Person Tracking*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08). IEEE Press, June 2008. (Cited on page 73.)

[Everingham 2010] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn and Andrew Zisserman. *The pascal visual object classes (voc) challenge*. International journal of computer vision, vol. 88, no. 2, pages 303–338, 2010. (Cited on pages vii, 10, 27, 53, 94 and 95.)

[Fei-Fei 2006] Li Fei-Fei, Rob Fergus and Pietro Perona. *One-shot learning of object categories*. Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 28, no. 4, pages 594–611, 2006. (Cited on page 32.)

[Felzenszwalb 2004] Pedro F Felzenszwalb and Daniel P Huttenlocher. *Efficient graph-based image segmentation*. International journal of computer vision, vol. 59, no. 2, pages 167–181, 2004. (Cited on page 13.)

[Felzenszwalb 2010] Pedro Felzenszwalb, Ross Girshick, David McAllester and Deva Ramanan. *Object detection with discriminatively trained part-based models*. IEEE transactions on pattern analysis and machine intelligence, vol. 32, no. 9, pages 1627–1645, 2010. (Cited on pages 12 and 13.)

[Ferrari 2007] Vittorio Ferrari, Frederic Jurie and Cordelia Schmid. *Accurate object detection with deformable shape models learnt from images*. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–8, 2007. (Cited on page 12.)

[Ferrari 2010] Vittorio Ferrari, Frederic Jurie and Cordelia Schmid. *From images to shape models for object detection*. International journal of computer vision, vol. 87, no. 3, pages 284–303, 2010. (Cited on page 12.)

[Ferryman 2009] J Ferryman and A Shahrokni. *Pets2009: Dataset and challenge*. In Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on, pages 1–6. IEEE, 2009. (Cited on pages 72 and 73.)

[Fink 2005] Michael Fink. *Object classification from a single example utilizing class relevance metrics*. Advances in Neural Information Processing Systems (NIPS), vol. 17, pages 449–456, 2005. (Cited on page 33.)

[Freund 1995] Yoav Freund and Robert E Schapire. *A desicion-theoretic generalization of on-line learning and an application to boosting*. In European conference on computational learning theory, pages 23–37. Springer, 1995. (Cited on page 12.)

[Gao 2008] Jing Gao, Wei Fan, Jing Jiang and Jiawei Han. *Knowledge transfer via multiple model local structure mapping*. In ACM International Conference

on Knowledge Discovery and Data Mining (ACM SIGKDD), pages 283–291. ACM, 2008. (Cited on page 33.)

[Gao 2010] Wenshuo Gao, Xiaoguang Zhang, Lei Yang and Huizhong Liu. *An improved Sobel edge detection.* In Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on, pages 67–71. IEEE, 2010. (Cited on page 85.)

[Gao 2012] Tianshi Gao, Michael Stark and Daphne Koller. *What makes a good detector?–structured priors for learning from few examples.* In European Conference on Computer Vision (ECCV), pages 354–367. Springer, 2012. (Cited on pages 33 and 35.)

[Garcia 2002] Christophe Garcia and Manolis Delakis. *A neural architecture for fast and robust face detection.* In Pattern Recognition, 2002. Proceedings. 16th International Conference on, volume 2, pages 44–47. IEEE, 2002. (Cited on page 13.)

[Geiger 2012] Andreas Geiger, Philip Lenz and Raquel Urtasun. *Are we ready for autonomous driving? the kitti vision benchmark suite.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3354–3361, 2012. (Cited on pages 10 and 107.)

[Girshick 2014a] Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik. *Rich feature hierarchies for accurate object detection and semantic segmentation.* In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014. (Cited on page 14.)

[Girshick 2014b] Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik. *Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014. (Cited on pages 27, 52, 60, 62 and 88.)

[Girshick 2015a] Ross Girshick. *Fast r-cnn.* In Proceedings of the IEEE International Conference on Computer Vision, pages 1440–1448, 2015. (Cited on pages 14, 23, 27, 52, 60 and 62.)

[Girshick 2015b] Ross Girshick. *Fast r-cnn.* In Proceedings of the IEEE International Conference on Computer Vision, pages 1440–1448, 2015. (Cited on pages 71 and 94.)

[Glorot 2011] Xavier Glorot, Antoine Bordes and Yoshua Bengio. *Domain adaptation for large-scale sentiment classification: A deep learning approach.* In Proceedings of the 28th International Conference on Machine Learning (ICML-11), pages 513–520, 2011. (Cited on page 41.)

[Goodfellow 2012] Ian J Goodfellow, Aaron Courville and Yoshua Bengio. *Spike-and-slab sparse coding for unsupervised feature discovery.* arXiv, 2012. (Cited on page 41.)

[Guerry 2017] Joris Guerry. *Reconnaissance visuelle robuste par reseaux de neurones dans des scenarios d'exploration robotique.* PhD thesis, Universite PARIS-SACLAY, 2017. (Cited on page 22.)

[Guyon 2011] Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor and David W Aha. *Unsupervised and transfer learning challenge.* In IJCNN, pages 793–800. IEEE, 2011. (Cited on page 86.)

[Han 2004] Mei Han, Wei Xu, Hai Tao and Yihong Gong. *An algorithm for multiple object trajectory tracking.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, pages I–I. IEEE, 2004. (Cited on page 66.)

[Han 2006] Feng Han, Ying Shan, Ryan Cekander, Harpreet S Sawhney and Rakesh Kumar. *A two-stage approach to people and vehicle detection with hog-based svm.* In PMIS, pages 133–140. Citeseer, 2006. (Cited on page 85.)

[He 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification.* In Proceedings of the IEEE international conference on computer vision, pages 1026–1034, 2015. (Cited on page 22.)

[He 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. *Deep residual learning for image recognition.* In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. (Cited on pages 24, 25 and 26.)

[He 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár and Ross Girshick. *Mask r-cnn.* arXiv preprint arXiv:1703.06870, 2017. (Cited on page 14.)

[Hoffer 2015] Elad Hoffer and Nir Ailon. *Deep metric learning using triplet network.* In International Workshop on Similarity-Based Pattern Recognition, pages 84–92. Springer, 2015. (Cited on page 66.)

[Howard 2017] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto and Hartwig Adam. *Mobilenets: Efficient convolutional neural networks for mobile vision applications.* arXiv preprint arXiv:1704.04861, 2017. (Cited on pages 27 and 100.)

[Htike 2014] Kyaw Kyaw Htike and David C Hogg. *Efficient non-iterative domain adaptation of pedestrian detectors to video scenes.* In 2014 22nd International Conference on Pattern Recognition (ICPR), pages 654–659. IEEE, 2014. (Cited on pages 40, 55, 56, 84, 95 and 96.)

[Hu 2015] Junlin Hu, Jiwen Lu and Yap-Peng Tan. *Deep Transfer Metric Learning.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 325–333, 2015. (Cited on page 86.)

[Huang 2005] Chang Huang, Haizhou Ai, Yuan Li and Shihong Lao. *Vector boosting for rotation invariant multi-view face detection.* In ICCV, pages 446–453. IEEE, 2005. (Cited on page 85.)

[Huang 2006] Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M Borgwardt and Bernhard Schölkopf. *Correcting sample selection bias by unlabeled data.* In Advances in neural information processing systems (NIPS), pages 601–608, 2006. (Cited on page 31.)

[Huang 2012] Gary B Huang, Honglak Lee and Erik Learned-Miller. *Learning hierarchical representations for face verification with convolutional deep belief networks.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2518–2525, 2012. (Cited on pages 10 and 86.)

[Huang 2016a] Chen Huang, Chen Change Loy and Xiaoou Tang. *Local similarity-aware deep feature embedding.* In Advances in Neural Information Processing Systems, pages 1262–1270, 2016. (Cited on page 66.)

[Huang 2016b] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama*et al. Speed/accuracy trade-offs for modern convolutional object detectors.* arXiv preprint arXiv:1611.10012, 2016. (Cited on page 15.)

[Ioffe 2015] Sergey Ioffe and Christian Szegedy. *Batch normalization: Accelerating deep network training by reducing internal covariate shift.* In International Conference on Machine Learning, pages 448–456, 2015. (Cited on page 22.)

[Jain 2011] Mihir Jain, Hervé Jégou and Patrick Gros. *Asymmetric hamming embedding: taking the best of our bits for large scale image search.* In the 19th ACM International Conference on Multimedia (ICM), pages 1441–1444. ACM, 2011. (Cited on page 30.)

[Jia 2014] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama and Trevor Darrell. *Caffe: Convolutional architecture for fast feature embedding.* In ACM, pages 675–678. ACM, 2014. (Cited on pages 16, 95 and 100.)

[Jiang 2007] Jing Jiang and ChengXiang Zhai. *Instance weighting for domain adaptation in NLP.* In 45th Annual Meeting of the Association for Computational Linguistics (ACL), volume 7, pages 264–271. The Association for Computational Linguistics, 2007. (Cited on page 31.)

[Jin 2007] Xiaoying Jin and Curt H Davis. *Vehicle detection from high-resolution satellite imagery using morphological shared-weight neural networks.* IVC, pages 1422–1431, 2007. (Cited on page 85.)

[Kalogerakis 2017] Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji and Siddhartha Chaudhuri. *3D Shape Segmentation With Projective Convolutional Networks.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017. (Cited on page 65.)

[Kataoka 2015] Hirokatsu Kataoka, Kenji Iwata and Yutaka Satoh. *Feature Evaluation of Deep Convolutional Neural Networks for Object Recognition and Detection.* arXiv preprint arXiv:1509.07627, 2015. (Cited on pages 20 and 25.)

[Kim 2015] Chanho Kim, Fuxin Li, Arridhana Ciptadi and James M Rehg. *Multiple hypothesis tracking revisited.* In Proceedings of the IEEE International Conference on Computer Vision, pages 4696–4704, 2015. (Cited on pages 64, 65, 66, 73, 74, 76 and 77.)

[Kong 2016] Tao Kong, Anbang Yao, Yurong Chen and Fuchun Sun. *Hypernet: Towards accurate region proposal generation and joint object detection.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 845–853, 2016. (Cited on page 14.)

[Krizhevsky 2012] Alex Krizhevsky, Ilya Sutskever and Geoffrey E Hinton. *Imagenet classification with deep convolutional neural networks.* In ANIPS, pages 1097–1105, 2012. (Cited on pages 16 and 24.)

[Kumar 2016] BG Kumar, Gustavo Carneiro, Ian Reid*et al. Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5385–5394, 2016. (Cited on page 66.)

[Kuzborskij 2013] Ilja Kuzborskij, Francesco Orabona and Barbara Caputo. *From n to n+ 1: Multiclass transfer incremental learning.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3358–3365. IEEE Computer Society, 2013. (Cited on page 35.)

[Leal-Taixé 2016] Laura Leal-Taixé, Cristian Canton-Ferrer and Konrad Schindler. *Learning by tracking: Siamese CNN for robust target association.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 33–40, 2016. (Cited on pages 65 and 66.)

[LeCun 1989] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard and Lawrence D Jackel. *Backpropagation applied to handwritten zip code recognition.* Neural computation, vol. 1, no. 4, pages 541–551, 1989. (Cited on pages 20 and 52.)

[LeCun 1998] Yann LeCun, Léon Bottou, Yoshua Bengio and Patrick Haffner. *Gradient-based learning applied to document recognition.* Proceedings of the IEEE, pages 2278–2324, 1998. (Cited on page 21.)

[Lee 2004] BJ Lee, JB Park, YH Joo and SH Jin. *Intelligent Kalman filter for tracking a manoeuvring target.* IEE Proceedings-Radar, Sonar and Navigation, vol. 151, no. 6, pages 344–350, 2004. (Cited on pages 65 and 93.)

[Levin 2003] Anat Levin, Paul A Viola and Yoav Freund. *Unsupervised Improvement of Visual Detectors using Co-Training.* In ICCV, volume 1, pages 626–633, 2003. (Cited on page 40.)

[Li 2015a] Jianan Li, Xiaodan Liang, ShengMei Shen, Tingfa Xu, Jiashi Feng and Shuicheng Yan. *Scale-aware fast R-CNN for pedestrian detection.* arXiv preprint arXiv:1510.08160, 2015. (Cited on page 85.)

[Li 2015b] Xudong Li, Mao Ye, Min Fu, Pei Xu and Tao Li. *Domain adaption of vehicle detector based on convolutional neural networks.* International Journal of Control, Automation and Systems, vol. 13, no. 4, pages 1020–1031, 2015. (Cited on pages 40, 41, 53, 54, 55, 56 and 84.)

[Lim 2011] Joseph J Lim, Ruslan Salakhutdinov and Antonio Torralba. *Transfer learning by borrowing examples for multiclass object detection.* In Advances in neural information processing systems, pages 118–126, 2011. (Cited on pages 32 and 35.)

[Lin 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár and C Lawrence Zitnick. *Microsoft coco: Common objects in context.* In European Conference on Computer Vision, pages 740–755. Springer, 2014. (Cited on pages 10 and 15.)

[Liu 2016] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu and Alexander C Berg. *Ssd: Single shot multibox detector.* In European conference on computer vision, pages 21–37. Springer, 2016. (Cited on pages 14, 15, 27, 64 and 100.)

[Long 2015] Jonathan Long, Evan Shelhamer and Trevor Darrell. *Fully convolutional networks for semantic segmentation.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3431–3440, 2015. (Cited on page 10.)

[Lowe 1999] David G Lowe. *Object recognition from local scale-invariant features.* In International Conference on Computer vision (ICCV), volume 2, pages 1150–1157. IEEE, 1999. (Cited on pages 17 and 18.)

[Maâmatou 2016a] Houda Maâmatou, Thierry Chateau, Sami Gazzah, Yann Goyat and Najoua Essoukri Ben Amara. *Sequential Monte Carlo filter based on multiple strategies for a scene specialization classifier.* EURASIP Journal

on Image and Video Processing, vol. 2016, no. 1, page 40, 2016. (Cited on pages 29 and 87.)

[Maâmatou 2016b] Houda Maâmatou, Thierry Chateau, Sami Gazzah, Yann Goyat and Najoua Essoukri Ben Amara. *Sequential Monte Carlo Filter Based on Multiple Strategies for a Scene Specialization Classifier*. EURASIP Journal on Image and Video Processing (EURASIP - JIVP), vol. 2016, no. 1, page 40, 2016. (Cited on page 37.)

[Maâmatou 2016c] Houda Maâmatou, Thierry Chateau, Sami Gazzah, Yann Goyat and Najoua Essoukri Ben Amara. *Transductive Transfer Learning to Specialize a Generic Classifier Towards a Specific Scene*. In VISAPP, pages 411–422, 2016. (Cited on pages vii, 40, 55, 56, 60, 61, 62, 84, 85, 95, 96 and 97.)

[Maâmatou 2016d] Houda Maâmatou, Thierry Chateau, Sami Gazzah, Yann Goyat and Najoua Essoukri Ben Amara. *Transductive Transfer Learning to Specialize a Generic Classifier Towards a Specific Scene*. In International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 4: Computer VISion Theory and APPlications (VISAPP), volume 4, pages 411–422. SciTePress, 2016. (Cited on page 30.)

[Maas 2013] Andrew L Maas, Awni Y Hannun and Andrew Y Ng. *Rectifier nonlinearities improve neural network acoustic models*. In Proc. ICML, volume 30, 2013. (Cited on page 22.)

[Malisiewicz 2011] Tomasz Malisiewicz, Abhinav Gupta and Alexei A Efros. *Ensemble of exemplar-svms for object detection and beyond*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 89–96. IEEE, 2011. (Cited on page 17.)

[Mao 2015] Yunxiang Mao and Zhaozheng Yin. *Training a scene-specific pedestrian detector using tracklets*. In Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on, pages 170–176. IEEE, 2015. (Cited on pages 37, 40, 55, 56, 84, 95 and 96.)

[Matti 2017] Damien Matti, Hazım Kemal Ekenel and Jean-Philippe Thiran. *Combining LiDAR space clustering and convolutional neural networks for pedestrian detection*. In Advanced Video and Signal Based Surveillance (AVSS), pages 1–6. IEEE, 2017. (Cited on page 82.)

[McCulloch 1943] Warren S McCulloch and Walter Pitts. *A logical calculus of the ideas immanent in nervous activity*. The bulletin of mathematical biophysics, vol. 5, no. 4, pages 115–133, 1943. (Cited on pages 15 and 18.)

[McLaughlin 2015] Niall McLaughlin, Jesus Martinez Del Rincon and Paul Miller. *Enhancing linear programming with motion modeling for multi-target tracking.* In Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on, pages 71–77. IEEE, 2015. (Cited on page 66.)

[Mei 2011] Xue Mei and Haibin Ling. *Robust visual tracking and vehicle classification via sparse representation.* IEEE transactions on pattern analysis and machine intelligence, vol. 33, no. 11, pages 2259–2272, 2011. (Cited on page 46.)

[Mesnil 2012] Grégoire Mesnil, Yann Dauphin, Xavier Glorot, Salah Rifai, Yoshua Bengio, Ian J Goodfellow, Erick Lavoie, Xavier Muller, Guillaume Desjardins, David Warde-Farley*et al. Unsupervised and Transfer Learning Challenge: a Deep Learning Approach.* ICML Unsupervised and Transfer Learning, pages 97–110, 2012. (Cited on page 41.)

[Mhalla 2016a] Ala Mhalla, Thierry Chateau, Sami Gazzah and Najoua Essoukri Ben Amara. *Scene-Specific Pedestrian Detector Using Monte Carlo Framework and Faster R-CNN Deep Model: PhD Forum.* In Proceedings of the 10th International Conference on Distributed Smart Camera, pages 228–229. ACM, 2016. (Cited on page 84.)

[Mhalla 2016b] Ala Mhalla, Houda Maâmatou, Thierry Chateau, Sami Gazzah and Najoua Essoukri Ben Amara. *Faster R-CNN Scene Specialization with a Sequential Monte-Carlo Framework.* In International Conference on Digital Image Computing: Techniques and Applications (DICTA), pages 1–7. IEEE, 2016. (Cited on pages 11, 40, 84, 85, 86, 87, 94, 95, 96, 97 and 99.)

[Mhalla 2017] Ala Mhalla, Thierry Chateau, Houda Maamatou, Sami Gazzah and Najoua Essoukri Ben Amara. *SMC faster R-CNN: Toward a scene-specialized multi-object detector.* Computer Vision and Image Understanding, vol. 164, pages 3–15, 2017. (Cited on pages 64 and 87.)

[Milan 2013] Anton Milan, Konrad Schindler and Stefan Roth. *Detection-and trajectory-level exclusion in multiple object tracking.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3682–3689, 2013. (Cited on pages 74, 76 and 77.)

[Milan 2014] Anton Milan, Stefan Roth and Konrad Schindler. *Continuous energy minimization for multitarget tracking.* IEEE transactions on pattern analysis and machine intelligence, vol. 36, no. 1, pages 58–72, 2014. (Cited on page 64.)

[Milan 2017] Anton Milan, Seyed Hamid Rezatofighi, Anthony R Dick, Ian D Reid and Konrad Schindler. *Online Multi-Target Tracking Using Recurrent Neural Networks.* In AAAI, pages 4225–4232, 2017. (Cited on page 65.)

[Naiel 2014] Mohamed A Naiel, M Omair Ahmad, MNS Swamy, Yi Wu and Ming-Hsuan Yang. *Online multi-person tracking via robust collaborative model.* In Image Processing (ICIP), 2014 IEEE International Conference on, pages 431–435. IEEE, 2014. (Cited on page 64.)

[Nair 2004] Vinod Nair and James J Clark. *An unsupervised, online learning framework for moving object detection.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, pages II–II, 2004. (Cited on pages 55 and 56.)

[Nam 2016] Hyeonseob Nam and Bohyung Han. *Learning multi-domain convolutional neural networks for visual tracking.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4293–4302, 2016. (Cited on pages 10 and 65.)

[Neelima 2012] D Neelima and Gowtham Mamidisetti. *A Computer Vision Model for Vehicle Detection in Traffic Surveillance.* International Journal of Engineering Science & Advanced Technology (IJESAT), vol. 2, no. 5, pages 1203–1209, 2012. (Cited on page 85.)

[Ng 2015] Hong-Wei Ng, Viet Dung Nguyen, Vassilios Vonikakis and Stefan Winkler. *Deep learning for emotion recognition on small datasets using transfer learning.* In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, pages 443–449. ACM, 2015. (Cited on page 41.)

[Noh 2015] Hyeonwoo Noh, Seunghoon Hong and Bohyung Han. *Learning deconvolution network for semantic segmentation.* In Proceedings of the IEEE International Conference on Computer Vision, pages 1520–1528, 2015. (Cited on page 10.)

[Ohn-Bar 2015] Eshed Ohn-Bar, Ashish Tawari, Sujitha Martin and Mohan M Trivedi. *On surveillance for safety critical events: In-vehicle video networks for predictive driver assistance systems.* Computer Vision and Image Understanding, vol. 134, pages 130–140, 2015. (Cited on page 82.)

[Ojala 1996] Timo Ojala, Matti Pietikäinen and David Harwood. *A comparative study of texture measures with classification based on featured distributions.* Pattern recognition, vol. 29, no. 1, pages 51–59, 1996. (Cited on page 17.)

[Oquab 2014] Maxime Oquab, Leon Bottou, Ivan Laptev and Josef Sivic. *Learning and transferring mid-level image representations using convolutional neural networks.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1717–1724, 2014. (Cited on page 41.)

[Pan 2008] Sinno Jialin Pan, James T Kwok and Qiang Yang. *Transfer Learning via Dimensionality Reduction.* In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (CAI), volume 8, pages 677–682. AAAI Press, 2008. (Cited on page 34.)

[Pan 2010a] Sinno Jialin Pan and Qiang Yang. *A survey on transfer learning.* KDE, pages 1345–1359, 2010. (Cited on pages 28, 29 and 30.)

[Pan 2010b] Xinting Pan, Yunlong Guo and Aidong Men. *Traffic surveillance system for vehicle flow detection.* In Computer Modeling and Simulation, 2010. ICCMS'10. Second International Conference on, volume 1, pages 314–318. IEEE, 2010. (Cited on page 27.)

[Pan 2011] Sinno Jialin Pan, Ivor W Tsang, James T Kwok and Qiang Yang. *Domain adaptation via transfer component analysis.* IEEE Transactions on Neural Networks, vol. 22, no. 2, pages 199–210, 2011. (Cited on pages 34 and 35.)

[Quanz 2012] Brian Quanz, Jun Huan and Meenakshi Mishra. *Knowledge transfer with low-quality data: A feature extraction issue.* IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 10, pages 1789–1802, 2012. (Cited on page 40.)

[Quattoni 2008] Ariadna Quattoni, Michael Collins and Trevor Darrell. *Transfer learning for image classification with sparse prototype representations.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–8, 2008. (Cited on page 34.)

[Raina 2007] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer and Andrew Y Ng. *Self-taught learning: transfer learning from unlabeled data.* In International conference on Machine learning (ICML), pages 759–766. ACM, 2007. (Cited on pages 30 and 33.)

[Redmon 2016a] Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi. *You only look once: Unified, real-time object detection.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 779–788, 2016. (Cited on pages 14 and 27.)

[Redmon 2016b] Joseph Redmon and Ali Farhadi. *YOLO9000: better, faster, stronger.* arXiv preprint arXiv:1612.08242, 2016. (Cited on pages 27 and 82.)

[Ren 2015a] Shaoqing Ren, Kaiming He, Ross Girshick and Jian Sun. *Faster R-CNN: Towards real-time object detection with region proposal networks.* In Advances in neural information processing systems, pages 91–99, 2015. (Cited on pages 71 and 73.)

[Ren 2015b] Shaoqing Ren, Kaiming He, Ross Girshick and Jian Sun. *Faster R-CNN: Towards real-time object detection with region proposal networks.* In Advances in neural information processing systems, pages 91–99, 2015. (Cited on pages 86, 87, 88, 94, 95, 96, 97 and 99.)

[Ren 2015c] Shaoqing Ren, Kaiming He, Ross Girshick and Jian Sun. *Faster R-CNN: Towards real-time object detection with region proposal networks.* In

Advances in Neural Information Processing Systems (NIPS), pages 91–99, 2015. (Cited on pages 10, 14, 15, 27, 41, 50, 52, 56 and 62.)

[Rosenberg 2005] Chuck Rosenberg, Martial Hebert and Henry Schneiderman. *Semi-supervised self-training of object detection models.* In Seventh Workshop on Applications of Computer Vision (WACV). IEEE Press, 2005. (Cited on page 40.)

[Rota Bulo 2017] Samuel Rota Bulo, Gerhard Neuhold and Peter Kontschieder. *Loss Max-Pooling for Semantic Image Segmentation.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017. (Cited on page 65.)

[Saenko 2010] Kate Saenko, Brian Kulis, Mario Fritz and Trevor Darrell. *Adapting visual category models to new domains.* In European conference on computer vision (ECCV), pages 213–226. Springer, 2010. (Cited on page 34.)

[Sermanet 2013] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus and Yann LeCun. *Overfeat: Integrated recognition, localization and detection using convolutional networks.* arXiv preprint arXiv:1312.6229, 2013. (Cited on pages 12, 13 and 26.)

[Shen 2017] Zhiqiang Shen, Zhuang Liu, Jianguo Li, Yu-Gang Jiang, Yurong Chen and Xiangyang Xue. *DSOD: Learning Deeply Supervised Object Detectors From Scratch.* In The IEEE International Conference on Computer Vision (ICCV), Oct 2017. (Cited on pages 27 and 64.)

[Shrivastava 2016] Abhinav Shrivastava, Abhinav Gupta and Ross Girshick. *Training region-based object detectors with online hard example mining.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 761–769, 2016. (Cited on page 27.)

[Shu 2012] Guang Shu, Afshin Dehghan, Omar Oreifej, Emily Hand and Mubarak Shah. *Part-based multiple-person tracking with partial occlusion handling.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1815–1821, 2012. (Cited on page 64.)

[Simonyan 2014] Karen Simonyan and Andrew Zisserman. *Very deep convolutional networks for large-scale image recognition.* arXiv preprint arXiv:1409.1556, 2014. (Cited on pages 3, 25, 52, 73, 88, 94 and 100.)

[Sivaraman 2013] Sayanan Sivaraman and Mohan Manubhai Trivedi. *Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis.* IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 4, pages 1773–1795, 2013. (Cited on page 85.)

[Smal 2007] Ihor Smal, Wiro Niessen and Erik Meijering. *Advanced particle filtering for multiple object tracking in dynamic fluorescence microscopy images.* In BIFNM, pages 1048–1051. IEEE, 2007. (Cited on page 46.)

[Smith 2013] Adrian Smith, Arnaud Doucet, Nando de Freitas and Neil Gordon. Sequential monte carlo methods in practice. Springer Science & Business Media, 2013. (Cited on pages 41 and 46.)

[Song 2011] Zheng Song, Qiang Chen, Zhongyang Huang, Yang Hua and Shuicheng Yan. *Contextualizing object detection and classification.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1585–1592, 2011. (Cited on page 35.)

[Srivastava 2014] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov. *Dropout: a simple way to prevent neural networks from overfitting.* Journal of machine learning research, vol. 15, no. 1, pages 1929–1958, 2014. (Cited on page 22.)

[Stark 2009] Michael Stark, Michael Goesele and Bernt Schiele. *A shape-based object class model for knowledge transfer.* In International Conference on Computer Vision (ICCV), pages 373–380. IEEE, 2009. (Cited on page 33.)

[Sugiyama 2008] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul V Buenau and Motoaki Kawanabe. *Direct importance estimation with model selection and its application to covariate shift adaptation.* In Advances in neural information processing systems (NIPS), pages 1433–1440, 2008. (Cited on page 31.)

[Suleiman 2017] Amr Suleiman, Zhengdong Zhang and Vivienne Sze. *A 58.6 mW 30 Frames/s Real-Time Programmable Multiobject Detection Accelerator With Deformable Parts Models on Full HD* 1920*times 1080Videos. *IEEEJournalofSolid − StateCircuits, vol.* 52, *no.* 3, *pages*844 − −855, 2017. (*Citedonpage* 12.)

[Sun 2014] Yi Sun, Yuheng Chen, Xiaogang Wang and Xiaoou Tang. *Deep learning face representation by joint identification-verification.* In Advances in neural information processing systems, pages 1988–1996, 2014. (Cited on page 66.)

[Szegedy 2013] Christian Szegedy, Alexander Toshev and Dumitru Erhan. *Deep neural networks for object detection.* In Advances in neural information processing systems, pages 2553–2561, 2013. (Cited on pages 12 and 13.)

[Szegedy 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich. *Going deeper with convolutions.* In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1–9, 2015. (Cited on page 65.)

[Taigman 2014] Yaniv Taigman, Ming Yang, Marc Aurelio Ranzato and Lior Wolf. *Deepface: Closing the gap to human-level performance in face verification.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1701–1708, 2014. (Cited on pages 10, 66 and 86.)

[Tang 2012] Kevin Tang, Vignesh Ramanathan, Li Fei-Fei and Daphne Koller. *Shifting weights: Adapting object detectors from image to video.* In Advances in Neural Information Processing Systems (NIPS), pages 638–646, 2012. (Cited on page 36.)

[Tang 2017] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres and Bernt Schiele. *Multiple People Tracking by Lifted Multicut and Person Re-identification.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3539–3548, 2017. (Cited on pages 64 and 65.)

[Tao 2016] Ran Tao, Efstratios Gavves and Arnold WM Smeulders. *Siamese instance search for tracking.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1420–1429, 2016. (Cited on page 66.)

[Tian 2015] Yonglong Tian, Ping Luo, Xiaogang Wang and Xiaoou Tang. *Pedestrian detection aided by deep learning semantic tasks.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5079–5087, 2015. (Cited on page 85.)

[Tommasi 2013] Tatiana Tommasi. *Learning to learn by exploiting prior knowledge.* PhD thesis, École Polytechnique Fédérale de Lausanne, 2013. (Cited on pages 29 and 35.)

[Torralba 2004] Antonio Torralba, Kevin Murphy and William Freeman. *Sharing features: efficient boosting procedures for multiclass object detection.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, pages II–II, 2004. (Cited on page 13.)

[Torralba 2007] Antonio Torralba, Kevin P Murphy and William T Freeman. *Sharing visual features for multiclass and multiview object detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 5, pages 854–869, 2007. (Cited on page 13.)

[Uijlings 2013] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers and Arnold WM Smeulders. *Selective search for object recognition.* International journal of computer vision (IJCV), vol. 104, no. 2, pages 154–171, 2013. (Cited on pages 13, 14 and 26.)

[Valmadre 2017] Jack Valmadre, Luca Bertinetto, Joao Henriques, Andrea Vedaldi and Philip H. S. Torr. *End-To-End Representation Learning for Correlation Filter Based Tracking.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017. (Cited on page 65.)

[Vermaak 2005] Jaco Vermaak, Simon J Godsill and Patrick Perez. *Monte Carlo filtering for multi target tracking and data association.* IEEE Transactions on Aerospace and Electronic systems, vol. 41, no. 1, pages 309–332, 2005. (Cited on page 65.)

[Viola 2001a] Paul Viola and Michael Jones. *Rapid object detection using a boosted cascade of simple features.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages I–I, 2001. (Cited on page 12.)

[Viola 2001b] Paul Viola and Michael Jones. *Robust real-time object detection.* International Journal of Computer Vision (IJCV), vol. 4, pages 51–52, 2001. (Cited on page 17.)

[Wang 2008] Zheng Wang, Yangqiu Song and Changshui Zhang. Transferred dimensionality reduction, pages 550–565. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. (Cited on page 34.)

[Wang 2009] Xiaogang Wang, Xiaoxu Ma and W Eric L Grimson. *Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models.* PAMI, pages 539–555, 2009. (Cited on page 53.)

[Wang 2010] Gang Wang, David Forsyth and Derek Hoiem. *Comparative object similarity for improved recognition with few or no examples.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3525–3532, 2010. (Cited on page 32.)

[Wang 2011] Meng Wang and Xiaogang Wang. *Automatic adaptation of a generic pedestrian detector to a specific traffic scene.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3401–3408, 2011. (Cited on pages 30 and 36.)

[Wang 2012a] Meng Wang, Wei Li and Xiaogang Wang. *Transferring a generic pedestrian detector towards specific scenes.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3274–3281, 2012. (Cited on pages 40, 53, 60 and 94.)

[Wang 2012b] Xiaoyu Wang, Gang Hua and Tony X Han. *Detection by detections: Non-parametric detector adaptation for a video.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 350–357. IEEE, 2012. (Cited on page 36.)

[Wang 2014a] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen and Ying Wu. *Learning fine-grained image similarity with deep ranking.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1386–1393, 2014. (Cited on page 66.)

[Wang 2014b] Xiaogang Wang, Meng Wang and Wei Li. *Scene-specific pedestrian detection for static video surveillance.* PAMI, pages 361–362, 2014. (Cited on pages 40, 41, 54, 55, 56, 60, 95 and 96.)

[Wang 2014c] Xinchao Wang, Engin Türetken, François Fleuret and Pascal Fua. *Tracking interacting objects optimally using integer programming.* In European Conference on Computer Vision, pages 17–32. Springer, 2014. (Cited on pages 64 and 65.)

[Wang 2016a] Bing Wang, Li Wang, Bing Shuai, Zhen Zuo, Ting Liu, Kap Luk Chan and Gang Wang. *Joint learning of convolutional neural networks and temporally constrained metrics for tracklet association.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 1–8, 2016. (Cited on pages 64, 65 and 66.)

[Wang 2016b] Lijun Wang, Wanli Ouyang, Xiaogang Wang and Huchuan Lu. *Stct: Sequentially training convolutional networks for visual tracking.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1373–1381, 2016. (Cited on pages 10, 65 and 66.)

[Wang 2017] Mengmeng Wang, Yong Liu and Zeyi Huang. *Large Margin Object Tracking With Circulant Feature Maps.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017. (Cited on page 65.)

[Wei Zheng 2013] Luhong Liang Wei Zheng Hong Chang. *Strip Features for Fast Object Detection.* 2013. (Cited on page 85.)

[Wu 2007] Bo Wu and Ram Nevatia. *Cluster boosted tree classifier for multi-view, multi-pose object detection.* In ICCV, pages 1–8. IEEE, 2007. (Cited on page 85.)

[Wu 2013] Zheng Wu, Jianming Zhang and Margrit Betke. *Online Motion Agreement Tracking.* In BMVC, page 7, 2013. (Cited on page 65.)

[Xiang 2014] Yu Xiang, Roozbeh Mottaghi and Silvio Savarese. *Beyond pascal: A benchmark for 3d object detection in the wild.* In Applications of Computer Vision (WACV), pages 75–82. IEEE, 2014. (Cited on page 107.)

[Xiang 2017] Yu Xiang, Wongun Choi, Yuanqing Lin and Silvio Savarese. *Subcategory-aware convolutional neural networks for object proposals and detection.* In Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on, pages 924–933. IEEE, 2017. (Cited on page 14.)

[Xie 2015] Haihui Xie, Qingxiang Wu, Binshu Chen, Yanfeng Chen and Sanliang Hong. *Vehicle Detection in Open Parks Using a Convolutional Neural Network.* In Sixth International Conference on Intelligent Systems Design and

Engineering Applications (ISDEA), pages 927–930. IEEE, 2015. (Cited on page 37.)

[Xue 2008] Gui-Rong Xue, Wenyuan Dai, Qiang Yang and Yong Yu. *Topic-bridged PLSA for cross-domain text classification.* In the 31st annual international ACM SIGIR conference on Research and Development in Information Retrieval (RDIR), pages 627–634. ACM, 2008. (Cited on page 34.)

[Yang 2007] Jun Yang, Rong Yan and Alexander G Hauptmann. *Adapting SVM classifiers to data with shifted distributions.* In Seventh International Conference on Data Mining Workshops (ICDMW), pages 69–76. IEEE, 2007. (Cited on page 33.)

[Yang 2016] Fan Yang, Wongun Choi and Yuanqing Lin. *Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2129–2137, 2016. (Cited on page 14.)

[Yao 2011] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas and Li Fei-Fei. *Human action recognition by learning bases of action attributes and parts.* In International Conference on Computer Vision (ICCV), pages 1331–1338. IEEE, 2011. (Cited on page 34.)

[Ye 2017] Qixiang Ye, Tianliang Zhang, Wei Ke, Qiang Qiu, Jie Chen, Guillermo Sapiro and Baochang Zhang. *Self-Learning Scene-Specific Pedestrian Detectors Using a Progressive Latent Model.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017. (Cited on pages 37 and 85.)

[Yoo 2017] Haanju Yoo, Kikyung Kim, Moonsub Byeon, Younghan Jeon and Jin Young Choi. *Online Scheme for Multiple Camera Multiple Target Tracking Based on Multiple Hypothesis Tracking.* IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 3, pages 454–469, 2017. (Cited on page 65.)

[Zeiler 2014] Matthew D Zeiler and Rob Fergus. *Visualizing and understanding convolutional networks.* In European conference on computer vision, pages 818–833. Springer, 2014. (Cited on page 100.)

[Zeng 2014] Xingyu Zeng, Wanli Ouyang, Meng Wang and Xiaogang Wang. *Deep learning of scene-specific classifier for pedestrian detection.* In ECCV, pages 472–487. Springer, 2014. (Cited on pages 41, 55 and 56.)

[Zhang 2008] Cha Zhang, Raffay Hamid and Zhengyou Zhang. *Taylor expansion based classifier adaptation: Application to person detection.* In Computer Vision and Pattern Recognition, Conference on, pages 1–8. IEEE, 2008. (Cited on page 35.)

[Zhang 2016] Xiaofan Zhang, Feng Zhou, Yuanqing Lin and Shaoting Zhang. *Embedding label structures for fine-grained feature representation.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1114–1123, 2016. (Cited on page 66.)

[Zivkovic 2006] Zoran Zivkovic and Ferdinand Van Der Heijden. *Efficient adaptive density estimation per image pixel for the task of background subtraction.* Pattern recognition letters, vol. 27, no. 7, pages 773–780, 2006. (Cited on page 49.)

[Zweig 2007] Alon Zweig and Daphna Weinshall. *Exploiting object hierarchy: Combining models from different category levels.* In International Conference on Computer Vision (ICCV), pages 1–8. IEEE, 2007. (Cited on page 32.)