Ensemble Methods for Pedestrian Detection in Dense Crowds

Jennifer Vandoni

université PARIS-SACLAY

école normale supérieure paris-saclay

UNIVERSITÉ PARIS SUD
Comprendre le monde, construire l'avenir

NNT : 2019SACLS116

Thèse de doctorat

Ensemble Methods for Pedestrian Detection in Dense Crowds

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'Université Paris-Sud

École doctorale n°**580**
Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Traitement du Signal et des Images

Thèse présentée et soutenue à Gif-sur-Yvette, le 17 mai 2019, par

Jennifer Vandoni

Composition du Jury :

Julien Pettré Directeur de recherche, INRIA Rennes	Président
François Brémond Directeur de recherche, INRIA Sophia Antipolis	Rapporteur
John Klein Maître de conférences, Université de Lille	Rapporteur
Quoc Cuong Pham Chercheur, CEA-LIST	Examinateur
Sylvie Le Hégarat-Mascle Professeure des Universités, Université Paris-Saclay	Directrice de thèse
Emanuel Aldea Maître de conférences, Université Paris-Saclay	Encadrant

université PARIS-SACLAY | ÉCOLE DOCTORALE Sciences et technologies de l'information et de la communication (STIC)

SATIE

Acknowledgements

Firstly I would like to thank my family, especially Nicola and my parents, for always encouraging me to push my limits and for always being there for me. Then I would like to thank my Ph.D. advisors, Sylvie and Emi, for all the help, support and trust I received throughout these years. Last but not least, I want to thank all my friends, both the old ones ("*so close, no matter how far*") and the ones I met during the Ph.D, for all the good moments we spent and will spend together.

Abstract

The interest surrounding the study of crowd phenomena spanned during the last decade across multiple fields, including computer vision, physics, sociology, simulation and visualization. There are different levels of granularity at which crowd studies can be performed, namely a finer *micro-analysis*, aimed to detect and then track each pedestrian individually; and a coarser *macro-analysis*, aimed to model the crowd as a whole.

One of the most difficult challenges when working with human crowds is that usual pedestrian detection methodologies do not scale well to the case where only heads are visible, for a number of reasons such as absence of background, high visual homogeneity, small size of the objects, and heavy occlusions. For this reason, most micro-analysis studies by means of pedestrian detection and tracking methodologies are performed in low to medium-density crowds, whereas macro-analysis through density estimation and people counting is more suited in presence of high-density crowds, where the exact position of each individual is not necessary.

Nevertheless, in order to analyze specific events involving high-density crowds for monitoring the flow and preventing disasters such as stampedes, a complete understanding of the scene must be reached. This study deals with pedestrian detection in high-density crowds from a mono-camera system, striving to obtain localized detections of all the individuals which are part of an extremely dense crowd. The detections can be then used both to obtain robust density estimation, and to initialize a tracking algorithm.

In presence of difficult problems such as our application, supervised learning techniques are well suited. However, two different questions arise, namely which classifier is the most adapted for the considered environment, and which data to use to learn from.

We cast the detection problem as a Multiple Classifier System (MCS), composed by two different ensembles of classifiers, the first one based on SVM (SVM-ensemble) and the second one based on CNN (CNN-ensemble), combined relying on the Belief Function Theory (BFT) designing a fusion method which is able to exploit their strengths for pixel-wise classification.

SVM-ensemble is composed by several SVM detectors based on different gradient, texture and orientation descriptors, able to tackle the problem from different perspectives. BFT allows us to take into account the imprecision in addition to the uncertainty value provided by each classifier, which we consider coming from possible errors in the calibration procedure and from pixel neighbor's heterogeneity in the image space due to the close resolution of the target (head) and descriptor respectively.

However, scarcity of labeled data for specific dense crowd contexts reflects in the impossibility to easily obtain robust training and validation sets. By exploiting belief functions directly derived from the classifiers' combination, we therefore propose an evidential Query-by-Committee (QBC) active learning algorithm to automatically select the most informative training samples.

On the other side, we explore deep learning techniques by casting the problem as a segmentation task in presence of soft labels, with a fully convolutional network architecture designed to recover small objects (heads) thanks to a tailored use of dilated convolutions. In order to obtain a pixel-wise measure of reliability about the network's predictions, we create a CNN-ensemble by means of dropout at inference time, and we combine the different obtained realizations in the context of BFT.

To conclude, we show that the dense output map given by the MCS can be employed not only for pedestrian detection at microscopic level, but also to perform macroscopic analysis, bridging

the gap between the two levels of granularity. We therefore finally focus our attention to people counting, proposing an evaluation method that can be applied at every scale, resulting to be more precise in the error and uncertainty evaluation (disregarding possible compensations) as well as more useful for the modeling community that could use it to improve and validate *local* density estimation.

Synthèse en français

L'étude des phénomènes liés aux foules a évolué durant la dernière décennie en s'étendant à plusieurs domaines dont la vision par ordinateur, la physique, la sociologie, ou la simulation et visualisation. Les études sur les foules peuvent être réalisées à différents niveaux de granularité, ainsi la micro-analyse vise à detecter et suivre chaque piéton individuellement, tandis que la macro-analyse vise à modéliser la foule comme un unique objet (système) déformable.

L'un des défis de l'analyse de foules en vision par ordinateur est que les méthodologies classiques utilisées pour la détection de piétons s'adaptent mal au cas où seulement les têtes sont visibles, de part l'absence d'arrière plan, l'homogénéité visuelle de la foule, la petite taille des objets et la présence d'occultations très forts. Pour cette raison, la plupart des analyses microscopiques exploitant des flux vidéo sont effectuées dans le cas de foules de faible ou moyenne densité, tandis que les cas de foules très denses sont traités par analyse macroscopique basée sur l'estimation de champs de densité s'affranchissant de la connaissance de la position exacte de chaque individu.

Toutefois, une analyse microscopique de la scène est nécessaire pour contrôler le flux et prévenir des catastrophes tels que des bousculades y compris et notamment dans des foules très denses. Cette thèse s'intéresse alors à la détection des piétons dans des foules très denses depuis un système mono-camera, avec comme but d'obtenir des détections localisées de toutes les personnes. Ces détections peuvent être utilisées soit pour obtenir un estimation robuste de la densité, soit pour initialiser un algorithme de suivi. En présence des problèmes difficiles tels que notre application, les approches à base d'apprentissage supervisé sont bien adaptées. Les deux questions qui en découlent sont quel classifieur et quelles données utiliser pour l'apprentissage.

Pour notre problème de détection, nous considérons un système à plusieurs classifieurs (Multiple Classifier System, MCS), composé de deux ensembles différents, le premier basé sur les classifieurs SVM (SVM-ensemble) et le deuxième basé sur les CNN (CNN-ensemble), et combinés dans le cadre de la Théorie des Fonctions de Croyance.

Précisément, l'ensemble SVM est composé de plusieurs classifieurs SVM chacun exploitant les données issues d'un descripteur différent (gradient, texture et orientation), afin d'appréhender différentes caractéristiques des objets recherchés, à savoir les têtes des piétons. La Théorie des Fonctions de Croyance nous permet de prendre en compte, en sus de la valeur d'incertitude fournie par chaque classifieur, une valeur d'imprécision supposée correspondre soit à une imprécision dans la procédure de calibration, soit à une imprécision spatiale due à la résolution des objets recherchés versus les descripteurs considérés.

Cependant, le manque de données labellisées pour le cas spécifique des foules très denses nuit à la génération d'ensembles de données d'entrainement et de validation robustes. Nous avons alors proposé un algorithme d'apprentissage actif de type Query-by-Committee (QBC) qui permet de sélectionner automatiquement de nouveaux échantillons d'apprentissage. Cet algorithme s'appuie sur des mesures évidentielles déduites des fonctions de croyance modélisant l'information issue des différents classifieurs.

Pour le second ensemble, pour exploiter les avancées de l'apprentissage profond, nous avons reformulé notre problème comme une tâche de segmentation en soft labels. Une architecture entiérement convolutionelle a été conçue pour détecter les petits objets grâce à un ensemble de convolutions dilatées. Nous nous sommes appuyés sur la technique du dropout pour obtenir un ensemble CNN capable d'évaluer la fiabilité sur les prédictions du réseau lors de l'inférence. Les réalisations de cet ensemble sont ensuite combinées dans le cadre des Fonctions de Croyance.

Pour conclure, nous montrons que la sortie du MCS peut être utile non seulement pour la détection de piétons au niveau microscopique, mais aussi pour l'analyse macroscopique, ce qui nous a permis de relier les deux niveaux de granularité. Pour le comptage de personnes, nous avons proposé une méthodologie d'evaluation multi-échelle, qui est à la fois plus informative car elle contraint les compensations d'erreur, et très utile pour la communauté de modélisation car elle lie incertitude (probabilité d'erreur) et imprécision sur les valeurs de densité estimées.

Introduction

The problem

The study of crowded scenes gained traction in recent years, due to the increased frequency of large scale social events, and due to the security risks linked to this context. Applications in this area span from pedestrian detection for video surveillance, to human behaviour analysis and understanding, and to crowd density estimation. Despite the continuous improvement of computer vision and machine learning techniques, several complex problems still remain, especially in presence of high-density crowd situations, which are indeed the most dangerous and need more effort in order to understand the mechanisms that lead to crushes and stampedes possibly causing loss of human lives.

A crowd, defined in [190] as "a large group of individuals in the same physical environment, sharing a common goal", is far more than a simple sum of individuals. It can assume different and complex behaviours with respect to those of its composing individuals if they were alone. Collective characteristics can emerge, and people can lose their individualities and adopt the behaviour of the crowd entity.

Safe crowd conditions can be usually assumed for densities up to two-three persons per square meter, and a maximum acceptable flow of 80 persons per meter and minute. Progressive crowd collapse usually occurs at a density of about six or seven persons per square meter, although the situation can start to be dangerous with a density of four or five persons per square meter where congestion can arise quickly, which implies high risk for people to stumble or fall (particularly in presence of uneven ground). Crushes often happen during religious pilgrimages or large entertainment events, where people are surrounded by other individuals on all the sides and cannot move freely. Besides, people in a dense crowd cannot see what happens a few meters away from them, and they are not aware of the pressure in front. In those situations, episodes of panic are frequent as people feel constricted and cannot breathe, causing a natural desire to leave the crowd. As people try to get away however, they cause actual waves, that advance progressively and inexorably towards the other people close to them, resulting in a domino effect where it is difficult for the people involved to avoid falling and being trampled.

Problems related to queues of people aggravate this *stop-and-go wave* phenomenon. In such situations, people will subconsciously reduce their distance and the so-called *queuing effect* will create the impression of progress. If people have to wait long and are not informed about the reason, they will indeed become impatient and may eventually start to push intentionally, assuming that the progress can be accelerated. However, this will eventually cause a substantial compression of the crowd, especially in the front of the queue. When the distance between people is small, there will be also inadvertent body contacts, which can cause unintentional pushing and an inevitably reduction of the *proxemics* [96], the personal space of each individual that represents the comfort zone during interpersonal communication. The transition from an acceptable situation with rare body contacts to a stressful situation with frequent body contacts can happen quite abruptly, and pushing others away can become unavoidable in order to be able to breathe. In fact, excessive body compression can cause asphyxia, and the thermal heat of surrounding people can cause weakening and fainting.

Unfortunately, during the last decades there have been numerous examples of such episodes,

distributed all over the world. We can mention here the 1989 Hillsborough disaster, that with 96 fatalities and 766 injuries is the worst disaster in British sporting history; the crowd disaster in 2010 during the Love Parade in Duisburg, where 21 people died and more than 500 were injured; the 2014 Shanghai stampede, where 36 people were killed and 47 injured during the New Year's celebrations that gathered around 300.000 people in the city. Well documented episodes are the numerous tragedies that happened at Makkah during the Hajj, the annual Islamic pilgrimage, when the region has to accommodate nearly three million pilgrims in one month. In particular, in 1990, 1426 people were suffocated and trampled to death inside a pedestrian tunnel near Makkah. In 2006, a stampede during the Stoning of the Devil ritual on the last day of the Hajj in Mina killed at least 346 pilgrims and injured at least 289 more. A similar accident happened very recently, in 2015, where more than 2000 people were killed as soon as two large groups of pilgrims intersected from different directions in the same street, in an area that was not previously identified as a dangerous bottleneck. The high number of deaths caused by this accident makes it the deadliest Hajj disaster in history.

It is believed that most major crowd disasters can be prevented by simple crowd management strategies and monitoring. For example, by analysing some of the video sequences recorded during the unfortunate event in Mina of 2006, the authors of [103] report abnormal patterns in the flow that could have possibly been identified as early as 30 minutes before the tragedy happened. In the same way, a posterior study of the Duisburg's Love Parade disaster [105] through analysis of publicly available video sequences helped in defining the most critical moments of the unfortunate event, that happened due to a series of contributory causes. The authors provided also a table to help assess the level of criticality of the situation in the crowd, stating that the presence of stop-and-go waves is an indicator that the outflow capacity is considerably reduced and the situation may thus escalate quickly. These studies prove that in both cases an automated video analysis system would have helped to understand what was going to happen, still being in time for taking proactive measures to avoid or at least mitigate crowd disasters and preventing the ensuing stampedes through carefully planned security measures.

Since both the world population and the number of large scale social events continue to increase, it becomes more and more important to study the causes of these tragedies in order to be able to design better infrastructures the could help in preventing future accidents while at the same time increasing and not diminishing the comfort of participants.

As pointed out in [129, 163], crowded scene study becomes thus important for a number of different yet interconnected applications:

- *Visual surveillance* – It is related to the implementation of automatic or semi-automatic surveillance systems for public spaces that every day gather thousands of people, like railway stations or shopping malls. Conventional surveillance systems expect one or more human observers in charge of monitoring video streams coming from an always increasing number of camera sources. However, psycho-physical research has stressed severe limitations in their ability to monitor simultaneous signals [252]. Alternatively, in an almost automated setting computer vision algorithms could help safety and security personnel in their tasks of anomaly detection and raising alarms, providing flux statistics, congestion analysis and prevention;

- *Crowd management* – It can be used for developing strategies to avoid crowd disasters like the ones mentioned above, and ensure public safety during mass gathering events, for example by preventing people to enter in highly crowded areas through access control or by deviating the mass flow to avoid the creation of queues and bottlenecks. It can be useful also for the creation of *intelligent environments*, e.g. taking real-time decisions on how to split a crowd in a museum, based on the behaviour of the mass;

- *Public space design* – It is useful to perform a-priori studies and modeling of public spaces, to be able to design better infrastructures that increase the efficiency and safety of crowded areas such as train stations, airport terminals or buildings and open spaces for large scale

events, through the implementation of ad-hoc barriers and emergency exits at particular locations. It can be used to validate or empower with real data mathematical models used in crowd simulations as well.

Although the social relevance of the possible applications related to crowd studies, several challenges still remain and need to be tackled by the different scientific communities, such as Computer Vision, Applied Mathematics and Physics, Cognitive Psychology, Computer Graphics, and possibly others. This makes crowd study a complex and highly multidisciplinary field of study. However, despite the fact that the different scientific fields are studying the same physical entity, i.e. a crowd composed by pedestrians, research ideas have evolved almost independently. The variety of aspects that can be investigated reflects in the impossibility of having an unified framework and in a lack of shared baselines on real scenarios.

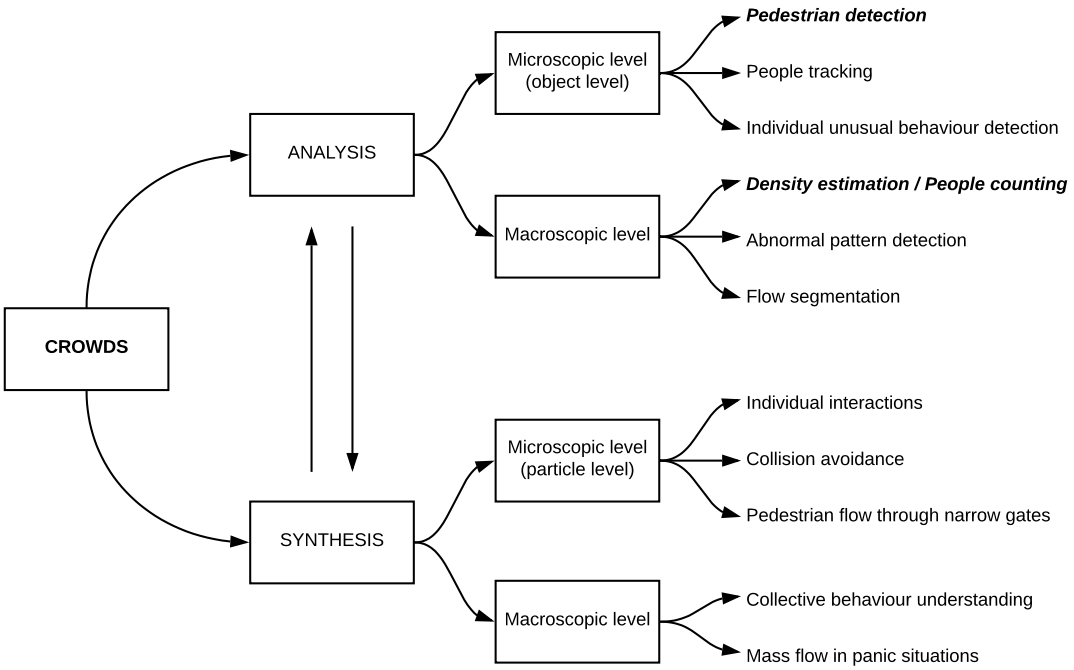


Figure 1: Proposed crowd taxonomy for crowd analysis and synthesis. The topics explored in the context of this work are highlighted with bold, italic text.

To this extent, Fig. 1 proposes a taxonomy for crowd understanding, which includes two main fields of study:

- *Crowd Analysis*, usually investigated by the Computer Vision community, aimed to analyze real scenes for visual surveillance and crowd management;

- *Crowd Synthesis*, usually performed by Mathematics, Physics and Computer Graphics communities, aimed to model the crowds in different scenarios, and to assist with infrastructure and large scale event design through realistic simulations.

These two fields of study are indeed intrinsically connected. On the one hand, in the context of crowd synthesis there exists the need to exploit real data concerning crowd dynamics to be able to validate, calibrate and integrate macroscopic and microscopic models, in order to obtain always better and more realistic simulations. On the other hand, simulated data could be useful to perform crowd analysis, both for augmenting the available training information regarding emergency situations which is usually very scarce, and to create a common baseline for the validation of the obtained results.

Besides, both crowd analysis and synthesis can be tackled at different levels of granularity, depending on the interpretation of the concept of crowd:

- The *Microscopic* level, that interprets the crowd as a (possibly structured) collection of pedestrians maintaining their own individualities;
- The *Macroscopic* level, that models the crowd as a single entity.

Both levels of granularity are equally important as they address different but nevertheless crucial tasks for crowd understanding. If the first type strives to study the interactions among various individual behaviours, by means of pedestrian detection and tracking, the second type takes a broader view of the scene and computes indicators that can be later exploited in a model, both for collective motion analysis in macroscopic simulations, and to validate simulated data with cues like density estimation performed on real data.

Outline of the work

This work deals with single-camera crowd analysis, and it is organized to span between the two levels of granularity, partially bridging the gap between theoretical field and the actual phenomena, maintaining a Computer Vision perspective yet promoting and facilitating a joint effort between the different scientific fields. Indeed, our final aim is to propose a method for crowd analysis that can be easily integrated in simulation models, from both microscopic and macroscopic points of view.

Our objective is to perform pedestrian detection in high-density crowds from a mono-camera system. The problem is slightly different with respect to previous works reported in literature, in that high-density crowds are usually macroscopically studied (e.g. for density estimation purposes), whereas individual detection is usually performed only in presence of low to medium density crowds. From our side, we strive to obtain localized detections of all the individuals which are part of the dense crowd. The detections can be then used both to initialize a tracking algorithm, and to obtain robust density estimation.

We start in Chapter 1 by an extensive examination of the state of art about the different fields of study related to crowd understanding, namely crowd synthesis and analysis, highlighting their complementarity and the need of a joint effort between the research communities.

Then, Chapter 2 introduces supervised learning along with classifier combination techniques, which are the two main "building blocks" of the proposed approach which is indeed a Multiple Classifier System (MCS) composed by two ensemble of classifiers, based on SVM (SVM-ensemble) and CNN (CNN-ensemble) respectively.

Concerning SVM-ensemble, Chapter 3 highlights, among several SVM descriptors for pedestrian detection, those that are more adapted in the context of high-density crowds, relying on different gradient, texture and orientation information, fact that increases the complementarity among them which will be particularly useful when performing fusion.

To this extent, in Chapter 4, after a brief recall about Belief Function fundamentals, we propose a fusion method among the SVM-ensemble members which is based on the Belief Function Theory (BFT) in order to take into account the imprecision in addition to the uncertainty value provided by each classifier. We focus our attention on two different types of imprecision that can arise and are worth being better examined. Firstly, SVM is in its standard form a crisp classifier. In order to obtain class probabilities, logistic regression is usually performed with respect to a calibration set, considering the distances from the calibration samples to the learned separation hyperplane. However, this process may be quite dependent on the chosen calibration set, especially in presence of overlapping classes due to the impossibility of finding a perfect separation plane. Thus, a first source of imprecision is taken into account by obtaining a Basic Belief Assignment (BBA) for every sample out of its SVM classification score, i.e. distance from the hyperplane, through a discounting operation. Secondly, we note that the classification is performed densely for every pixel of the tested image. Although neighboring pixels should provide similar values of uncertainty, in practice this could not happen due to the close resolution of the detector and the target heads respectively. For this reason, a second discounting is applied in the spatial domain, with respect

to the heterogeneousness of a pixel neighborhood. The proposed BBA allocation takes thus into account the local decision for every pixel and goes beyond a single value which is only based on a global reliability computed for the whole classifier.

After BBA allocation, BFT provides specific rules for the fusion of the different sources of information. However, we note that final results can be quite dependent on the heterogeneity of the training set being used. Finding the most informative samples to perform the training would lead to a better class separation hyperplane, reducing thus the amount of imprecision out of the classification. To this extent, Active Learning (AL) algorithms are designed in order to automatically select the data from which to learn. We thus propose in Chapter 5 an evidential Query-By-Committee (QBC) AL method that incrementally augments the training set with samples on which the committee of classifiers, built from the SVM-ensemble, does not agree, quantifying the level of disagreement with different evidential measures. We prove that our evidential QBC active learning algorithm built from a set of carefully chosen classifiers is able to exploit at maximum the available information, reaching high levels of accuracy also in presence of a small training set. In addition, it leads to the simultaneous improvement of the single classifiers performance.

This approach is particularly well suited for applications where a complete analysis of specific scenes is required but at the same time the available labeled data for that specific scene is scarce. In such situations, it is impossible to blindly apply recent deep learning techniques based on complex models that are usually better in exploiting large quantity of heterogeneous data. Nevertheless, in Chapter 6 we propose a deep learning based solution which casts the problem as segmentation in presence of soft labels to perform pedestrian detection, with a network architecture especially designed to recover small targets thanks to a tailored use of dilated convolutions.

The proposed network is then trained on a very limited amount of data. In such situations, the reliability of pixel-wise predictions becomes a critical information to take into account, and in order to obtain it in Chapter 7 we cast the learning model as a Bayesian Neural Network by applying dropout at inference time to obtain several realizations of the same perturbed network (referred to as the CNN-ensemble). Instead of just computing the standard deviation out of this ensemble, we propose a BBA allocation that is based on a discounting of each source on the basis of its deviation from the median value of the distribution. The evidential combination then allows us to obtain evidential measures of imprecision that can be interpreted as measures of robustness and reliability of the network. The final MCS composed by both SVM-ensemble and CNN-ensemble is finally presented, showing that in presence of scarce labeled data for the analysis of specific scenes we can still exploit deep learning methods (with particular attention to regularization techniques) by combining them with traditional classifiers such as SVM based on hand-crafted features and more adapted to work with less data (selected through active learning).

Lastly, in Chapter 8 we show how the obtained dense map out of the MCS for pedestrian detection (and thus microscopic analysis) can be employed to perform also macroscopic analysis. We finally focus our attention on people counting, proposing a validation method that can be applied at every scale, resulting in a more precise error evaluation (disregarding possible local compensations) as well as a more pertinent method for the modeling community that could use it to improve and validate *local* density estimation, partially bridging the gap between analysis and synthesis.

The thesis terminates with conclusion and perspectives, with a particular attention to the possible exploitation of temporal information (coming e.g. from optical flow) to perform data association between detections at different frames.

Contents

List of Figures

List of Tables

Glossary

AL - *Active Learning*

AUPRC - *Area Under Precision-Recall Curve*

BIF - *Biologically Inspired Features*

BBA - *Basic Belief Assignment*

BF - *Belief Function*

BFT - *Belief Function Theory*

BKS - *Behavior-Knowledge Space*

BNN - *Bayesian Neural Network*

CCNN - *Counting Convolutional Neural Network*

CD - *Counting by Detection*

CDE - *Counting by Density Estimation*

CNN - *Convolutional Neural Network*

CR - *Counting by Regression*

CRF - *Conditional Random Field*

ECOC - *Error Correcting Output Code*

ELF - *Ensemble of Local Features*

EMOC - *Expected Model Output Change*

FCN - *Fully Convolutional Network*

FE - *Front End*

FN - *False Negative*

FP - *False Positive*

GMM - *Gaussian Mixture Model*

GLCM - *Grey Level Co-occurrence Matrix*

HIK - *Histogram Intersection Kernel*

HMM - *Hidden Markov Model*

HOG - *Histogram of Oriented Gradients*

ICF - *Integrate Channel Features*

IoU - *Intersection over Union*

KL - *Kullback-Leibler*

KLT - *Kanade-Lucas-Tomasi*

LFE - *Local Feature Extraction*

MAD - *Median Absolute Deviation*

MAE - *Mean Absolute Error*

MC - *Maximum Conflict*

MCNN - Multi-column Convolutional Neural Network

MCS - *Multiple Classifier System*

MDNN *Multi-column Deep Neural Network*

ME - *Maximum Entropy*

MI - *Maximum Ignorance*

MKL - *Multiple Kernel Learning*

MSE - *Mean Squared Error*

MRF - *Markov Random Field*

NMS - *Non-Maximum Suppression*

NN - *Neural Network*

P3D - *Pseudo-3D*

PCA - *Principal Component Analysis*

PEP - *Prediction Error Probability*

PF - *Particle Filter*

PRBEP - *Precision-Recall Break Even Point*

QBC - *Query-By-Committee*

RBF - *Radial Basis Function*

ReLU - *Rectified Linear Unit*

RI - *Relative Imprecision*

RMSE - *Root Mean Squared Error*

RPN - *Region Proposal Network*

SDALF - *Symmetry-Driven Accumulation of Local Features*

SIFT - *Scale Invariant Feature Transform*

SMKL - *Selective Multiple Kernel Learning*

SMO - *Sequential Minimal Optimization*

SRT - *Stochastic Regularization Technique*

SURF - *Speeded Up Robust Features*

SVE - *Soft Vote Entropy*

SVM - *Support Vector Machine*

TN - *True Negative*

TP - *True Positive*

ViF - *Violent Flow*

Chapter 1

Crowd understanding

Contents

1.1 State of the art

Following the taxonomy proposed in Fig. 1, we present the state-of-the-art methods for both crowd analysis and synthesis. Even though this work deals with crowd analysis, we find it useful to briefly recall the proposed models used by the simulation community, to highlight once again the complementarity between the two fields.

1.1.1 Crowd Synthesis

Pedestrian flow can be modeled both at microscopic level, by postulating rules for the behaviour of each individual with respect to his/her goal and to the possible interactions with other people or obstacles, or at macroscopic level, by considering the pedestrian mass in an aggregate way. On the one hand, microscopic approaches are useful to model individual interactions and collision avoidance mechanisms, but they are computationally expensive as each individual is related to a dynamics equation, constrained by the solutions for neighboring pedestrians, to be solved at each timestep. Besides, the size of the system to be solved depends on the number of people to be modeled. On the other hand, macroscopic models are computationally less expensive, allowing to treat analytically much larger environments, but they lack in modeling single behaviours, although being useful for collective motion analysis.

Regarding the microscopic description, several models have been proposed over the past decades. One of the most popular ones is the *Social Force Model* [104] for pedestrian movement dynamics. The model describes pedestrian motion dynamics taking into account personal goals and environmental constraints. The motion of each individual is indeed determined by an attractive force toward his/her destination, and a repulsive force with other individuals and objects in the neighborhood. Parameters can be tuned in order to model the urgency and aggressiveness of each individual. The motivation behind this model comes from the *Least-Effort Hypothesis*, by which people try to choose the least-effort route to reach their goal, together with the observed psychological tendency of human beings to maintain a social distance among individuals. However, it has been noted that the repulsive term is too strict in some cases, as it is not linearly dependent on the

number of pedestrians . This model has been later extended by including additional terms to simulate different behaviours, such as socially bounded groups of people moving together, queuing behavior, and behavior in case of a panic situations [102, 189].

Recently, extensive experiments have been conducted to study the microscopic dynamics of pedestrian flows through narrow doorways [194]. The flow results to be orderly for polite crowds, with narrowly distributed time lapses between egresses, while increasing the fraction of participants with selfish behaviour the flow gets disorderly and vanishing time lapses tend to emerge. Regardless of the behaviours of the participants to the experiments, the flow rate and other flow properties such as the disorder in the passages and the pressure perceived by the participants exhibit a simple dependence on the density in the exit zone. This suggests that in a macroscopic approach, for a given composition of the crowd, the behavioural aspects can be left aside and confirms the key role played by the density parameter in determining the flow rate. However, in the experiments people were not allowed to push their neighbours. In a real panic scenario, at high densities the global flow rate will strongly depend on other parameters, such as the pressure in the crowd.

Motivated by the difficulty to put in place large-scale experiments involving moving crowds, the same authors of [194] performed another set of tests by exploiting the analogies between crowds and granular matter, in presence of a static dense crowd perturbed by a cylindrical "intruder" [195]. Intruding a cylinder into a medium is actually a classical mechanical test, that allows for the discrimination between granular media and viscous fluids. In the case of the crowd, the authors found some similarities with respect to granular matter, such as a depletion behind the intruder after his passage, and a fast decay of the perturbation in the transverse direction. Nevertheless, they found also some differences, such as the absence of high-density formed in front of the intruder, and the displacement of the individuals almost exclusively directed laterally (outward or inward), at odds with the loop-like pattern with retro-circulation eddies seen in grains. However, these differences were only visible when people were allowed to see the intruder approaching them, while if the intruder was arriving from the back of the crowd, the analogy resulted to be verified. This fact is in accordance with [118] where the crowd is seen as a *thinking fluid*. Indeed, the crowd presents some specificities which are proper to pedestrians, which are able to anticipate and initiate a movement after a sensory stimulus (e.g. visual or auditory), without the need of the touch of other grains in order to be able to move (self-propulsion).

Instead of setting the path of each person individually, the *Boids Model* [224] was proposed to simulate the aggregate motion of groups of individuals using local rules. In particular, it aims to reproduce the collective motion of a flock of birds using three rules that regulate the interactions, namely separation, alignment and cohesion. The separation rule provides the collision-avoidance behaviour, the alignment rule influences the individual velocity and direction such that it results to be aligned with the neighborhood, while the cohesion rule simulate the tendency of an individual to move closer to the average position of the local neighbours. This model reflects the *lane formation phenomenon*, by which it takes less effort for people to follow immediately behind someone that is already moving in the same direction, rather than push others in the crowd.

The concept of crowd as a thinking fluid has been further developed in the *Continuum Crowd Model* [261] to produce more realistic crowd behavior, being able to capture phenomena including lane formation and short lived vortices during turbulent congestions. However, the continuum crowd model is not appropriate for all crowd behaviors. For example, it does not take into account the case where people are so tightly packed that contact forces between them dominate the physics. It is also limited by the requirement that people move with a common goal. This in particular could not be true in panic situations.

To this extent, macroscopic models for pedestrian flows specifically for panic situations have been proposed [44, 89]. They are able to describe mass evacuation from a narrow corridor or bridge, assuming that the escaping pedestrians have to pass through an exit after having passed through an obstacle to regulate the evacuation process. The presence of the mentioned obstacle indeed, following the Braess' paradox, has been shown to favour a decrease in the time of evacua-

tion, if carefully placed in a well-chosen position in front of the exit.

Macroscopic models can be also grounded on microscopic phenomenological observations. This is the case of [29], that studies the collective evolution of crowds along footbridges from individual behaviours. In particular, the authors considered the macroscopic model proposed in [208] and they provide a mathematical procedure to obtain, out of the equations governing the motion of single individuals, an equation describing the collective evolution of the crowd. This work is particularly interesting in that it stresses, in the context of crowd synthesis, the possibility of microscopic and macroscopic descriptions to be employed altogether for a common cause.

1.1.2 Crowd Analysis

In the same spirit of crowd synthesis, crowd analysis as well can be performed at different levels of granularity:

- A finer *micro-analysis*, aimed to detect and then track each pedestrian individually, usually applied in presence of low-density to medium-density crowds;
- A coarser *macro-analysis*, aimed to study the crowd as a whole, particularly suitable for very high-density crowds.

From a Computer Vision perspective, this two levels assume completely different approaches that should be thus inspected independently.

1.1.2.1 Micro-Analysis

Microscopic analysis of crowds relies on the analysis of trajectories of the various moving entities extracted from video sequences. This approach is generally divided in different steps:

1. Detection of the targets in the scene;
2. Tracking of the detected targets;
3. Analysis of the trajectories to extract dominant flow, unusual behaviours, etc.

Regarding the detection of the targets, although in the last years many efforts have been devoted to improve the performance of pedestrian detection, baseline methods cannot be always applied in crowded contexts.

Pedestrian detection by itself is noticeably one of the most challenging categories of object detection. There exists indeed a large variability in the local and global pedestrian's appearance, due to the variety of possible body shapes, or different styles and types of clothes and accessories which may perturb the silhouette of the individuals. Besides, in real-world scenarios several people can occupy the same region, partially occluding each other, and this phenomenon tends to become not negligible with the increase of crowd density. The advances proposed in the literature are not usually transferable to high-density crowd detections for multiple reasons, among which we can recall the absence of background, the heavy occlusion of body parts, the high visual homogeneity of the scene and the small size of the targets.

Traditionally, in the context of supervised learning the Histogram of Oriented Gradients (HOG) descriptor [51] has been proposed for the scope, but its performance can be easily affected by the presence of background clutter and occlusions. Alternatively, deformable part-based models [74] consider the appearance of each part of the body and the deformation among parts for detection. Pedestrians are modeled as collections of parts, firstly generated by learning local features such as edgelets and orientation features, but in presence of severe occlusions and high visual homogeneity the various body parts can be hidden, and the difference between them becomes too slightly to be exploited as clue. Background subtraction is also usually employed, to perform motion-based detections. Beside removing potentially significant parts of the scene which do not contain objects

of interest, background subtraction allows for the use of descriptors built upon the blobs associated to the foreground, such as their skeleton or the shape of the foreground connected components [183]. Unfortunately, in a cluttered scene this approach is ineffective due to the limited presence of background. Recently, neural networks which make use of a region proposal step have been employed in the context of pedestrian detection, in conjunction with hand-crafted features based on variants of Integrate Channel Features [183] (ICF) detector [110, 257], or stand-alone [158, 294, 295]. Late fusion of multiple convolutional layers has been recently proposed in [263] relying on Region Proposal Networks (RPNs), showing that earlier convolutional layers are better at handling small-scale and partially occluded pedestrians. Again, in presence of dense crowds, region proposal step loses its interest as the number of targets becomes too large to be tractable.

For all the highlighted limitations, a straightforward extension of the techniques designed for pedestrian detection in non-crowded scenes is not suitable for dealing with crowded situations. Furthermore, pedestrian detection in crowds highly depends on the level of crowd density, and methods adapted to lower-density crowds may fail as the number of people in the scene increases.

In low-density crowds, object-level analysis can be successfully performed to identify the individuals in the scene. Haar wavelet transform can be used to extract the areas of the head-like contour [167]. Alternatively, a combination between local and global features can be exploited to obtain the probability of a person being present, comparing small patches of learned human appearance and occurrence distribution [154]. Temporal information can be also exploited in order to build a spatio-temporal descriptor based on 3D gradients [140], or exploiting a cascade of classifiers of Haar-like features trained to deal with different motion directions [127]. However, these methods are not suited to handle the presence of too many occlusions.

To this extent, a multiple camera setting can be exploited. The use of multiple cameras for video analysis (mainly surveillance) is an extensive topic [2, 124], that has been applied also to pedestrian detection in crowds. A small scale experiment has been proposed in [71], proving the potential of multiple camera tracking in occluded scenes. This study proposes an effective solution for exploiting jointly hypotheses related to the presence of a head in multiple cameras, where the consistency is evaluated using the pixel intensity information. Other methods [33, 125, 206, 214] heavily rely on foreground extraction, while [137] even requires feet visibility in order to work.

Unfortunately, cameras are usually placed with low pitch angles and would not observe sufficient empty areas among proximate pedestrians in order to benefit from foreground extraction, and frequent occlusions may make people feet invisible. Recently, [205] tackles these problems by performing multiple camera based pedestrian detection exploiting low level information fusion. The authors propose an unsupervised detection method which exploits the visual consistency of the pixels in multiple views in order to estimate the pedestrian occupation, without the necessity of performing any background segmentation and showing good performance even in presence of high visual homogeneity. However, in all these works the considered scenes are rather small and the crowd density is not extreme. Extending these types of solution to large scale scenarios raises several difficult problems. Nevertheless, in many video surveillance circumstances several cameras may not be available due to limitations of the infrastructures.

In a mono-camera situation and in presence of dense crowds, like the setting we are considering in this work, classifiers are usually trained to recognize heads, which in an occluded environment are almost the only visible part of a person. Cameras tend indeed to be placed above the heads of the people and tilted to face the scene downwards, thus reducing the amount of occluded heads with respect to other body parts, allowing for the detection of head-shoulder aggregations, i.e. the so-called Ω-*shape* [161]. Single classifiers however may fail due to the complexity of the problem, and it becomes therefore essential to rely on multiple complementary visual detectors which are able to provide different interpretations of the input data. The reader is referred to Sec. 3.1 for a more detailed digression about state-of-the-art pedestrian detectors.

The obtained detections may then be used to initialize a tracker. Initial attempts [301, 302] proposed effective approaches based on mean-shift [45] or based on 3D human models integrated into a Bayesian framework, but these methods cannot handle properly persistent occlusions or

multiple close-by subjects. A feature tracking algorithm, namely the Kanade-Lucas-Tomasi (KLT) tracker [243], has been used in order to analyze the coherence of the movement through clustering and assist the detection process to segment individuals in a crowd [218]. A clear limitation however is the applicability of the method to only almost stationary crowds. KLT tracker has been employed also in [226], after having obtained initial detections through the minimization of a joint energy function incorporating scores of individual detections and local density estimation. Optical flow is then used in [26] in conjunction with a totally unsupervised Bayesian clustering to detect targets based on the assumption that points with similar motion vectors should be part of the same entity. However, since rigid motion is assumed, the algorithm may fail in presence of arm movements or in presence of particularly dense flow in the same direction.

Particle Filter (PF) framework can be used to perform visual tracking [16]. Over the years, a number of extensions to the original framework that was only based on colour clues were proposed, to be able to exploit a combination of colour and contour features [216], and to be able to track multiple targets simultaneously [5, 198]. However, in presence of too many targets PF solutions can become intractable, and optimizing detection assignments over a temporal window scales more conveniently for a large number of targets [56, 271].

Finally, the knowledge of individual trajectories in a crowd can be exploited to identify the main flow and detect possible unusual behaviours from some individuals. To this extent, [40] presents a framework to automatically identify dominant motions in crowded scenes, independently tracking low-level features using optical flow and clustering them into trajectories based on longest common subsequences. Individual motions which are not coherent with dominant flows are highlighted and marked as potentially unusual behaviours (e.g. a person making a U-turn where all the people move towards the same goal). Without the need of tracking as well as human labeling, [274] clusters moving pixels into atomic activities in a completely unsupervised way, allowing to discover the different interactions in the crowd and to detect anomalies. However, all these methods assumes dominant motions from which individual unusual behaviours can be extracted, and they cannot be applied in case of unstructured crowded scenes where the motion of individuals within a crowd appears to be random, with different participants moving in different directions over time (although in case of high-density crowds this assumption could hold since people proximity limits the freedom of movement).

Lastly, the complete knowledge of all the pedestrians composing a scene can be useful also for the related application of people counting [160]. *Detection-based* approaches require indeed a preliminary detection step, usually performed in a sliding window fashion, that allows for an automatic inference of the number of people present in the scene by simply counting the number of total detections obtained. Again, although being a successful strategy in presence of low density crowd scenes, this method fails with higher levels of density characterized by strong occlusions and background clutter.

1.1.2.2 Macro-Analysis

Macroscopic analysis, also referred to as *holistic*, interprets the crowd as a unique entity, being particularly convenient when the crowd starts to become denser. This type of analysis is usually employed for two different tasks which may assume different input data:

- People counting and density estimation, usually performed from still images coming from possibly completely different scenes;
- Flow segmentation and abnormal motion pattern detection, assuming the availability of video sequences.

Recently, automated crowd density estimation and counting has received attention for safety control, playing an essential role in crowd monitoring. It can be useful indeed for measuring the comfort level of the crowd and for preventing potential overcrowded situations. Besides the aforementioned detection-based approaches, there exist also the so-called *regression-based* ones,

which are indeed holistic methods more appropriate in presence of high-density crowds with strong occlusions and clutter. This methods attempt to learn a mapping between features extracted from local image patches to their counts, in order to free from the necessity to formerly localize each target. The initial works mainly use hand-crafted features [119, 155], while the more recent works are rather based on Convolutional Neural Networks (CNNs) [164, 199, 245, 298]. For a more comprehensive examination of the different methods, the reader is referred to Sec. 8.2. Irrespective of the method used for people counting, the main concern is that the results are usually evaluated with respect to the whole testing images, allowing for possible error compensations and not performing an analysis at every scale, which could be useful for the simulation community interested also in local information in order to better characterize the crowd.

Other tasks that can be performed in a macroscopic setting are flow segmentation and abnormal motion pattern detection, where motion pattern refers to a set of dominant displacements observed in a crowded scene over a given time scale. In [6] the authors propose an approach to segment the crowd flow and possibly to detect instabilities based on Lagrangian particle dynamics and optical flow. Also [181] uses particle dynamics, learning normal behaviour in the scene using a bag of words to detect abnormal ones, without the need of any segmentation.

A different approach [142] avoids the use of optical flow-based motion description by extracting 3D spatio-temporal cuboids and computing spatio-temporal gradients of pixel intensities, represented using a 3D Gaussian Mixture Model (GMM). The authors model normal behaviour using a Hidden Markov Model (HMM) and label a new observation as abnormal if it does not fit the learned model. GMMs are employed as well in [230] to generate a model of normality, encoding optical flow information using a 3D Grey Level Co-occurrence Matrix (GLCM). Temporal analysis of GLCM-based texture measures has been employed recently in [168] for the detection of abnormal activities.

An interesting approach appears in [78], where the crowd is modeled as an evolving graph in time. The vertices of this graph correspond to a set of local features, which are spatially and temporally connected using Delaunay triangulation and sparse tracking. This compact representation preserves local information and bypasses any group segmentation step usually involved in microscopic methods. However, sparse feature extraction and the proposed triangulation method are not suited for high-density crowds where the proxemics between people is consistently reduced.

Abnormal motion pattern detection can be performed also to detect violent flows. The Violent Flow (ViF) method is proposed in [99] to identify dangerous crowd flows in densely populated areas using changes in optical flow magnitude. Criticisms about the inability of ViF to capture potentially important changes in orientation are underlined in [88], where the authors introduce a variant of the ViF descriptor that utilizes both orientation and magnitude of optical flow. A related but different work is the framework proposed in [248], which is able to correctly identify multiple crowd behaviors (bottlenecks, lanes, arches, and blocking) through stability analysis for dynamical systems.

Generally, all these methods assume normal behaviour by learning a representation of it, and find anomalies as evidences neglecting the learned baseline model. However, they show their limitations when significant overlap of motion patterns is present in the scene, or when there is a lack of consistency in the flow's characteristic.

It should be finally noted that works related to density estimation could be easily extended for abnormality detection in crowds, as in [283] where a Support Vector Machine (SVM) is trained on top of the people counting stage, in order to detect potential danger due to overcrowdedness.

1.1.3 Coupling crowd synthesis and analysis

Crowd synthesis and analysis are two different fields of study that assume the use of totally different techniques to reach their scopes. However, since both of them involve the study of the same physical entity - albeit from different perspectives - they could intrinsically benefit from each other. Over the years, there have been several works about crowd synthesis helping analysis and vice-versa, but open problems in both directions still remain.

An interesting example of crowd analysis exploiting in an original way concepts related to crowd synthesis is [181], which deals with abnormal crowd behaviour detection. It uses indeed the concept of Social Force Model, by modeling the crowd as a collection of interacting particles with associated attractive force (related to personal goal) and repulsive force (related to the psychological tendency of human beings to maintain a certain distance with respect to other pedestrians and the environment). In this way, the regions of anomalies in the abnormal frames are localized using interaction forces.

More practically, a common problem of crowd analysis techniques is the limited availability of labeled data, to perform both training and validation. The majority of algorithms rely indeed on a training stage, which requires the existence of a considerable amount of data to learn from. Again, labeled data are needed also to be able to perform a robust evaluation of the results. The task of labeling the data is tiresome, as much as being inevitable. In presence of high-density crowd images, the task is particularly time-consuming and prone to errors. For people tracking applications then individual trajectories should be also manually extracted. Besides this, there exists also the intrinsic problem of scarce availability of data which represent emergency situations (e.g. panic or violent episodes), which would be very interesting and useful to analyze. Visual evidences of such scenarios would be also unsafe to reproduce in a controlled scenario. To this extent, there exist some approaches which explore crowd simulations in order to obtain data to train or validate crowd analysis techniques. Video sequences of dangerous situations are generated in [8], such as blocked exits or individual collapses. Crowd simulation algorithms can be then used to generate ground-truth data [7, 167] for validation purposes. A complete dataset, the Agoraset Dataset [48], has been proposed to be used for evaluation of low-level video crowd analysis methods, such as tracking or segmentation. It has been designed to reflect classic crowd flow observed in real life situations, such as flow of humans in a free environment or in an environment with obstacles, evacuation through a door, and crossing flows.

The advantages of using simulated data for crowd analysis are the possible control over the simulation features, the possibility to generate an associated ground truth and to use the simulated data to bootstrap machine learning techniques. However, using crowd simulations to help analysis still presents some problems. Firstly, crowd simulators are based on mathematical models that reproduce the average behaviours of the individuals, being not able to simulate totally unpredictable behaviours. Secondly, as the main objective of computer vision analysis technique is to be used in real-world situations, the realism of the simulations is a major issue despite recent advances of computer graphics techniques. This is related to the modeling of realistic humans (e.g. different shapes and clothing), as well as realistic environments (e.g. surrounding structures, varying lightning conditions, camera perspective noise simulation and lens distortion), which complicates the model and increases the number of its parameters.

To this extent, crowd analysis can be used to help crowd synthesis. In order to obtain realistic scenarios, information from real world can be fed into a crowd simulator, as in [49] where an optical flow-based method is used to capture the movements of people, generating more realistic velocity fields over time. In [136] the authors propose a computer vision tool to provide useful information that could be helpful in the initial configuration of the particles of simulation models, providing support in the validation phase as well.

The concept of *data-driven* simulations has been investigated recently in [15], where the authors propose a framework to model both macroscopic and microscopic behaviours. Macroscopic modeling is done extracting and clustering pedestrian trajectories from real videos, in order to compute the velocity field associated with each exit region. Then, microscopic modeling exploits both the Social Force Model and the computed velocity fields. The authors use real data recorded from a drone, where the particular top-view position of the camera allows for an accurate mapping of the environment as well as an accurate projection of the velocity fields to the ground plane. The applicability of the method is then limited to low density crowds. In general, defining quantitative metrics to evaluate the realism of a crowd simulation is still an open question, and needs the availability of real data to validate the simulated one.

Posterior analysis of specific crowded scenes can then help in the studies of mass gatherings to ensure the security of future events. Many empirical studies have been performed during the last years. Some of them exploit private video sequences of real episodes of stampedes. In [103] for instance, the authors analyzed real video sequences from the 2006 stampede that happened in Mina, revealing two subsequent, sudden transitions from laminar (i.e. ordered) to stop-and-go and turbulent flows, which arise questions about the goodness of many previous simulation models in presence of overcrowded unidirectional flows. However, this type of data representing real emergency situations rarely exists and is generally difficult to obtain.

Other works assume an a-priori scene equipment with specific sensors, like [296] in which active infrared counters have been employed to count the number of people by detecting the retro-diffused light from each individual. The study, by analysing the recorded data during the 2011 Chinese lantern festival along with sensor information, has been effective in quantitatively investigating the properties of unidirectional flow in a crowded large street, estimating the capacity of the street (which is indeed closely related to the maximum flow rate) that resulted to be higher with respect to the recommended value of guidelines and specifications of facilities design in architecture. This allowed to better exploit the available spaces in the subsequent editions of the event. The work is also interesting because it stresses the differences between unidirectional large flows and bidirectional unstructured flows, and the necessity of applying totally different models for the two situations. However, it is not easy for every large scale event to find good open spaces where to install the needed equipment, and there are (international) regulations to follow as well as privacy concerning issues that increase the difficulty in obtaining legal permissions of recordings and sensor placing.

To this extent, other works exploit only publicly available sources, e.g. investigation reports by public authorities and media, maps from Google Earth, 360° photographs, YouTube videos, documents released by Wikipedia and other sources. An example is the posterior study of the Duisburg's Love Parade disaster [105], which helped in the definition of a common scale of situation criticality levels as long as the measures to face each stage, from access control to crowd evacuation and emergency first aid disposal. The problem of this approach however resides in the fragmentation of information and in the lack of synchronization and possibly agreement across the various sources. The insight of these studies led to organizational changes that helped in the organization of safer future editions of the events, pinpointing the necessity of posterior studies tailored to the specific situations. To this extent, density estimation analysis at every scale will be helpful in the definition of realistic models for unidirectional flows.

1.2 The Dataset

A major challenge in crowd analysis is the scarce availability of images or video sequences, which can be used either for training or validation purposes. Usually, large datasets for pedestrian detection and tracking in crowds using video sequence, like PETS2009 [70, 75] or EWAP [204] do not deal with very high-density crowds but are rather adapted for applications where all the body of the people is visible. To face occlusion problems, they can be composed of several synchronized camera views placed in different locations, like PETS2009, or they can exploit almost vertical single cameras, like EWAP. Besides, they usually focus on indoor detection, like the Mall Dataset [34], where cameras are easy to install and there is an adequate level of environmental lightning. These types of dataset are appropriate for applications such as person re-identification [91], i.e. when the same person has to be repeatedly recognized by several cameras in order to track his or her path, or behavior analysis and study of group dynamics and collision avoidance mechanisms. In particular, the Grand Central Dataset [304] provides KLT keypoint trajectories extracted from the video, and is well suited for understanding and learning crowd behaviors, although the crowd is not very dense.

Conversely, high-density crowd large datasets are well adapted for the task of density estimation but they do not suit our needs for two reasons. Some of them, like UCF-CC-50 [119] and the

Figure 1.1: Example images coming from dataset composed by temporal sequences of images/frames, for the task of pedestrian detection and tracking in low to medium dense crowds. (a), (b) Different views of PETS2009 dataset; (c) Grand Central dataset; (d) Mall dataset.

more recent ShanghaiTech Dataset [293, 298] are composed by still images coming from different scenes. This allows the learning algorithm to be more robust to scene variations, but limits the applicability of the data just to a small subset of applications related to people counting, with no possibility of an extension to people tracking. On the other hand, some other datasets deal with video sequences of large crowds, but do not have labeled ground-truth with precise localization coordinates for all the individuals, providing instead just the total number of people entering and leaving the scene per frame, like PCDS [250], or the bounding box information of the salient regions, like the Crowd Dataset [165]. In the same way, the very recent Crowd-11 dataset [69] provides over 6000 video sequences of crowds, but the ground-truth is expressed in terms of brief video-level based annotations describing the crowd rather than the individuals (it has indeed been proposed for the different task of behavior understanding).

In the very recent past, thanks to the advances in computer graphics and related hardware, always more simulated datasets have been proposed. For example, the authors of [235] introduced a new optical flow dataset exploiting the possibilities of recent video engines to generate sequences with ground-truth optical flow for large crowds in different scenarios. However, their ground-truth is expressed in terms of optical flow, and not precise trajectories of the individuals. On the contrary, the Agoraset dataset [48] provides simulated video sequences of high-density crowds along with the positions of all the pedestrians, but the main problem is the lack in photo-realism of the scene, so that the detection task loses its interest. Alternatively, JTA [72] dataset is a recently proposed huge dataset for pedestrian pose estimation and tracking in urban scenarios created by exploiting the highly photo-realistic video game Grand Theft Auto V developed by Rockstar North. It includes 512 full-HD videos of 30 seconds along with 3D annotations for each frame. It presents impressive realism of the scenes, however high-density scenarios are rare due to the specific game

Figure 1.2: Example images coming from dataset of heterogeneous images of high-density crowds. (a), (b) Different images of UCF_CC_50 dataset; (c), (d) Different images of ShanghaiTech dataset.

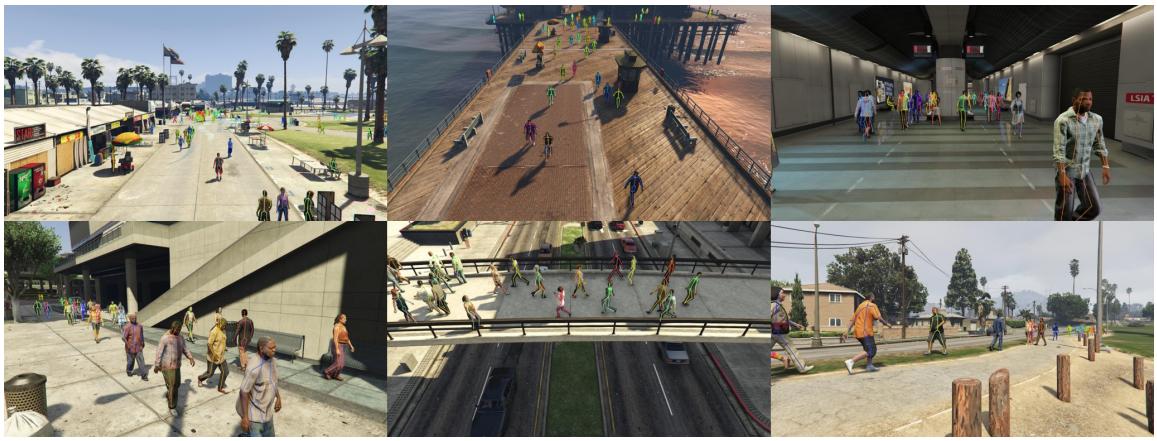


Figure 1.3: Example images from the JTA dataset.

environment (see Fig. 1.3).

The dataset used in this project is composed by gray-scale images acquired at Makkah during very congested times of the Hajj period, in October 2012. It has been partially manually labeled with head's center coordinates as explained in Appendix A, in order to have different training, validation and testing images. As shown in Fig. 1.4, the pictures in this dataset represent very highly crowded scenes in which the people to be detected are really small and not entirely visible, i.e. very vulnerable to occlusions, and difficult to detect because of the absence of static background. Moreover, due to the dynamics of the scene, people are often side-viewed or back-viewed, diminishing the visual appearance available details. Thanks to the camera pitch angle, the human body

Figure 1.4: Example image from the Makkah dataset (considered in the context of this work).

parts which are mostly visible are the heads, that will thus become our targets for the detection task. Both for privacy reasons and implicit camera recording limits however, facial traits of the people composing the crowd are not delineated, another factor that increases the difficulty of the scene's analysis. Nevertheless, a classifier can be efficiently trained to perform head-shoulder detection, that is characterized by a distinctive omega-like shape in almost all view angles [161]. As depicted in Fig. 1.5 however, our setting is definitely more difficult than the usual head-shoulder detection problem and needs wider processing and discussion.

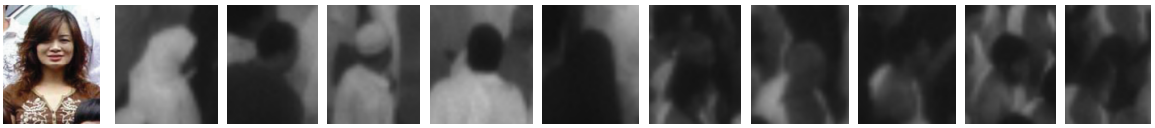


Figure 1.5: Comparison between a typical head used to perform head-shoulder detection (first image taken from [159], RGB and rich in texture and information) with respect to examples of heads from our Makkah dataset (grey-scale images with high clutter and frequent occlusions).

Chapter 2

Supervised learning and classifier combination

Contents

2.1 Introduction

2.1.1 The role of machine learning in computer vision

The objective of computer vision is to implement systems with human-like perception capabilities. Over the years, the field has evolved from pattern recognition and image processing applications to advanced methods of image understanding and knowledge-based vision. There has been an increasing demand to address *real-world* problems, requiring algorithms which are able to work under partial occlusion, high clutter, low contrast and changing environmental conditions. To face these challenges, computer vision techniques must be robust and simultaneously flexible with respect to the variety of possible given tasks.

At the same time, the field of machine learning uses statistical techniques to give computer systems the ability to "learn from data", i.e. progressively improve performance on a specific task. With the recent advances in hardware and software, machine learning techniques can be exploited by more and more practical applications. To this extent, computer vision provides interesting and challenging problems to drive advances in the machine learning field. Machine learning technology has indeed a strong potential to contribute to the development of flexible learning-based vision systems, with great generalization ability.

2.1.2 Learning models

Machine learning includes several types of techniques, which can be divided into different groups on the basis of specific learning models:

- *Supervised Learning* – In supervised learning the algorithm is given a set of samples along with their actual labels, and it learns a mapping function from the samples to the set of possible outputs, which can be used later to classify unseen test examples. The labeling of the training data is usually done by an external mechanism (e.g. humans), hence the name "supervised";

- *Unsupervised Learning* – On the contrary, in unsupervised learning the samples do not come along with their label. Unsupervised learning tries to find regularities in the unlabeled training data (such as different clusters under some metric space), infer the class labels and under certain circumstances even the number of classes. The most common unsupervised learning technique is clustering;

- *Reinforcement Learning* – Reinforcement learning refers to goal-oriented algorithms, which learn how to attain a complex objective. The algorithm starts from a blank state, and learns to take decisions (or actions) given its current state by maximizing a quantifiable *reward* signal. There is no concept of labeled/unlabeled data involved, rather the learning process is done by iteratively trying sequences of actions and getting rewards until the integrated reward signal is maximized;

- *Semi-supervised Learning* – Semi-supervised learning falls between supervised and unsupervised learning. The algorithms in this category use a small amount of labeled samples in conjunction with a large amount of unlabeled data, and are particularly useful in applications where it is difficult to obtain the labels of a large quantity of data, since it often requires skilled human agent intervention. The cost associated with the labeling process may thus make the building of a fully labeled training set infeasible, whereas the acquisition of unlabeled data remains relatively inexpensive. A simple method to perform semi-supervised learning is *pseudo-learning*, where a classifier is trained on a small amount of labeled data, tested on the unlabeled one and re-trained on the whole dataset with the obtained estimated labels. An alternative is co-training [18], that requires at least two views of the data. It first learns a separate classifier for each view using any labeled examples, then the most confident predictions of each classifier on the unlabeled data are used to iteratively construct additional labeled training data;

- *Active Learning* – Active learning is similar to semi-supervised learning in that it uses both labeled and unlabeled data, even though it is based on a completely different assumption. Active learning allows indeed the algorithm to choose which training samples have to be added to the training set, in order to enhance the performance of the classifier. The algorithm starts with a small amount of labeled data, then the current model is tested on the unlabeled data, and finally samples for which the model is the least certain are selected for queries about their true label to an oracle. The main difference with respect to semi-supervised learning resides thus in the fact that, once in the training set, a given sample selected with active learning is surely associated with its corrected label (assigned by the oracle), but this is not necessarily true in semi-supervised learning. Nevertheless, there is the need for an expert who annotates the samples selected by the algorithm.

In this work, we will focus our attention on supervised and active learning. In this Chapter, we will summarize some of the most common supervised learning algorithms and the possible ways of combining them. Later, in Chapter 5, we will go deeper into active learning, explaining its necessity for our particular application.

2.2 Supervised learning

Supervised learning is the task of learning a function that maps an input to an output based on example input-output pairs. When the possible outputs are categories, we can speak of a *classification* task. On the contrary, when the outputs live in a continuum space of values, we can speak of a *regression* task. The boundary between these two tasks is however quite soft; for instance, probability-predicting regression models can be used as part of a classifier by imposing a decision rule (this is often done with logistic regression). Classifiers can be then categorized into *probabilistic*, which produce probabilistic scored outputs, and *non-probabilistic*, which are only able to discriminate between the different classes (although maybe providing non-calibrated scores). However, the majority of classification algorithms have been extended to produce predictive conditional probabilities of class labels given the input, even though they were not originally built on any probabilistic framework. In other cases, such as SVM, the classifier has not been extended to probabilistic framework, but it is still possible to get probability values out of the uncalibrated classification scores by applying methods such as logistic regression.

2.2.1 Bias–variance trade-off

Whatever the algorithm chosen for the specific task, ideally we would like to obtain a model that accurately captures the regularities of the training data while at the same time generalizing well to unseen data. The bias–variance dilemma is thus the conflict in trying to simultaneously reach these two goals, i.e. simultaneously minimize two different sources of error:

- *Bias* – An error due to limited flexibility of the algorithm to learn the true model from the training dataset. High bias can cause an algorithm to fail to capture the structure exhibited by the data (*underfitting*). The solution is usually to increase the complexity of the model;
- *Variance* – An error due to sensitivity to small fluctuations in the training set. High variance can cause an algorithm to model noisy or unrepresentative training data, failing to generalize well during testing (*overfitting*). A first reason behind this could be the lack of sufficient amount of training data. To mitigate overfitting then it is possible to apply *regularization* techniques, that consist in penalizing some parameter values in order to control the flexibility and generalization ability of the model.

2.2.2 Notation

Before introducing some supervised learning algorithms which are used in this work, let us establish the notation for the following. We will denote the *input variables*, i.e. *input features*, as $\mathbf{x}$, and the *output variables*, i.e. *target variables* or *labels* as y. A pair $\left(\mathbf{x}^{(i)}, y^{(i)}\right)$ represents the i^{th} *training sample* (or *example*), so that the entire set of training samples we will use to learn from – a list of m training samples $\left\{\left(\mathbf{x}^{(i)}, y^{(i)}\right) : i \in \{1, \ldots, m\}\right\}$ – is called *training set*. Each input feature $\mathbf{x}^{(i)}$ is represented by a vector living in $\mathbb{R}^d$, i.e. a d-dimensional vector, while the corresponding $y^{(i)} \in \Omega$, where Ω is the set of possible labels (e.g., for binary classification $\Omega = \{0, 1\}$). The m x d matrix containing all the feature vectors is called *design matrix*.

Denoting $\mathcal{X}$ the space of input values, and $\mathcal{Y}$ the space of output values, the goal of supervised learning is, given a training set, to learn a function $h : \mathcal{X} \mapsto \mathcal{Y}$ so that $h(x)$ is a "good" predictor for the corresponding value of y. The function $h(\cdot)$ is called *hypothesis function*, and represents the learned *model* that we will apply to unseen testing samples at inference time. Note that $\mathcal{Y}$ is different from Ω, since the former represents the space of possible real output values, while the latter represents the discrete set of possible labels to which the output can belong to.

We denote as $\mathscr{D}_{train}$ the portion of samples from the dataset used for training, as $\mathscr{D}_{calib}$ the portion of samples that are used to validate the model's parameters, and as $\mathscr{D}_{test}$ the portion of samples used to finally test the algorithm's behavior in presence of unseen data in order to evaluate it.

2.2.3 Logistic regression

Logistic regression is a probabilistic discriminative model, since it learns predictive probabilities of class labels given the examples only. It relies on the *sigmoid* or *logistic function* σ, which maps real-valued numbers into the unit interval. Denoting by θ_i the *parameters* (also called *weights*) of the model, logistic regression maps a linear combination of the input entries along with their weights into a non linear output which is the predictive probability.

Therefore, the hypothesis function will be of the form:

$$h_\theta(\mathbf{x}) = \sigma(\theta^\intercal \cdot \mathbf{x}) = \frac{1}{1+e^{-\theta^\intercal \cdot \mathbf{x}}}, \tag{2.1}$$

where $\theta^\intercal \cdot \mathbf{x} = \theta_0 + \sum_{j=1}^{d} \theta_j \cdot \mathbf{x}_j$ following the convention of letting $x_0 = 1$, and

$$\sigma(z) = \frac{1}{1+e^{-z}}, \tag{2.2}$$

is called *logistic function* or *sigmoid function*. Note that $\sigma(z)$, and hence $h_\theta(\mathbf{x})$, is always bounded between 0 and 1: $\sigma(z)$ tends towards 1 as $z \to +\infty$, and tends towards 0 as $z \to -\infty$.

The set of parameters is learned from the training data $\mathscr{D}_{train}$ using maximum likelihood estimation.

Considering a binary classification problem, let us assume that:

$$\begin{aligned} \mathrm{P}(y=1|\mathbf{x};\theta) &= h_\theta(\mathbf{x}), \\ \mathrm{P}(y=0|\mathbf{x};\theta) &= 1 - h_\theta(\mathbf{x}). \end{aligned} \tag{2.3}$$

Equations (2.3) can be written in a more compact way as:

$$\mathrm{P}(y|\mathbf{x};\theta) = (h_\theta(\mathbf{x}))^y \, (1-h_\theta(\mathbf{x}))^{1-y}. \tag{2.4}$$

Therefore, assuming m independent training examples, we can derive the likelihood of the parameters to be maximized as:

$$\mathrm{L}(\theta) = \prod_{i=0}^{m} \mathrm{P}(y^{(i)}|\mathbf{x}^{(i)};\theta), \tag{2.5}$$

$$= \prod_{i=0}^{m} \left(h_\theta(\mathbf{x}^{(i)})\right)^{y^{(i)}} \left(1-h_\theta(\mathbf{x}^{(i)})\right)^{1-y^{(i)}}, \tag{2.6}$$

which is equivalent to maximize the log-likelihood $l(\theta) = \log \mathrm{L}(\theta)$ in order to avoid numerical problems:

$$l(\theta) = \log \mathrm{L}(\theta), \tag{2.7}$$

$$= \sum_{i=0}^{m} y^{(i)} \log h_\theta(\mathbf{x}^{(i)}) + \left(1-y^{(i)}\right) \log\left(1-h_\theta(\mathbf{x}^{(i)})\right). \tag{2.8}$$

To solve this, gradient descent algorithm can be employed for the minimization of the negative log-likelihood. The update of the parameters can be done once for all the training samples, or sequentially using mini-batches of training data.

Note that a generalization for the multi-class classification problem exists, and takes the name of Softmax regression (or Multinomial logistic regression), under the assumption that the classes are mutually exclusive.

2.2.4 Support Vector Machines

Support Vector Machines [21] (SVMs) are non-probabilistic models used for supervised binary classification. They have been originally designed to perform linear classification, but they have been extended to deal with the non-linear case by means of the *kernel trick*.

The main task of an SVM consists in predicting whether a test sample belongs to one of two classes. For convenience, let us assume labels $y^{(i)} \in \Omega = \{-1, +1\}$ and features $\mathbf{x}^{(i)} \in \mathbb{R}^d$.

2.2.4.1 Optimal margin classifier

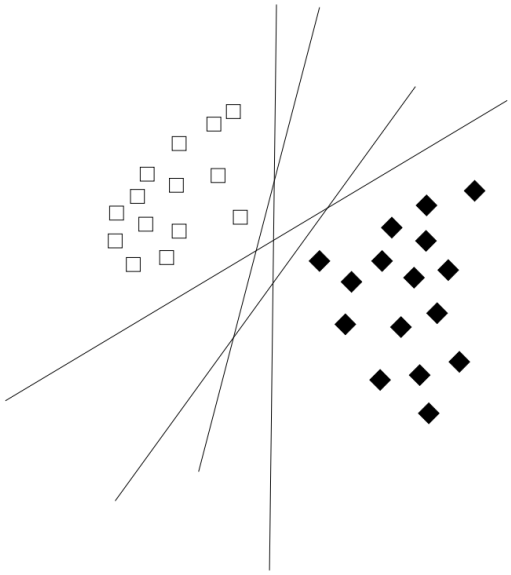


Figure 2.1: Examples of several possible hyper-planes which solve the linearly separable classification problem on the given training data, for the bi-dimensional case.

Considering the case of linearly separable data, we could find a hyper-plane splitting the inputs such that samples of the same class would lay in the same region of the input space. To this extent, a hyper-plane $\mathcal{P}$ can be defined as:

$$\mathcal{P}: \quad \mathbf{w}^\intercal \cdot \mathbf{x} + b = 0, \tag{2.9}$$

where $\mathbf{w}$ is the normal vector to $\mathcal{P}$ and b is the bias term which represents the intercept, so that $\frac{|b|}{\|\mathbf{w}\|}$ is the distance from the hyper-plane to the origin. The problem is solved by assigning:

$$y^{(i)} = \begin{cases} -1, & \text{if } \mathbf{w}^\intercal \cdot \mathbf{x}^{(i)} + b < 0, \\ +1, & \text{if } \mathbf{w}^\intercal \cdot \mathbf{x}^{(i)} + b \geq 0. \end{cases} \tag{2.10}$$

However, without any further constraints, there is an infinite number of solutions to this problem. Figure 2.1 shows several possible hyper-planes for the bi-dimensional case. These hyper-planes solve the separation problem for the training data, but may have different performance on the unseen test samples. Among all the possible solutions, we should choose the one which provides the best generalization ability. Intuitively, this corresponds to choosing the separation plane to be as far as possible from samples of both classes. The SVM learning function is thus computed by maximizing the distance, i.e. the *margin*, between the hyper-plane and the closest training input vector(s) for each label, as depicted in Fig. 2.2.

To this extent we now define two specific hyper-planes, representing the planes that cut through the closest training examples on either side. We can call them *support hyper-planes*, because they are defined from the feature vectors that do indeed support the planes, i.e. the *support vectors*:

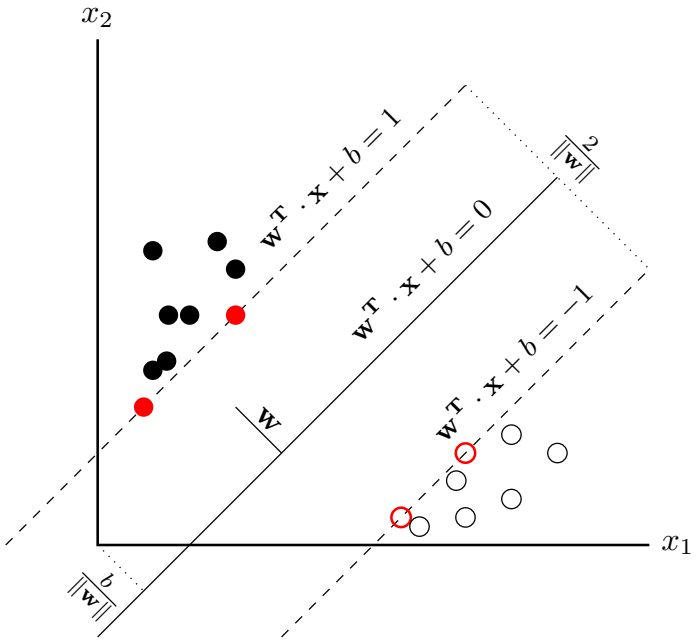


Figure 2.2: Illustration of a linear SVM for binary classification problem in the bi-dimensional space.

$$\begin{aligned} \mathcal{P}_1: &\quad \mathbf{w}^\intercal \cdot \mathbf{x} + b = +1, \\ \mathcal{P}_2: &\quad \mathbf{w}^\intercal \cdot \mathbf{x} + b = -1, \end{aligned} \tag{2.11}$$

In this way, points on or above $\mathcal{P}_1$ are assigned to $y = +1$, while points on or below $\mathcal{P}_2$ are assigned to $y = -1$. From this, we can derive two conditions to be respected:

$$\begin{cases} \mathbf{w}^\intercal \cdot \mathbf{x}^{(i)} + b \geq +1, & \text{if} \quad y^{(i)} = +1, \\ \mathbf{w}^\intercal \cdot \mathbf{x}^{(i)} + b \leq -1, & \text{if} \quad y^{(i)} = -1. \end{cases} \tag{2.12}$$

We can write these constraints in a more compact way as:

$$y^{(i)} \left(\mathbf{w}^\intercal \cdot \mathbf{x}^{(i)} + b \right) - 1 \geq 0. \tag{2.13}$$

Geometrically, regardless of the value of b, the distance between the two hyper-planes $\mathcal{P}_1$ and $\mathcal{P}_2$, i.e. the margin we want to maximize, is equal to $\frac{2}{\|\mathbf{w}\|}$. This corresponds to the following convex quadratic minimization problem in its primal form:

$$\begin{aligned} \underset{\mathbf{w},b}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & y^{(i)} \left(\mathbf{w}^\intercal \cdot \mathbf{x}^{(i)} + b \right) \geq 1, \quad i = 1, \ldots, m. \end{aligned} \tag{2.14}$$

The solution to this problem gives the *optimal margin classifier*. Let us now talk about its dual form, which is important for two reasons. Firstly, it allows us to use kernels to be able to work in very high dimensional spaces. Secondly, it allows us to derive an efficient algorithm for solving the optimization problem, namely the Sequential Minimal Optimization (SMO) [210].

2.2.4.2 Lagrangian dual form

The optimization problem reported in Eq. (2.14) can be rewritten in its Lagrangian dual form. To do so, let us build the Lagrangian for the specific problem:

$$\mathscr{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i \left[y^{(i)} \left(\mathbf{w}^\intercal \cdot \mathbf{x}^{(i)} + b \right) - 1 \right]. \tag{2.15}$$

where α_i are the Lagrange multipliers, one for each constraint of the primal form. The Lagrangian primal problem is:

$$\begin{aligned} &\min_{\mathbf{w},b} \max_{\alpha} \quad \mathscr{L}(\mathbf{w}, b, \alpha) \\ &\text{subject to} \quad \alpha_i \geq 0, \quad i = 1, \ldots, m. \end{aligned} \tag{2.16}$$

Solving this problem involves setting to zero partial derivatives of $\mathscr{L}$ with respect to $\mathbf{w}$ and b:

$$\nabla_{\mathbf{w}} \mathscr{L}(\mathbf{w}, b, \alpha) = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y^{(i)} \mathbf{x}^{(i)} = 0, \tag{2.17}$$

$$\frac{\partial \mathscr{L}(\mathbf{w}, b, \alpha)}{\partial b} = -\sum_{i=1}^{m} \alpha_i y^{(i)} = 0. \tag{2.18}$$

Equation (2.17) implies that

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y^{(i)} \mathbf{x}^{(i)}. \tag{2.19}$$

Substituting Eqs. (2.18) and (2.19) in the Lagrangian formulation of Eq. (2.15) we obtain the *Wolfe dual Lagrangian function*:

$$\mathrm{W}(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} \left(\mathbf{x}^{(i)} \right)^\intercal \mathbf{x}^{(j)}. \tag{2.20}$$

The optimization problem is now called *Wolfe dual problem*:

$$\begin{aligned} &\underset{\alpha}{\text{maximize}} \quad \mathrm{W}(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} \langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle \\ &\text{subject to} \quad \alpha_i \geq 0, \quad i = 1, \ldots, m \\ &\qquad\qquad \sum_{i=1}^{m} \alpha_i y^{(i)} = 0. \end{aligned} \tag{2.21}$$

The main advantage of the Wolfe dual problem over the Lagrangian problem is that the objective function W now depends only on the Lagrange multipliers.

When we solve the Wolfe dual problem, we obtain a vector containing all the α_i Lagrange multipliers. However, our main goal was to find $\mathbf{w}$ and b. From Eq. (2.19) we can easily derive $\mathbf{w}$. With this, considering the constraints of the primal problem, we can derive b by taking the average of the nearest positive support vector and the nearest negative support vector:

$$b = -\frac{\max_{y^{(i)}=-1}(\mathbf{w}^\intercal \mathbf{x}^{(i)}) + \min_{y^{(i)}=+1}(\mathbf{w}^\intercal \mathbf{x}^{(i)})}{2}. \tag{2.22}$$

Finally, let us suppose to have solved the optimization problem and found the optimal value of $\mathbf{w}$ in terms of α_is. Now, if we want to make a prediction for an unseen input $\mathbf{x}_{\text{test}}$, we would use the following hypothesis function:

$$h(\mathbf{x}_{\text{test}}) = \text{sign}(\mathbf{w}^\intercal \mathbf{x}_{\text{test}} + b). \tag{2.23}$$

Using Eq. (2.19), this can be rewritten as

$$h(\mathbf{x}_{\text{test}}) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i y^{(i)} \langle \mathbf{x}^{(i)}, \mathbf{x}_{\text{test}} \rangle + b \right). \tag{2.24}$$

Note that α_i will always be zero except for the support vectors. Hence, in order to make a prediction, we have to calculate a quantity that depends only on the inner products between the test sample and the support vectors (which are usually far less than the total number of training samples). It is worth noting also that the dual form requires only the dot product of each input vectors to be calculated, and this fact will be a key point for the application of the kernel trick described in the following.

2.2.4.3 Kernels

Let us now consider the case of non-linearly separable data. Often there are non-linear patterns in the data, and a linear classifier is not enough. The traditional way of transforming a linear classifier in a non-linear one is by mapping the data to higher dimensions, from the input space $\mathcal{X}$ to a feature space $\mathcal{F}$ using a non-linear function $\phi : \mathcal{X} \rightarrow \mathcal{F}$, where we hope that the transformed data will be linearly separable. However, this approach does not scale well with the number of features and in general the mapping can be expensive to compute.

Kernel methods solve this problem by avoiding the step of explicitly mapping the data to a high dimensional feature space, by means of the *kernel trick*, that provides a bridge between linearity and non-linearity for every problem that can be expressed in terms of dot products between two vectors. If an algorithm is described solely in terms of inner products in the input space indeed, it can be lifted into a feature space by replacing occurrences of those inner products by a kernel function.

Specifically, a kernel function k implicitly performs a mapping ϕ to a higher dimensional vector space in which a dot product is defined (Hilbert space):

$$k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle. \tag{2.25}$$

The kernel $k(\mathbf{x}, \mathbf{x}')$ takes two inputs in the $\mathcal{X}$ input space and gives their *similarity* in the $\mathcal{F}$ feature space:

$$\phi : \mathcal{X} \rightarrow \mathcal{F}, \quad k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}. \tag{2.26}$$

Now, since SVM optimization problem in its Wolfe dual form is indeed described only in terms of dot products (cf. Eq. (2.21)), we can generalize it to use kernel functions:

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \\ \text{subject to} \quad & \alpha_i \geq 0, \quad i = 1, \ldots, m \\ & \sum_{i=1}^{m} \alpha_i y^{(i)} = 0. \end{aligned} \tag{2.27}$$

This translates into the following optimization problem in the primal form (cf. Eq. (2.14)):

$$\begin{aligned} \underset{\mathbf{w}, b}{\text{minimize}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{subject to} \quad & y^{(i)} \left(\langle \mathbf{w}, \phi(\mathbf{x}^{(i)}) \rangle + b \right) \geq 1, \quad i = 1, \ldots, m. \end{aligned} \tag{2.28}$$

In order to make a prediction for an unseen input $\mathbf{x}_{\text{test}}$ (cf. Eq. (2.24)), we have now to compute the similarity between $\mathbf{x}_{\text{test}}$ and the support vectors, computed through the kernel function:

$$h(\mathbf{x}_{\text{test}}) = \text{sign} \left(\sum_{i \in \text{SV}} \alpha_i y^{(i)} k(\mathbf{x}^{(i)}, \mathbf{x}_{\text{test}}) + b \right), \tag{2.29}$$

with SV being the set of indexes of training samples corresponding to non-zero α_i.

Note that not all functions can be employed as kernels. Given the training set composed by m training samples $x^{(1)}, \ldots, x^{(m)}$, the *Kernel Matrix* $\mathbf{K}$, i.e. the *Gram Matrix*, is defined such that

Table 2.1: Popular kernel functions between two input vectors $\mathbf{x}$ and $\mathbf{x}'$.

Kernel	**Formulation**
Linear	$k(\mathbf{x},\mathbf{x}') = a\mathbf{x}^\top\mathbf{x}' + c$
Polynomial	$k(\mathbf{x},\mathbf{x}') = (a\mathbf{x}^\top\mathbf{x}' + c)^d$
Radial Basis Function (RBF)	$k(\mathbf{x},\mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$
Hyperbolic Tangent	$k(\mathbf{x},\mathbf{x}') = \tanh(a\mathbf{x}^\top\mathbf{x}' + c)$
Power	$k(\mathbf{x},\mathbf{x}') = -\|\mathbf{x}-\mathbf{x}'\|^d$
Log	$k(\mathbf{x},\mathbf{x}') = -\log(\|\mathbf{x}-\mathbf{x}'\|^d + 1)$
Chi-Square (χ^2)	$k(\mathbf{x},\mathbf{x}') = \sum_{j=1}^{n} \frac{2x_j x'_j}{x_j + x'_j}, \quad x_j, x'_j \geq 0$
Histogram Intersection Kernel (HIK)	$k(\mathbf{x},\mathbf{x}') = \sum_{j=1}^{n} \min(x_j, x'_j), \quad x_j, x'_j \geq 0$

$$\mathbf{K}(i,j) = k(\mathbf{x}^{(i)},\mathbf{x}^{(j)}). \tag{2.30}$$

In order to be a valid kernel, i.e. a kernel which correctly performs a feature mapping ϕ : $k(\mathbf{x},\mathbf{x}') = \langle\phi(\mathbf{x}),\phi(\mathbf{x}')\rangle, \forall\mathbf{x},\mathbf{x}'$, some conditions must be satisfied. Firstly, the Kernel Matrix $\mathbf{K}$ has to be symmetric, i.e. $\mathbf{K}(i,j) = k(\mathbf{x}^{(i)},\mathbf{x}^{(j)}) = k(\mathbf{x}^{(j)},\mathbf{x}^{(i)}) = \mathbf{K}(j,i)$. Secondly, it has to be positive semi-definite, in accordance with the *Mercer's theorem* [123]. This ensures convexity of the optimization problem and uniqueness of the solution.

Choosing the most appropriate kernel and fine-tuning its possible parameters highly depends on the specific problem. Automatic kernel selection is possible although being not straightforward, as described in [111]. Kernel functions can be even learned, as in [185]. Table 2.1 reports some typical choices of kernel functions along with their formulation. Almost all the kernels require the setting of one or more parameters, e.g. a and c regulates slope and intercept respectively, σ regulates the width of the RBF kernel, d is the degree of polynomial used. While the majority of kernels are positive semi-definite, some of them are only conditionally positive yet performing well (Hyperbolic tangent, Power and Log kernels) [22].

2.2.4.4 Regularization

While mapping the data to a higher dimensional space generally increases the likelihood to be able to find a linear separation, one cannot guarantee it. In some cases, the SVM optimization problem outlined in Eq. (2.28) does not converge, since it is not possible to satisfy its constraints.

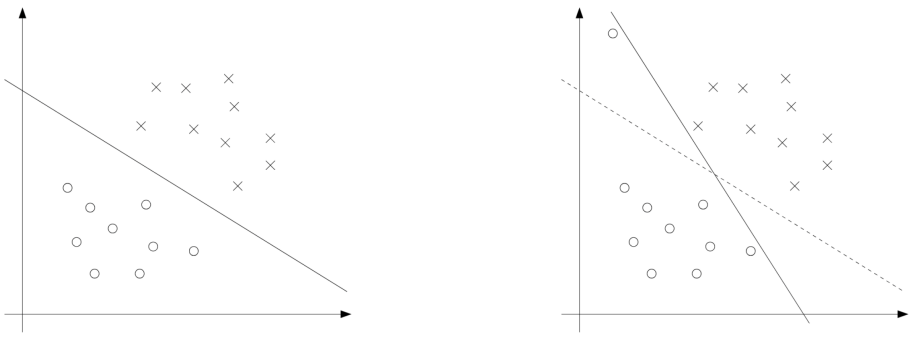


Figure 2.3: Optimal margin classifier changes in presence of a single outlier.

Moreover, even assuming that the nature of the problem is purely linear or may become linear in a high dimensional space, it is necessary to take into account that training samples are often

a result of noisy observations of real world data, yielding the training set to be often populated by misclassified, ambiguous or outlier samples. Figure 2.3 shows the impact of a single outlier point (added in the upper-left region in the right figure), which causes the decision boundary to dramatically change the slope, resulting in a classifier which much smaller margin. We would like thus to allow some points to be misclassified in order to obtain a larger margin which would result in a more robust classifier, less sensitive to outliers present in the training data.

To this extent, we reformulate the primal optimization problem of Eq. (2.28) introducing a L1 regularization term as:

$$\begin{aligned} \underset{\mathbf{w},b,\xi_i}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \mathrm{C}\sum_{i=1}^{m}\xi_i \\ \text{subject to} \quad & y^{(i)}\left(\langle \mathbf{w}, \phi(\mathbf{x}^{(i)})\rangle + b\right) \geq 1-\xi_i, \quad i=1,\ldots,m \\ & \xi_i \geq 0, \quad i=1,\ldots,m. \end{aligned} \tag{2.31}$$

This corresponds to the *Soft-Margin* SVM problem formulation proposed in [47]. The ξ_i variables are the *slack variables* that allow an example to be within the margin ($0 \leq \xi_i \leq 1$) or to be misclassified ($\xi_i > 1$). Note that $\sum_{i=1}^{m}\xi_i$ is an upper bound on the number of misclassified examples, since an example is misclassified if the value of its slack variable is greater than one. The parameter $\mathrm{C} > 0$ controls the relative importance of simultaneously maximizing the margin and minimizing the amount of misclassified examples. Small values of C allow for the presence of more misclassified examples, obtaining a model which generalizes better but with the risk of running into high-bias problem, i.e. underfitting. On the other hand, increasing the value of C, we increase the penalty assigned to misclassified examples, therefore the resulting margin will be smaller and we increase the risk of high-variance, i.e. overfitting. The optimal value of this parameter depends on the application and the considered data, and can be found using cross-validation techniques.

Accordingly, the Wolfe dual problem of Eq. (2.27) becomes:

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \mathrm{W}(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y^{(i)}y^{(j)}k(\mathbf{x}^{(i)},\mathbf{x}^{(j)}) \\ \text{subject to} \quad & 0 \leq \alpha_i \leq \mathrm{C}, \quad i=1,\ldots,m \\ & \sum_{i=1}^{m}\alpha_i y^{(i)} = 0. \end{aligned} \tag{2.32}$$

Note that the only change made by the addition of the regularization term is in the constraints relative to the admissible values of α_i, which now are upper-bounded by C. The problem continues to be only dependent on the inner products, so that it continues to be easily kernelized. Moreover, we can continue to use Eq. (2.29) to make predictions about unseen test samples, even though the support vectors will now include the points within the margin and the misclassified ones, besides the examples exactly on the margin as before.

2.2.5 Neural networks

With the term "Neural Networks" (NNs) we refer to a large family of machine learning algorithms. They have originally been inspired by the biological neural systems (hence the name) which is composed by billions of neurons connected through synapses. For a given neuron, the dendrite receives signals from other neurons and then the cell body sums all the incoming signals. When the sum reaches a threshold value, the neuron fires and the signal travels down the axon to the other neurons. The amount of signal transmitted depends on the strength (synaptic weights) of the connections.

In the same way, the basic building block of a neural network is a *neuron*, which is a computational unit. Neurons are interconnected among each others and the network can be trained to learn the synaptic strengths (i.e. the *weights w*) that control the strength of influence of one neuron on another. Then, the firing rate of the biological neuron is modeled through an *activation*

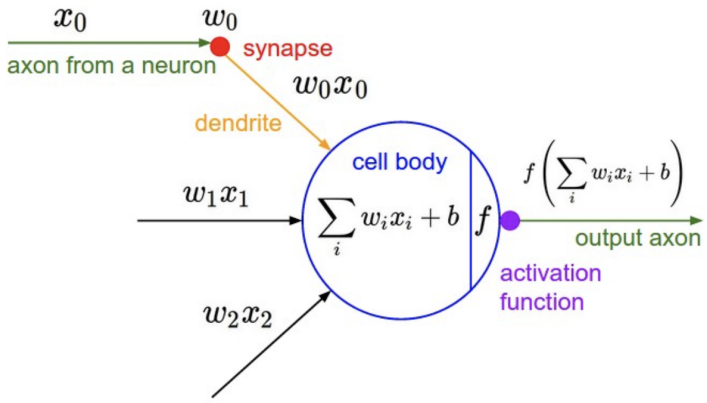


Figure 2.4: Mathematical model of a biological neuron.

function f, which is usually non-linear. Like the human brain, a neural network acquires knowledge through learning. Figure 2.4 shows this mathematical modelization of a biological neuron. Each neural unit, i.e. *perceptron*, produces a single output value based on several inputs coming from the output of other neural units. The generic expression of a neural unit output is thus:

$$y = f\left(\mathbf{w}^\mathsf{T}\mathbf{x}\right) \tag{2.33}$$

where f is the activation function, $\mathbf{w}$ denotes the vector of weights, $\mathbf{x}$ is the vector of inputs, and b is the bias term.

Neural Networks are modeled as collections of neurons that are connected in an acyclic graph in a layered structure, so that the outputs of some neurons become inputs to other neurons of the successive layer, but neurons within a single layer do not share connections. A N-layer network consists therefore of an input layer, an output layer (which for classification task is usually fully-connected and implements a classifier such as SVM or Softmax) and at least one intermediate hidden layer. However, the input layer is not counted among the N layers, so that a single-layer neural network describes a network without hidden layers. In this sense logistic regression or SVM can be viewed as a special case of single-layer neural networks. The size of a NN is usually measured in terms of its number of *parameters*, i.e. the total number of learnable weights and biases.

Modern NNs are composed of many stacked layers (hence the name *deep learning*). Increasing the number of layers indeed (and possibly their size), the capacity of the network (i.e. the space of representable functions) increases. However, it may become easier to overfit the training data. To prevent overfitting in NN there exist many techniques, such as loss regularization, weight decay, dropout, or data augmentation.

Training a NN means learning from the labeled examples $\left(\mathbf{x}^{(i)}, y^{(i)}\right)$, $i = 1, \ldots, m$, good values for all the weights and the biases (i.e. learnable parameters) of the network. Note that input features $\mathbf{x}^{(i)}$ in the case of a NN are raw inputs (images or signals in general), and the intermediate features are directly learned by the network, contrarily to e.g. SVM where usually the input is a hand-crafted vector obtained through some descriptor. In this sense, the learning process of a NN is referred to as *end-to-end*. Often the neural network will discover complex features which are very useful for the given task, but may be difficult for a human to understand since they do not have necessarily a common meaning. In this sense, neural networks can be seen as *black boxes*, as it can be difficult to understand the features they create.

Network parameter optimization is done by minimizing a *Loss function* with algorithms such as Gradient Descent which iteratively update the weights in the direction of the optimal solution evaluating the derivative of the loss function with respect to the weights, through the *backpropagation* technique, over the training samples. The *learning rate* determines how fast weights change. Usually instead of using all the dataset, small *batches* of data are considered at each iteration (in this case, the algorithm is called Batch Gradient Descent), and an *epoch* is a complete pass through

the dataset which may require several iterations. Multiple epochs are then required to converge to a stable solution. To increase the stability of the network and speed up convergence, *batch normalization* [121] is often used. It normalizes the output of a previous layer by subtracting the batch mean and dividing by the batch standard deviation.

The optimal parameter values W^*, namely the ones that allow to converge to a local minimum, are then found through:

$$\begin{aligned} \mathrm{W}^* &= \underset{\mathrm{W}}{\operatorname{argmin}}\, \mathscr{L}(\mathrm{W}) + \lambda \cdot \mathrm{R}(\mathrm{W}), \\ &= \underset{\mathrm{W}}{\operatorname{argmin}}\, \frac{1}{m}\sum_{i=1}^{m} \mathscr{L}\left(y^{(i)}, \hat{y}^{(i)}\right) + \lambda \cdot \mathrm{R}(\mathrm{W}), \\ &= \underset{\mathrm{W}}{\operatorname{argmin}}\, \frac{1}{m}\sum_{i=1}^{m} \mathscr{L}\left(y^{(i)}, f\left(\mathbf{x}^{(i)}; \mathrm{W}\right)\right) + \lambda \cdot \mathrm{R}(\mathrm{W}), \end{aligned} \tag{2.34}$$

where λ is the parameter that controls the impact of regularization introduced by the function $\mathrm{R}(\cdot)$, $f(\cdot)$ is the activation function and $\mathscr{L}(\cdot)$ is the Loss function. Let us now investigate these different terms.

2.2.5.1 Loss function

The Loss function is used to measure the inconsistency between predicted value $\hat{y}$ and the actual label y.

For classification tasks, the two most commonly used output layers are SVM and Softmax classifiers. In case of SVM classifier, the Hinge Loss is used:

$$\mathscr{L} = \frac{1}{m}\sum_{i=1}^{m} \max(0, 1 - y^{(i)} \cdot \hat{y}^{(i)}), \tag{2.35}$$

while in case of Softmax classifier the cross-entropy Loss is employed:

$$\mathscr{L} = \frac{1}{m}\sum_{i=1}^{m} \left[y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]. \tag{2.36}$$

For regression tasks instead, it is common to compute the loss between the predicted quantity and the true real-valued answer. Usually, the Mean Square Error (MSE) or L2 loss is used:

$$\mathscr{L} = \frac{1}{m}\sum_{i=1}^{m} \left(y^{(i)} - \hat{y}^{(i)} \right)^2. \tag{2.37}$$

2.2.5.2 Regularization

There are several ways of avoiding NN overfitting. The most common form of regularization is to employ a regularization term directly in the objective function that sums with the data loss, i.e. the $\mathrm{R}(\mathrm{W})$ function in Eq. (2.34). Usually, L2 regularization is used: for every weight w of the network, the term $\frac{1}{2}\lambda w^2$ is added to the objective, where λ is the regularization strength. Alternatively, L1 regularization adds for every weight the term $\lambda|w|$ to the objective function.

Another commonly used regularization technique is *dropout* [251], which was initially proposed in [106] as a form of regularization applied to neural network layers. Each element of a layer's output is kept with probability p, being otherwise set to 0 with probability $(1-p)$. Dropout improves the network's generalization ability, mitigating thus the risk of overfitting.

Dropout is often presented as an ensemble technique [275], using a different set of hidden units in every learning iteration. At each training step in a mini-batch, the learning procedure with dropout acts like creating a different network (by randomly removing some units) which is then trained using backpropagation as usual. Since at each epoch a different version of the same

network is derived removing some neurons, mathematically this approximates ensemble averaging. Then, at test time the whole network is used (no unit is removed) but with accordingly scaled-down weights.

Other commonly used regularization strategies are early stopping and data augmentation. The first one is a method to prevent overfitting by stopping the learning after a number of epochs if the performance with respect to a defined measure does not improve on the validation set. The second method consists in creating augmented versions of the images of the training dataset, to improve the generalization abilities of the network, by common transformations such as contrast and illumination changes, flipping and rotations applied randomly at every epoch.

2.2.5.3 Activation functions

The most commonly used activation functions are:

- *Sigmoid* – The sigmoid non-linearity takes the form of the sigmoid function previously shown in Eq. (2.2). It takes a real-valued number and "squashes" it into the range between 0 and 1. This activation function has been one of the first to be employed, since it is able to model quite well the firing of a biological neuron. However, it is now rarely used because of two major drawbacks:

  1. *Vanishing gradient problem* – When the neuron's activation saturates at either 0 or 1, the gradient at these regions is almost zero, and almost no signal will flow through the neurons, and no update will be performed in the chain of the backpropagation;
  2. *Not zero-centered outputs* – This fact has implications on the gradient descent algorithm, possibly introducing undesirable zig-zagging dynamics in the gradient updates for the weights;

- *Tanh* – The Tanh, or hyperbolic tangent, takes the form:

$$f(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \tag{2.38}$$

  It can be seen as a scaled sigmoid function, and still suffers from vanishing gradient problem. However, since it "squashes" the values into the range between -1 and 1, its output is zero-centered;

- *ReLU* – The **Re**ctified **L**inear **U**nit (ReLU) is the most used activation function, and it computes the function:

$$f(z) = \max(0, z). \tag{2.39}$$

  It can therefore be implemented by simply thresholding a matrix of activations at zero. Besides, due to its linear, non-saturating form, it has been found to accelerate greatly the convergence of stochastic gradient descent compared to the sigmoid and Tanh functions. However, it can suffer from the *dying ReLU problem*, that is, a large gradient flowing through a ReLU neuron could cause the weights to update in such a way that the neuron will never activate again. This happens especially with a high learning rate, while setting a proper learning rate mitigates the issue;

- *Leaky ReLU* – Leaky ReLU attempts to fix the dying ReLU problem, by allowing a small negative slope for negative values:

$$f(z) = \begin{cases} z, & \text{if } z \geq 0, \\ \alpha z, & \text{if } z < 0, \end{cases} \tag{2.40}$$

where α is a small constant (usually $\alpha = 0.01$). However, the benefit of using Leaky ReLU over simple ReLU is still unclear, as improvements are not systematically obtained using it.

2.3 Classifier combination

2.3.1 Motivation

Classifier ensembles have been receiving increasing attention over the last decades, with many theoretical and empirical studies demonstrating the efficacy of ensembles over a single classifier under different circumstances [23, 79]. From the *No Free Lunch theorem* [279] we know that there is no universally optimal classification algorithm, and classifier combination stems from the imitation of the human nature, which is prone to seek several opinions before making any crucial decision. As humans, we ponder indeed the individual opinions (perhaps based on a prior level of confidence on the interlocutor), and combine them to reach a final decision [212].

According to [64], there are several theoretical and practical reasons why we may prefer an ensemble system over a single classifier:

- *Statistical reasons* – A statistical limitation of a learning algorithm arises when the amount of training data available is too small compared to the size of the hypothesis space. Without sufficient data, the learning algorithm can find several hypothesis functions that give the same accuracy on the training data, but are not able by themselves to generalize in presence of new data. In other words, there is a large *variance* in the decision function selection process (cf. Sec. 2.2.1). By building an ensemble out of all of these classifiers, the algorithm can combine their results, reducing the risk of choosing the wrong classier and of overfitting;

- *Representational reasons* – On the contrary, in many applications the learned model is too simple to approximate the optimal hypothesis function. This is the equivalent of having a large *bias* problem (cf. Sec. 2.2.1). By performing a combination between different hypotheses, it may be possible to expand the space of representable functions and therefore to reduce underfitting;

- *Computational reasons* – Beside the bias–variance trade-off, some classification algorithms face computational issues, due to the fact that they may get stuck in local optima when looking for the optimal decision function. Classification problems indeed are usually NP-hard. An ensemble of classifiers can circumvent the local optima problem by varying the initialization point among the committee.

In addition to these, there is another important motivation for the use of ensembles, as stated in [212]. In the field of data fusion indeed, several sets of features are obtained from various sources. They have however an intrinsically different nature, so that they cannot be used collectively to train a single classifier in an effective way. In such cases, features obtained from each source are used to train different classifiers whose outputs are later combined to make a more informed decision.

2.3.2 Taxonomies of classifier ensemble methods

The variety of ensemble techniques have arisen several taxonomies in the literature, which aim to categorize ensemble methods from different points of view.

Some of them concentrate on a particular classifier, or on a particular approach. For instance, [240] focuses on a taxonomy for neural networks, while the more recent [281] focuses on Multiple Classifier Systems (MCSs).

Kuncheva [148] firstly proposed a broader taxonomy by identifying four levels of questions which are helpful to determine the perspective from which the combination problem can be tackled:

A. *Combination level* – How are the individual inputs combined?

B. *Classifier level* – Do we use same or different classifiers? What base classifier is the best? How many classifiers are needed? Should the classifiers be trained together or incrementally?

C. *Feature level* – Shall we use all features or use a subset for each classifier? How do we select/extract such subsets?

D. *Data level* – How can we manipulate the data submitted for training to the base classifiers so as to ensure high diversity and high individual accuracy?

However, the more complete taxonomy remains Rokach's one [227], which is based on five different dimensions:

1. *Combiner* – It is responsible for combining the classifications of the various classifiers;
2. *Building the ensemble* – It deals with the way the ensemble is generated, i.e. each classifier is trained independently or not;
3. *Diversity* – It answers to how diversity is ensured among the classifiers which compose the ensemble;
4. *Ensemble size* – It determines the number of classifiers in the ensemble;
5. *Universality* – It is related to the fact that some ensemble approaches can be used with any classifier model while others are tied to a specific classifier type.

We will now take advange of the various dimensions of Rokach's taxonomy in order to investigate all the different aspects of ensemble methods.

2.3.2.1 Combiner

There are several ways in which the classifier's outputs can be combined. Firstly, we shall make the distinction between classifiers whose outputs are labels and those whose outputs are continuous values.

When combining classifier label outputs, the most straightforward way is to resort to a voting system. Denoting by $y_n(\mathbf{x})$ the decision of the n^{th} classifier about the test sample $\mathbf{x}$, $n = 1 \dots \text{N}$, and by l the number of possible labels, *Majority Voting* is a system in which the combined decision is the label which is predicted by the maximal number of classifiers:

$$\hat{y}(\mathbf{x}) = \underset{y_l \in \Omega}{\operatorname{argmax}} \sum_n g(y_n(\mathbf{x}), y_l), \tag{2.41}$$

where $g(\cdot,\cdot)$ is an indicator function defined as:

$$g(y_n(\mathbf{x}), y_l) = \begin{cases} 1, & \text{if} \quad y_n(\mathbf{x}) = y_l, \\ 0, & \text{if} \quad y_n(\mathbf{x}) \neq y_l. \end{cases} \tag{2.42}$$

Majority Voting is the standard baseline to compare with. A slight modification of this method is the *Unanimity Voting*, in which a combined decision is given only if all the classifiers agree on the label, rejecting otherwise the input $\mathbf{x}$. If the classifiers in the ensemble are not of identical accuracy, then the *Weighted Majority Voting* attempts to give the more competent classifiers more power in making the final decision:

$$\hat{y}(\mathbf{x}) = \underset{y_l \in \Omega}{\operatorname{argmax}} \sum_n w_n g(y_n(\mathbf{x}), y_l). \tag{2.43}$$

In particular, each classification output has a strength proportional to its assigned weight w_n, which can be fixed or dynamically determined for the specific instance to be classified. The accuracy level of the classifiers can be known *a-priori*, derived through analysis of the outputs or even learned.

Borda count [107] is considered to be one of the simplest non-linear combination algorithm. It consists of a a rank-based combination scheme where each classifier ranks the classes according to their chances to be the correct (true) class, and then the sum of accumulated scores of each label is calculated.

If the output of the classifier is a continuous value (probabilistic or not), voting approaches do not exploit all the available information, and thus some combination function can rather by used. Traditionally, some commonly employed combination functions are:

- *Average*: $\hat{y}(\mathbf{x}) = \frac{1}{\mathrm{N}} \sum_{n=1}^{\mathrm{N}} y_n(\mathbf{x})$,
- *Median*: $\hat{y}(\mathbf{x}) = \mathrm{median}_{n=1\ldots\mathrm{N}}\, y_n(\mathbf{x})$.

Note that the median function can be replaced by minimum or maximum functions for some particular applications.

Assuming independent probabilistic classifiers, we can transform the classification for classifier decision regarding class y_l into a probability $\mathrm{P}_n(y = y_l|\mathbf{x})$ for every classifier $n = 1\ldots\mathrm{N}$, and then use a product rule to obtain the joint probability across all the classifiers:

$$\hat{y}(\mathbf{x}) = \mathrm{P}(y = y_l|\mathbf{x}) = \prod_{n=1}^{\mathrm{N}} \mathrm{P}_n(y = y_l|\mathbf{x}). \tag{2.44}$$

These combiners are all called *non-trainable*, because once the individual classifiers are trained, their outputs can be fused to produce an ensemble decision, without any further training. On the contrary, *trainable* combiners include *Naive Bayes*, *Behavior-Knowledge Space* (BKS), and also Weighted Majority Voting where the various weights are learned.

In particular, using Naive Bayes classifier for combining various classifiers as in [228] assumes that the classifiers predictions are conditionally independent given the class. Behavior-Knowledge Space (BKS) [117] was proposed with the advantage of not relying on prerequisite hypotheses such as the statistical independence of classifier outputs. It is based on a look-up table of dimensions $\mathrm{N} \times \mathrm{L}$ which is learned at training time, where N is the number of classifiers and L the number of possible labels in Ω. However, the method does not scale well in either the number of base classifiers or the number of classes as there may be some configurations that are never visited.

We shall finally make a distinction between the probabilistic, evidential and fuzzy frameworks. If the combination techniques proposed above are rather applicable in a probabilistic framework, other frameworks imply the use of different rules of combination. Belief Function (BF) theory (also called *evidential*) [59, 247], by making a distinction between *imprecision* and aleatory *uncertainty* concepts, generalizes probability and set theories by providing a number of aggregation operators which are useful in defining new combination rules. Since BF framework will play a key role in the definition of the proposed method, we will explain it better in Chapter 4. Fuzzy set theory [13, 292] was conceived to present "soft" classification of elements, which is more adapted to the way people create categories in the real world. Thanks to fuzzy set theory, possibility theory [291] was later introduced to handle incomplete information.

2.3.2.2 Building the ensemble

This property refers to whether the various classifiers are *dependent* or *independent*. In a dependent framework the outcome of a given classifier affects the subsequent classifier. Alternatively, each classifier can be built independently and their results can be combined in some fashion in a second moment.

Specifically, there are three main categories of topology: parallel, sequential or hybrid combinations. In parallel fusion, the base classifiers work independently, and the feature vectors may

or may not be learned from the same training examples. The output of a given classifier may not be present in the input of another classifier [148]. In sequential fusion, base classifiers are stacked in a sequential way and the decision of one classifier depends on a previous one [219]. This approach implies some kind of ranking or ordering of the classifiers, and usually the primary one is the cheapest while the subsequent ones have higher exploitation cost [83]. Finally, there exists hybrid hierarchical fusion which consists in a combination of parallel and sequential architectures.

Among the parallel architectures, we can find *Bagging* and *Multiple Classifier Systems* (MCSs). The first one is based on a set of weak classifiers (whose discriminative powers are slightly stronger than the random one), while the second one relies on a smaller pool of stronger classifiers, not necessarily derived from the same model.

Bagging [23] (**B**ootstrap **Agg**regat**ing**) produces N training sets, containing a subset of r samples from the initial one, by random sampling with replacement. This means that some observations may be repeated in each new training dataset. It is based on the non-parametric Bootstrap sampling technique [233] which is a statistical method used to analyze the variability of an estimate and quantify its uncertainty. Intuitively, if the estimate values are similar when training the model with respect to different datasets (obtained by sampling from the original training set), then we can have a high level of confidence in the estimate. The various decisions of the classifiers are often combined using majority voting to obtain a global verdict. This method is particularly good in presence of high-variance, to reduce overfitting.

A variation of bagging is *Random Subspace Method* (RSM), also called *Feature Bagging*, since the same principle of bagging is applied to the feature space instead of to the sample set. Specifically, it consists in training the classifiers in random sub-spaces of dimensionality lower than the dimensionality of the original space, obtained by sampling the features instead of the training samples. Input vectors in high dimensional spaces are notoriously more prone to overfitting, and this splitting can reduce the risk of this problem increasing the generalization ability of the whole system. *Random Forest* [24] is a particular ensemble classifier that takes advantage of RSM. Specifically, it operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (for classification) or mean prediction (for regression) of the individual trees. Another established framework able to benefit from the information provided by multiple features is the decision tree analysis. Recent work highlighted that intrinsic uncertainty related to learning as well as uncertainty due to imprecise data may be jointly managed inside the decision tree by defining entropy intervals from evidential likelihood [176].

Restricted to SVM as base classifier, also Multiple Kernel Learning (MKL) falls under the parallel architecture category. It is a well established methodology which aims to combine different kernels relying on different data representations as a linear combination, by casting the information fusion task as a convex optimization problem [90]. It provides a way to benefit simultaneously from all the available features, automatically learning the optimal weights to linearly combine them. The problem scales very well with the number of individual classifiers, but the main limitation of MKL is the difficulty to interpret the final decision and to take into account the imprecision coming from different sources.

On the other side, MCSs [281] are systems composed by few, strong, heterogeneous classifiers which are usually learned from the same input data. For a given classification task, a MCS is generally able to exploit the strengths of the individual classifier models to produce a high quality compound system overcoming the performance of the individual classifiers by combining them with some combination rule able to exploit their complementarity.

Among the sequential architectures, we can recall *Boosting* (*AdaBoost, Gradient Boosting*), and *stacked generalization*. They all assume a sequence of classifiers trained after the other taking into account the previous model, but differ in the way the previous model is accounted for, and in the number of classifiers involved (many weak classifiers in boosting, few and stronger classifiers in stacking).

Boosting techniques [79] try to add new models that perform well where the previous models exhibit low performance. AdaBoost [80] (**Ada**ptive **Boo**sting) is similar to bagging, in the sense

that both methods rely on several weak classifiers trained on subsets of the initial training data. However, unlike bagging, the observations are weighted and therefore some of them (the most problematic to be classified) will take part in the new sets more often. In AdaBoost each classifier is thus trained on the data on which the previous classifier failed, redistributing the weights after each training step. Misclassified data increases its weights to emphasize the most difficult cases. In this way, subsequent learners will focus on them during their training.

Another popular boosting technique is Gradient Boosting [81], which casts the boosting problem to an optimization problem and can be interpreted as an ensemble gradient descent algorithm in function space. In Gradient Boosting, weak learners are added iteratively in such a way that the loss function, generally dependent on error residuals, is minimized.

A variant of Gradient Boosting which has proven to be particularly successful in machine learning competitions over the last years is XGBoost [39] (E**x**treme **G**radient **Boost**ing). It is based on Newton-Raphson method for approximation (hence it is named sometimes *Newton Boosting*), and it is useful in presence of trees-based classifiers, since it presents a clever penalization of trees and a proportional shrinking of leaf nodes. Generally, XGBoost is faster than Gradient Boosting, but this latter can be applied to a wider range of applications. Finally, as opposite to bagging, boosting techniques are particularly good in case of high bias problems, but they tend to suffer in presence of noisy data.

Another popular sequential architecture is stacked generalization [278] (or stacking), which like MCS usually implies far less models than the ones needed for bagging or boosting and the use of a heterogeneous pool of classifiers. However, unlike MCS, stacking uses a new model to learn how to combine the predictions from previous models trained on the dataset. The predictions from the existing models or sub-models are combined using a new one, and for this reason stacking is often referred to as *blending*, as the predictions from sub-models are blended together. The various models may be very different in nature, i.e. derived from different classifiers and blended together with another one which is usually called the *aggregator* model.

Regarding hybrid architectures, we can recall [273], which combines MKL with stacked generalization in order to achieve greater computational efficiency and greater performance in terms of predictive accuracy, by separating the kernel set into subsets, each subset of kernels leading to a different combination of kernels which are then aggregated together into a single prediction. Stacking is exploited also in [277] in conjunction with a bootstrap procedure to achieve further improvements on the performance of bagging for regression problems.

2.3.2.3 Diversity

In an ensemble, the combination of the output of several classifiers is only useful if they provide *diverse* responses, at least for some inputs [262]. Is it therefore important to be able to create diversified classifiers, which *may* lead to uncorrelated errors and *may* in turn improve classification accuracy when combined [114].

Brown et al. [27] focuses on the different ways *diversity* can be achieved within the ensemble, i.e. whether it is *implicitly* obtained by generating classifiers using different mechanisms (training them on different learning data samples or on different region of the features space), or whether it is *explicitly* ensured by a measured gain in diversity with some specific metric.

Implicit methods for inducing diversity involve:

- *Partitioning the data points* – This consists in training the base classifiers on different training sets, and it is particularly useful in two cases. Firstly, it allows us to apply bagging and boosting methodologies which implies the subdivision of the original training set in smaller ones obtained via (weighted) sampling of the input examples. Secondly, it is strongly connected to the distributed data paradigm, where huge databases may impede to train the classifiers under specified time constraints, imposing to resort to sampling techniques to obtain manageable dataset partitions. A well known approach is cross-validated committee [144], which requires the minimization of overlapping between dataset partitions;

- *Using different parameters in the training of the individual classifiers* – For example, use several random weight initializations for neural networks, or several different regularization hyper-parameters C for SVMs. A related approach is the use of Bayesian Neural Networks, which learn a distribution over the network's weights instead of point estimates [177, 191];

- *Selecting subsets of features* – It can be performed in many different ways. Firslty, RSM can be applied in order to perform feature bagging (e.g. with a Random Forest classifier). Another strategy may consist in selecting the classifier with the best performance for each partition of the feature space, using a clustering algorithm to partition the feature space, instead of a random sampling [147]. Then, we may carefully select different hand-crafted features to tackle the classification problem from different perspectives (e.g. by creating an ensemble of SVM classifiers based on different descriptors). Alternatively, information gain can be employed as in [221] for actively selecting features combining the collected evidence over time while taking into account the amount of available training data for each class. There exists also Selective Multiple Kernel Learning (SMKL) method [253], which preserves the sub-kernels with complementary information by guaranteeing the high discrimination and large diversity of pre-selected sub-kernels. Again, [284] extended the single kernel boosting method to multiple kernel boosting methods. Multiple kernels are used to construct a better classifier, and a kernel sampling method is designed to only sample a subset of kernels for combination in each iteration. Finally, a similar but different approach is presented in [87], where a set of base classifiers for each test sample is dynamically selected on the basis of a classification gain computed using a probabilistic model that exploits the outcome from previous observations. In this way, each classifier in a large ensemble is viewed as a potential observation that might inform the classification process itself.

- *Choosing different label targets* – In multi-class classification, each individual classifier may solve a different classification task. An example of a classifier ensemble approach in this category is the *Error Correcting Output Code* (ECOC) ensembles [65], where each classifier solves a dichotomy, separating two groups of classes. It consists in representing each class label by a code-word (string of "0" and "1" values only) of length N, where N is the number of classifiers involved in the fusion. Then, each classifier discriminates between two subsets of Ω only, and outputs a binary value representing a subset of labels. To decide the label of a test sample $\mathbf{x}$, we choose the label having the closest code-word to the obtained code-word of $\mathbf{x}$, measured e.g. in terms of Hamming distance;

- *Using different classifier models or hybrid ensembles* – This corresponds to using MCSs or stacked generalization techniques, where small pools of strong classifiers are combined in a parallel or serial way respectively. In these methods, the diversification between the models is ensured by using several, different classifiers, e.g. a multiple classifier system composed by a convolutional neural network and a SVM combined in a unique output, or stacking of k-Nearest Neighbors and Perceptron model aggregated by logistic regression. Hybrid approaches seek to exploit the strengths of the individual components, obtaining enhanced performance by their combination.

Although the ensemble community agrees on the importance of ensuring diversity among the various classifiers, the authors of [149] could not find a definitive connection between explicit measures and the improvement of the accuracy. They also stress the fact that "diversity" and "independence" are not synonyms and are not necessarily related. Thus, they conclude that it is unclear whether diversity explicit measures have any practical value in building classifier ensembles, privileging the use of implicit techniques to intuitively ensure diversity in the creation of the ensemble.

2.3.2.4 Ensemble size

There are three common approaches for determining the ensemble size:

1. *Pre-selection of the ensemble size* – This is the most simple way to determine the ensemble size, and relies on a hyper-parameter controlling the number of classifiers used in the ensemble. Algorithms such as bagging or MCSs belong to this category;

2. *Selection of the ensemble size while training* – In this case, a new classifier is added to the ensemble if its contribution to the ensemble performance is significant. Performance measures can include a variance reduction measure, or an accuracy-based measure. Usually these algorithms also have a controlling parameter which bounds the maximum size of the ensemble;

3. *Overproduce and Select* – This approach, also called *ensemble pruning*, derives from the observation that bigger ensembles are not necessarily better ensembles. As in decision tree induction, it is sometimes useful to let the ensemble grow freely and then prune the ensemble in order to get more effective and compact ensembles, to comply also with the accuracy–cost trade-off. Different strategies involve *pre-combining pruning*, where the pruning is performed before combining the classifiers based on their individual classification performance measured on a separate validation set, and post-combining pruning, where classifiers are rather removed from the ensemble if they provide insufficient contribution to the collective [215].

Moreover, there exist several factors that may define how many component classifiers should be used within an ensemble:

- *Accuracy–Cost trade-off* – Increasing the number of classifiers usually increases the computational cost related to their training; in addition, there is generally a limit in the possible increase in accuracy due to the addition of new classifiers to the ensemble (besides the two facts being not necessarily related);

- *The nature of the classification problem* – In some ensemble methods, the nature of the classification problem determines the number of classifiers;

- *The amount of computational power available* – For example, in independent methods the number of processors available for parallel learning could be an upper bound on the number of classifiers that can be considered.

There exist some common approaches in determining the ensemble's size, based on the type and goodness of classifiers which are involved. To this extent, Fig. 2.5 gives a view of classifier combination approaches in the space of ensemble's size (denoted by N) and performance of the individual base classifiers.

In the top-left corner there is the ideal system, composed by one yet perfect classifier. Increasing the number of classifiers but remaining in the upper-left quarter, we find methods that rely on small ensembles of strong classifiers. These types of systems are usually composed by models derived from a heterogeneous pool of classifiers, and usually the fusion is done between their independent outputs, with ingenious combination rules to draw upon their diversity. They are thus MCSs. Always in this part of the scheme, we can find classifier models combined through stacked generalization. The combiner is a classifier itself, built upon the outputs of the individual classifiers. The individual outputs, i.e. class labels or degrees of support for the various classes, are treated as intermediate features. Stacking treats the outputs of the individual classifiers as inputs for a new trainable classifier, which itself constitutes the combiner. Unlike bagging or boosting, stacking is used to combine a small number of models of different types. Generally, in presence of small heterogeneous ensembles of strong classifiers, each ensemble member usually knows well a part of the feature space only, so that their fusion can be seen as a *cooperation*.

On the contrary, in the bottom-right quarter we find large ensembles of weak classifiers. Popular ensemble methods in this group are bagging and boosting, where the committee of models

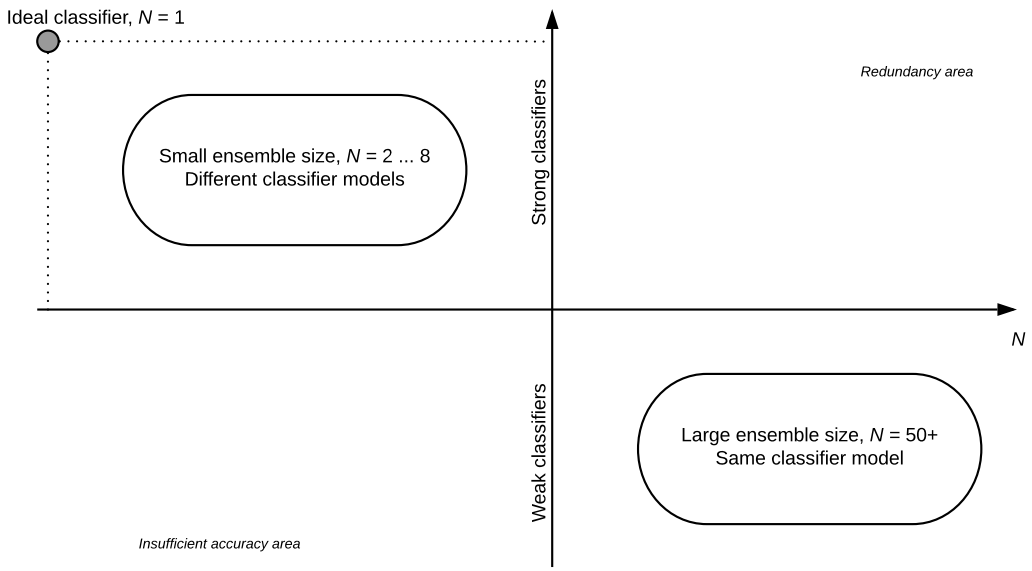


Figure 2.5: Visual representation of classifier combination approaches. We place our work in the upper-left quarter (figure partially inspired by [148]).

is derived from the same base classifiers e.g. changing the input data for the learning stage. Unlike the previous case, here each ensemble member is supposed to have knowledge of the whole feature space, even though not being very reliable. The fusion is done in a *competitive* situation (an exception is feature bagging, where each classifier is trained with a different, limited set of features).

Similarly to this subdivision, Valentini and Masulli [264] propose a dichotomized view of ensemble techniques, dividing them into *decision optimization* methods and *coverage optimization* methods. The former ones use a fixed set of carefully designed and highly specialized classifiers, and the goal is to find an optimal combination of their classifications. The latter ones generate a set of mutually complementary, generic classifiers that are combined to improve predictive performance. In this sense, this division is analogous to the cooperative vs. competitive classifier ensembles.

The upper-right quarter includes large ensembles of strong classifiers. However, this approach is not that interesting, because of the redundancy in the classifier outputs which leaves no room for diversity. It is a waste of resources, as the computational burden to perform the fusion would be too expensive for the actual derived gain.

Finally, the bottom-left quadrant corresponds to small ensembles of weak classifiers. It represents the most challenging situation, because diversity must be ensured in such a way that no accuracy is wasted, and that the classifiers complement each another. While being possible in theory, this approach is practically not easy to put in place and represents an active area of research.

2.3.2.5 Universality

Universality (with respect to the base classifier) is a property which is related to the fact that some ensemble approaches can be used with any classifier model, while others are tied to a specific classifier type.

Among the classifier-dependent methods, we can recall [98, 172, 241] which were developed specifically for neural networks. Other procedures were developed specifically for SVM [256], decision trees [228] and logistic regression [238] classifiers. Examples of classifier-specific ensembles are then the Random Forest and GXBoost, whose base classifiers are necessarily Random Trees. In the same way, also MKL is restricted to the use of SVM classifiers, although with different kernels.

Alternatively, there are classifier-independent methods which can be applied on any given

classifier or set of classifiers. Traditional bagging and boosting could be applied to any base classifiers (although in practice they are often used in conjunction with trees or, more in general, inexpensive classifiers since they involve a large number of them). Among these methods we can then find MCSs and ensembles created through stacking, where diversity in the type of classifiers involved is necessary and encouraged.

2.3.3 Our approach as taxonomy's entry

A contribution of this work is a MCS for pedestrian head detection in high-density crowds, that will be explained over the next Chapters. Now, we are able to categorize the proposed method with respect to Rokach's five dimensions taxonomy:

1. *Combiner* – The combination is performed in the BF framework after the proposed BBA allocation;

2. *Building the ensemble* – We rely on a MCS composed by an SVM-ensemble and a CNN-ensemble. The SVM-ensemble is composed by several SVM classifiers trained with respect to different features. We employ an SVM as base classifier for all the members of the ensemble and we rely on different descriptors in order to obtain the feature vectors, which give different view of the same input data. Conversely, the CNN-ensemble is given by the application of dropout at inference time over a CNN especially designed to recover small objects and to work with small amounts of data. Besides, our base classifiers being able to capture different shades of the same input data, their fusion is done in a cooperative way;

3. *Diversity* – It is implicitly obtained by using the different classifiers obtained with respect to different features which are able to capture different views of the same input data, both hand-crafted (SVM-ensemble) and automatically learned (CNN-ensemble);

4. *Ensemble size* – We place our algorithm in the upper-left quarter of Fig. 2.5, as we exploit indeed a small ensemble of rather strong SVM classifiers based on different descriptors in order to perform fusion of their independent outputs, and a small ensemble of CNN realizations through dropout;

5. *Universality* – The proposed approach will be illustrated with respect to the use of SVM and CNN as base classifiers. Regarding SVM, we have chosen the descriptors for our specific problem of pedestrian detection in high-density crowd, therefore the method is rather *application-dependent*. Moreover, the proposed BBA allocation is performed in two successive steps, the first one being dependent on the fact that we rely on SVMs for classification. The distance to the hyper-plane separation of the various samples will indeed play a key role in determining the discounting factor to derive the associated BBA. Nevertheless, the proposed method could be easily adapted to any classifier which provides a score as output of the classification. Finally, considering a different application, the proposed MCS could be easily applied with more adapted descriptors. Regarding the CNN-ensemble, the approach will be illustrated on the basis of the proposed network but it can be applied to any fully convolutional neural network with encoder-decoder structure.

Chapter 3

SVM descriptors for pedestrian detection in high-density crowds

Contents

3.1 State of the art

Pedestrian detection in high-density crowds is a difficult task, especially because common methods applied in sparse scenes are not applicable in denser scenarios, for a number of reasons explained in Sec. 1.1.2.1, such as absence of background, heavy occlusions of body parts, high visual homogeneity and small size of the targets.

When using traditional classifiers such as SVM, a feature engineering step is necessary in order to design a distinctive representation of the object of interest. In such difficult applications however, it is impossible to find a feature representation which is able to perfectly describe all the different possible shades of pedestrian's head appearance.

Among the appearance cues, the simplest descriptors rely on a local color histogram, which may be associated to skin, hair or clothes. However, this approach is limited by multiple factors: the object resolution needs to be relatively high, the color spaces are not discriminative enough for difficult tasks, and lastly many surveillance cameras provide gray level data.

Over the last decade many descriptors have been proposed for face detection, starting from the Viola-Jones [265] one. The Viola-Jones algorithm is an ensemble-based detector which exploit a cascade of classifiers based on Haar features, usually trained with AdaBoost technique. However, face detectors are unsuited for our application, since pedestrian faces are not detailed enough.

Several effective approaches have been then proposed in the field of person re-identification and human characterization, e.g. the viewpoint-invariant Ensemble of Local Features (ELF) [94], the Symmetry-Driven Accumulation of Local Features (SDALF) [12], gBiCov [175] which is a combination of Biologically Inspired Features (BIF) [225] and Covariance descriptors [113], local descriptors encoded by Fisher Vectors [174], salience matching [299], and mid-level filter [300]. However, they are not adapted in presence of clutter and occlusions.

Among the descriptors related to the image gradient, the Histogram of Oriented Gradients (HOG) descriptor [51] is very popular and has exhibited in various contexts an excellent performance when used either in conjunction with a linear SVM, or with a histogram intersection kernel (HIK) [270]. It has been proposed initially for pedestrian detection with all the visible body, but it has been applied to the recognition of other objects. More generally, the contour related to the specific shape of the head and shoulders is highly discriminative, but it may fade away due to clutter. Supervised learning may be used in order to enhance the local edge map according to a training contour dataset [269], but it is also advisable to rely on descriptors aimed at other features than shape. More recently, curvature histograms [77], i.e. second order features, have been employed as a natural expansion of first order features provided by HOG. However they are not meaningful enough by themselves to exceed the performance of the latter and come at a higher computational cost.

Another feature which has been often used for detection in crowded scenes is the Local Binary Pattern (LBP) operator [196]. The traditional use of LBP is in texture classification, but due to its local sampling strategy it exhibits a reasonable robustness to occlusion as well. Some alternative solutions are the covariance matrix based descriptors [113], but the main advantages of LBP are its compactness and low computational cost. Also related to texture representation, Gabor filter banks have been used for head detection [159] to encode the local frequency and orientation.

Related to the field of image-based matching we can find Scale Invariant Feature Transform (SIFT) [170] which is invariant to translations, rotations and scaling transformations in the image domain and robust to moderate perspective transformations and illumination variations. It works by accumulating statistics of local gradient directions of image intensities to give a summarizing description of the local image structures in a neighbourhood around each interest point. The procedure of SIFT therefore includes mainly three steps: keypoints detection, descriptor assembling, and keypoint association [282].

Over the years many variants have been proposed, like PCA-SIFT [132], CSIFT [1], SURF [11] and ASIFT [187], just to cite some of them. PCA-SIFT exploits Principal Component Analysis (PCA) [276] to be able to reduce the dimensionality of the resulting feature vector. CSIFT adds color invariance to the basis of SIFT, allowing to not discard the available information present in color images. Speeded Up Robust Features (SURF) is very similar to SIFT in its extent, but it adopts different processing methods in every step, e.g. it uses a Hessian matrix to determine candidate keypoints and adds a non-maxima suppression step, besides calculating Haar wavelet-based gradients in a circular area rather than squared. These improvements allows for a gain in performance. ASIFT (Affine SIFT) improves the performance in situations of strong affine issues, by simulating the rotation of camera's optical axis. Still inspired by SIFT, DAISY descriptor [258] has been proposed to estimate dense depth maps from wide-baseline image pairs, showing high robustness against many photometric and geometric transformations.

In their original formulations, these descriptors are used for matching *sparse* interest points between different images. However, they have also been applied at dense grids (e.g. dense SIFT) along with a supervised classifier, showing good performance thanks to their intrinsic robustness to perspective and lighting variations. The SIFT expansion to the dense setting has been proposed in [73], while SURF can be naturally adapted to be computed densely in an efficient way. DAISY can also be computed efficiently at every pixel, and unlike SURF does not introduce artifacts that degrade the matching performance when used densely.

3.2 Considered descriptors

Among the different detectors it is not immediately clear which ones are suited for high-density crowd pedestrian detection. In the following, we present the four descriptors used in this work, explaining why they have been chosen and performing a validation of their parameters.

3.2.1 HOG

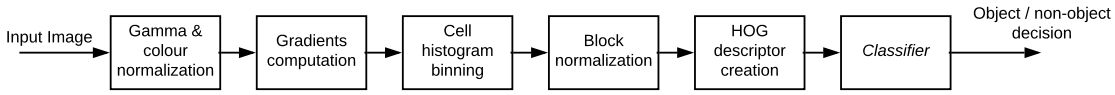


Figure 3.1: Overview of HOG descriptor (figure inspired by [51]).

The Histogram of Oriented Gradients (HOG), introduced in [51], grasps the shape of interest by histograms of local intensity gradients or edge directions. The basic idea behind it is that local object appearance and shape can be well characterized by the distribution of local intensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge positions.

In practice, for each pixel, HOG is computed considering a large window around it, which is in turn divided into small cells, and for each cell a local histogram of gradient directions is accumulated. Figure 3.1 provides an overview of the various steps required in order to perform object detection using the HOG descriptor:

- *Gamma and colour normalization* – The first step consists in a global gamma correction of the input image, although it provides only a modest effect on performance as stated by the authors;
- *Gradient Computation* – Gradients are computed at every location, after an optional Gaussian smoothing. The authors tested several discrete derivative kernels, and linear 1D $[-1, 0, 1]$ kernel (with its transposed form for the vertical response) resulted to be the best;
- *Cell histogram binning* – At this point, the window around the pixel of interest is divided into small cells. For each cell a histogram of gradient directions is accumulated, where the vote of each pixel is weighted with respect to the magnitude of its gradient, introducing an important source of non-linearity. The cells themselves can either be rectangular or radial in shape, and the histogram channels are evenly spread over 0 to 180 degrees or 0 to 360 degrees, depending on whether the gradient is unsigned or signed. The authors found that unsigned gradients used in conjunction with 9 histogram channels performed best in their human detection experiments;
- *Block normalization* – Cells are then grouped into possibly overlapping blocks, and the histograms contained in each cell are normalized for each block to take into account local changes in lighting and contrast, for better invariance to illumination and shadowing. Several block normalization schemes have been proposed, such as *L2-norm*, *L2-Hys* which is a L2 normalization followed by a clipping operation to impose an upper bound on the maximum value (by default set to 0.2) and re-normalization, *L1-norm* and *L1-sqrt*, i.e. a L1 normalization followed by square root. The block normalization step has been proved by the authors to be essential, reducing the performance of almost 30% when not applied;
- *HOG descriptor creation* – The final descriptor is composed by concatenating all the normalized cell histograms of every block in the detection window;
- *Classifier* – Once the descriptors have been obtained, we can train a classifier (e.g. a SVM) with them, and later use it for decision about new samples (which will be also represented in terms of HOG features).

Table 3.1: Example of the basic LBP operator applied in a 3×3 neighborhood at central pixel c having gray-scale value g_c. The binary code is obtained by concatenating binary values from the top-left corner (g_0 position) in a clock-wise fashion. In this example, the resulting binary code is 11010011, which corresponds to the decimal label 211.

Pixel's neighborhood:			Gray-scale values:			Binary thresholding:		
g_0	g_1	g_2	250	127	10	1	1	0
g_7	$\boldsymbol{g_c}$	g_3	44	**44**	112	1		1
g_6	g_5	g_4	45	21	3	1	0	0

Since the HOG descriptor operates on localized cells, the method upholds invariance to geometric and photometric transformations, except for object orientation. Moreover, as the authors discovered, coarse spatial sampling, fine orientation sampling, and strong local photometric normalization allows for the individual body movement of pedestrians being ignored as long as the individuals maintain a roughly upright position. Indeed, upper body parts are not concerned in significant appearance changing also in presence of pedestrian movements, and continue to keep their discriminative power. For this reason, the HOG descriptor is suited for human detection, and we decided to apply it in our high-density context as a head detector, for its ability of recognizing human head-shoulder patterns robustly.

3.2.2 LBP

The Local Binary Pattern (LBP)[196] is a powerful texture descriptor, whose aim is to efficiently summarize the local structures of an image. It has been applied to many applications, e.g. LBP-based facial image analysis has been one of the most popular and successful applications of it in recent years. The most important properties of LBP are its high tolerance to monotonic illumination changes and its computational simplicity.

The original LBP operator labels the pixels of a gray-scale image with decimal numbers, called *Local Binary Patterns* or *LBP codes*, which encode the local structure around each pixel. Table 3.1 shows an example of the basic LBP operator. For each pixel, the 3×3 neighborhood is thresholded with respect to the center pixel value, and the resulting string reading the neighbors' values clock-wise starting from the top-left corner is interpreted as a binary number and used as a label in its decimal form.

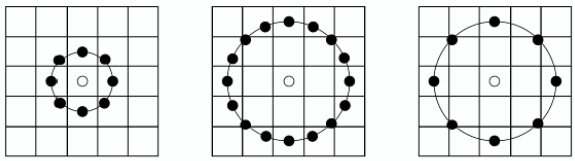


Figure 3.2: Examples of extended LBP operators, i.e. $\text{LBP}_{8,1}$, $\text{LBP}_{16,2}$, $\text{LBP}_{8,2}$.

One limitation of the basic LBP operator is that the small 3×3 neighborhood is not enough to capture dominant features in presence of larger scale structures. To perform texture detection at different scales, the operator has been later generalized to use neighborhoods of different sizes [197]. A local neighborhood is defined with respect to a central pixel as a set of points evenly sampled on a circle around it. The sampling points which do not fall exactly in the center of a pixel are bilinearly interpolated. This allows the use of any radius and any number of sampling points in the neighborhood. The notation $\text{LBP}_{p,r}$ denotes LBP operator computed with a neighborhood of p sampling points on a circle of radius r. Figure 3.2 shows some examples of the extended $\text{LBP}_{p,r}$ operator sampling scheme.

Formally, the LBP operator at (x_c, y_c) location having a gray-scale value of g_c is defined as:

$$\mathrm{LBP}_{p,r}(x_c, y_c) = \sum_{p=0}^{p-1} s(g_p - g_c)2^p, \tag{3.1}$$

where g_p are the various gray-scale values of the surrounding pixels in the circle neighborhood, and the function $s(x)$ is defined as:

$$s(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ 0, & \text{if } x < 0. \end{cases} \tag{3.2}$$

This formulation is by definition invariant to monotonic gray-scale transformations, being able to preserve pixel intensity order in the local neighborhoods.

Also in [197], the authors extended the descriptor in order to obtain rotation invariance, by performing circular clock-wise bit-wise right shifts so that a maximal number of the most significant bits is 0. This version goes under the notation of $\mathrm{LBP}^{ri}_{p,r}$, where ri stands for "**r**otation **i**nvariant".

However, it was shown that such a rotation-invariant definition of the LBP operator does not necessarily provide discriminative information, since the frequency of occurrence of the many possible individual patterns varies greatly. To this extent, the authors observed in [197] that only a limited subset of the 2^p patterns are indeed very discriminative, representing well local primitives such as corners or edges, and holding most of the information related to texture. These patterns are called *uniform patterns*, and present at most two transitions from 0 to 1 or vice-versa in the corresponding binary string. For instance, patterns 00000000 (0 transitions), 00001111 (1 transition) and 00110000 (2 transitions) are considered as *uniform*, whereas 00110011 (3 transitions) or 01010101 (7 transitions) are not. The enhanced version of LBP which takes into account the uniformity of patterns is denoted $\mathrm{LBP}^{riu2}_{p,r}$, where $u2$ means that **u**niform patterns with at most **2** 0-1 (or 1-0) transitions.

Besides the higher discriminative power, the use of uniform patterns allows us also to limit the number of possible value for the labels. For example, considering $\mathrm{LBP}^{ri}_{8,1}$, the label values span between 0 (00000000 pattern) and 255 (11111111 pattern), for a total of 256 possible different values. Now, considering $\mathrm{LBP}^{riu2}_{8,1}$, a simple look-up table allows us to assign a different label to the various uniform patterns while at the same time placing in the same last bin all the patterns which are not interesting, yielding to only 59 possible label values.

Finally, to obtain the texture descriptor, histograms of LBP labels are calculated over the region of interest. Thus, the reduction of the number of possible values using uniform patterns is particularly important as it allows to obtain more compact yet highly discriminative feature vectors.

In this study, we employ the LBP operator to derive a descriptor for the following reasons. Firstly, it can be noted that head-shoulder body parts of people in high-density crowds are so close one to another that they look like a texture. Then, among the various texture descriptors LBP can be computed fast and provides a compact representation which is particularly useful to mitigate the risk of possible overfitting in applications where the training set is quite small.

Lastly, among the various applications of LBP, we shall mention that HOG and LBP have been already used together in [272] to perform human detection with partial occlusion handling through part detectors applied in ambiguous windows. Instead of simply concatenating the two feature vectors corresponding to HOG and LBP, an LBP operator is computed for every cell of the HOG descriptor, raising up an augmented HOG-LBP feature.

3.2.3 Gabor filter banks

Gabor filters in their two dimensional version have been firstly introduced in [55] with the purpose of modeling simple receptive fields in striate cortex. Now, they are widely used in object recognition and texture segmentation, for capturing global and local information thanks to the flexibility in the choice of spatial scales and orientations. An input image $\mathrm{I}(x, y)$ is convolved with a Gabor

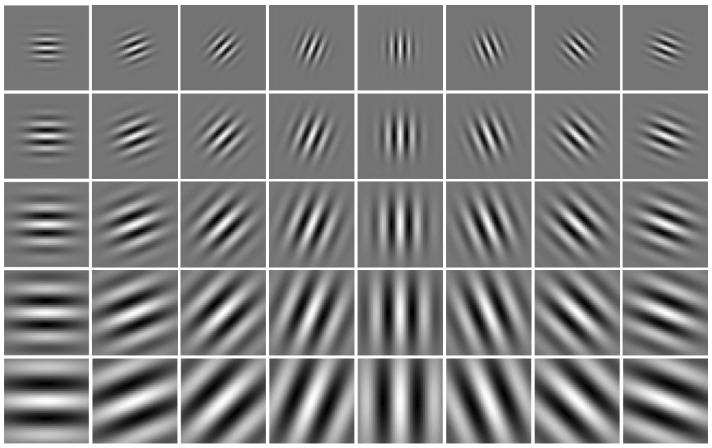


Figure 3.3: Examples of a Gabor filter bank composed by kernels at 5 scales and 8 orientations.

filter $g(x,y)$, i.e. a 2D Gaussian function $\mathrm{G}(x,y)$, known as the *envelope* modulated by an oriented complex sinusoidal signal $s(x)$, called the *carrier*, at different scales and orientations. Gabor filters are thus band-pass filters which are able to detect image gradients of a specific frequency and orientation.

Specifically, the two-dimensional complex Gabor function is defined as:

$$\begin{aligned} g(x,y) &= \frac{1}{2\pi\sigma_x\sigma_y}\cdot\exp\left(\left(\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi i f x+\phi\right)\right), \\ &= \mathrm{G}(x,y)\cdot\exp\left(\left(2\pi i f x+\phi\right)\right), \end{aligned} \tag{3.3}$$

where f and ϕ represent the frequency and the phase offset of the sinusoidal carrier function respectively, while

$$\mathrm{G}(x,y)=\frac{1}{2\pi\sigma_x\sigma_y}\cdot\exp\left(\left(\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)\right)\right), \tag{3.4}$$

is the 2D Gaussian envelope. Parameters σ_x and σ_y are the standard deviations of the Gaussian function in the two dimensions, regulating the ellipticity of the kernel.

The real and imaginary parts of the complex Gabor function can be computed as:

$$\begin{aligned} \Re\left(g(x,y)\right) &= \mathrm{G}(x,y)\cdot\cos\left(2\pi i f x+\phi\right), \\ \Im\left(g(x,y)\right) &= \mathrm{G}(x,y)\cdot\sin\left(2\pi i f x+\phi\right), \end{aligned} \tag{3.5}$$

where i represents the imaginary unit i, which is defined by its property $i^2=-1$.

A filter bank of Gabor functions $g_{s,k}(x,y)$ is generated by rotating and scaling the Gabor function of Eq. (3.3) as follows:

$$g_{s,k}(x,y)=\alpha^{-s}g(x',y'), \tag{3.6}$$

with

$$\begin{aligned} x' &= \alpha^{-s}\left(x\cos\theta_k+y\sin\theta_k\right), \\ y' &= \alpha^{-s}\left(-x\sin\theta_k+y\cos\theta_k\right). \end{aligned} \tag{3.7}$$

For given numbers of S scales and K orientations, α^{-s} is a scaling factor that guarantees the energy of the filter to be independent of the scale, $s=0,\ldots,\mathrm{S}-1$, while angles θ_k are computed as

$\theta_k = \frac{k\pi}{K}$, $k = 0, \ldots, K-1$. Figure 3.3 shows an example of a Gabor filter bank composed by kernels at $S = 5$ scales and $K = 8$ orientations.

Now, given an image patch $I(x, y) \in \mathbb{R}^{M \times N}$, its convolution with the Gabor filter bank gives raise to $S \times K$ filter responses such that:

$$W_{s,k}(x, y) = I(x, y) * g_{s,k}(x, y). \tag{3.8}$$

For a given scale s and orientation k, the corresponding attributes are computed as the mean and standard deviation of the absolute value filter response over the image patch:

$$\begin{aligned} \mu_{s,k} &= \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |W_{s,k}|, \\ \sigma_{s,k} &= \sqrt{\frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left(|W_{s,k}| - \mu_{s,k}\right)^2}, \end{aligned} \tag{3.9}$$

So that the final feature vector $\mathbf{x}$ is created by concatenating the resulting attributes computed at every scale and orientation:

$$\mathbf{x} = \left[\mu_{0,0}, \sigma_{0,0}, \mu_{0,1}, \sigma_{0,1} \ldots, \mu_{S-1,K-1}, \sigma_{S-1,K-1}\right]. \tag{3.10}$$

The motivation behind our choice to employ Gabor filters lies in the fact that it is an efficient way to perform multi-scale and multi-orientation analysis (the Gabor filter bank can be indeed pre-computed), allowing at the same time to obtain a compact feature vector. Besides, due to the absolute value of the filter response in the creation of the attributes, it does not matter for the final feature whether a person appears dark on bright background or vice-versa, allowing for a unified representation of people appearance as long as their responses to the different scales and orientations are consistent.

Gabor filter banks have been already used in conjunction with LBP features, e.g. in [255] for face recognition, firstly reducing their dimensionality through Principal Component Analysis (PCA) and then fusing them together at feature-level using kernel discriminative common vector methods, or in [297] for person re-identification exploiting a region covariance descriptor. Gabor filters have been used also in conjunction with HOG, e.g. in [46] where the HOGG descriptor is proposed by firstly applying a Gabor preprocessing that helps to emphasize the human body shape and improves the posterior gradient accumulation done by the HOG algorithm. However, the size of the resulting feature vector increases dramatically.

3.2.4 DAISY

DAISY is a more recent descriptor which has gained popularity particularly in the field of wide-baseline stereo matching [258], while at the same time being designed for an effective dense computation. Similarly to SIFT and Gradient Location and Orientation (GLOH) [186] descriptors, DAISY involves the computation of histograms of gradient locations and orientations. The differences among them lie in two aspects. Firstly, DAISY replaces the weighted sums of gradient norms used in SIFT and GLOH by convolutions of gradients in specific directions with several Gaussian filters that can be computed very quickly. Secondly, as GLOH, DAISY uses a circular neighborhood configuration instead of the rectangular grid used by SIFT, omitting however the PCA-based dimensionality reduction performed by GLOH.

For a given input image, H orientation maps are firstly computed and then convolved several times with Gaussian kernels of different Σ, on Q concentric layers having T circles centered on sampled locations. Then, histograms of orientations are derived from the central location and from all the sampled locations at every layer, and finally concatenated.

More specifically, for the given input image I, orientation maps G_o are firstly computed, one for each quantized direction o, $1 \leq o \leq H$. They are formally defined as:

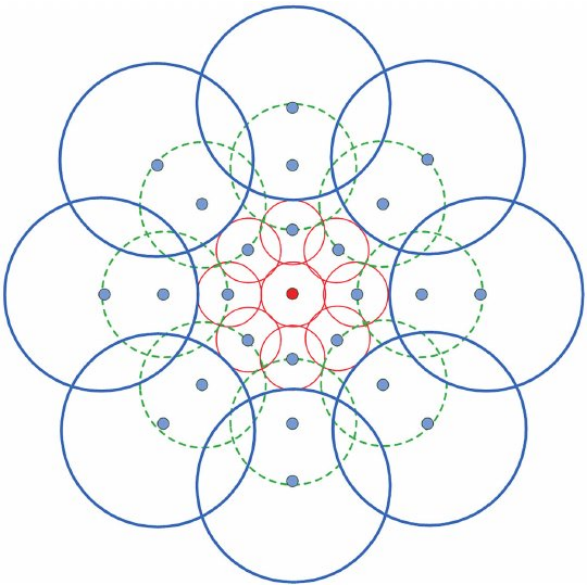


Figure 3.4: DAISY descriptor structure.

$$\mathrm{G}_o = \left(\frac{\partial \mathrm{I}}{\partial o}\right)^+, \quad 1 \leq o \leq \mathrm{H}, \tag{3.11}$$

where o is the orientation of the derivative and the "+" sign means that only positive values are kept in order to preserve the polarity of the intensity changes, such that $(a)^+ = \max(a, 0)$.

Each orientation map, representing thus the image gradient norm for that direction at all pixel locations, is then convolved several times with Gaussian kernels of different standard deviation Σ values, in order to obtain the convolved orientation maps. For given Gaussian kernel G^{Σ} and orientation o:

$$\mathrm{G}_o^{\Sigma} = \mathrm{G}^{\Sigma} * \left(\frac{\partial \mathrm{I}}{\partial o}\right)^+. \tag{3.12}$$

The efficiency of the DAISY descriptor is particularly evident in this computation. Indeed, since Gaussian filters are separable by nature, convolutions can be implemented very efficiently and in particular convolutions with large Gaussian kernels can be obtained by several consecutive convolutions with small ones, reducing the computational burden.

Figure 3.4 shows an example of the circular structure of the DAISY descriptor (which looks like a flower, hence the name), computed around the central pixel which is indicated by the filled, red dot. The other filled blue dots represents the sampled locations, namely T equally spaced points to cover all directions, in Q concentric layers. Then, each circle on the image contains one histogram vector of orientations which is built from the convolved orientation maps in different gradient directions for each region, where the amount of Gaussian smoothing is proportional to the radius of the circle, which increases in the outer rings.

For a given sampling location (x, y) in the image, its responses to the various convolved orientation maps are accumulated in a H-dimensional histogram such that:

$$h^{\Sigma_i}(x, y) = \left[\mathrm{G}_1^{\Sigma_i}(x, y), \ldots, \mathrm{G}_{\mathrm{H}}^{\Sigma_i}(x, y)\right]^{\mathsf{T}}, \tag{3.13}$$

where Σ_i represents the standard deviation of the Gaussian kernel for the particular layer, $i = 1, \ldots, \mathrm{Q}$. In the original version of the paper, for the objective of wide-baseline stereo matching, the various histogram vectors are independently normalized to the unit form (noted $\tilde{h}^{\Sigma_i}(x, y)$), but the authors suggest that different normalization schemes could be applied on the basis of the considered application.

Table 3.2: DAISY parameters.

Symbol	Description
R	Radius, i.e. distance from the center pixel to the outer most sampling point.
Q	Number of concentric layers.
T	Number of histograms in a single layer.
H	Number of considered orientations, i.e. histogram bins.

Finally, the DAISY descriptor of a given pixel (x, y) can be composed by concatenating the previously computed (normalized) vectors, starting from the central point and then considering all the sampling points in the outer rings.

Table 3.2 gives a summary of the various parameters involved in the DAISY descriptor. The number of total histogram used in the descriptors can be easily computed as $S = QT + 1$. Since the total descriptor is composed by the concatenation of the various histograms, its total size is SH.

There are several motivations behind our choice to employ the DAISY descriptor. As previously stated, descriptors which are usually computed sparsely for stereo matching applications have been shown to perform well even when computed densely and coupled with a supervised learning algorithm, thanks to their intrinsic robustness to scene variations and illumination changes. Among these descriptors, DAISY has been proposed to be computed *efficiently* in the dense setting, first reason why we have chosen it. Then, its Gaussian smoothing, together with the sampling overlap, naturally enforce spatial consistency which is indeed important for pixel-wise detection applications, showing more robustness to partial occlusions and ensuring a smoothly changing descriptor for neighbor pixels. The amount of Gaussian smoothing is then proportional to the radius of each concentric circle, giving rise to larger Gaussian kernels in the outers rings. For this reason, DAISY appears well suited for our application, as we benefit from a finer description of the contour of the head and a coarser description moving away from it, removing noise in the surroundings yet being able to consider spatial context information. Finally, the use of a circular sampling grid is interesting for many reasons. Firstly, it naturally fits our detection target shape (head). Then, it has been shown to have better localization properties with respect to the traditional grid used by SIFT, as stated in [186]. Lastly, combining an isotropic Gaussian kernel with a circular grid makes DAISY descriptor naturally robust to rotational perturbations.

To our knowledge, this descriptor has not been previously used for head detection in crowds.

3.3 Single-descriptor SVM learning

3.3.1 SVM learning overview

Figure 3.5 gives an overview of the traditional learning procedure using SVM. Firstly, a training set containing positive and negative training samples is collected. Then, for each sample location, a window around it is considered and a feature vector is obtained applying the chosen descriptor. Feature vectors are then fed to SVM which on the basis of the chosen kernel function returns a *model*, which encapsulates the optimal decision boundary learned from the various training samples and the chosen hyper-parameter C to control the generalization ability.

After the training, the learned model can be applied to classify unseen samples, i.e. in our application a dense classification over the test image pixels through a sliding window is performed. Since SVM is a binary non-probabilistic classifier, its decision is performed using Eq. (2.29) providing thus just a binary label regarding the class of the test sample, giving rise to the final detection image.

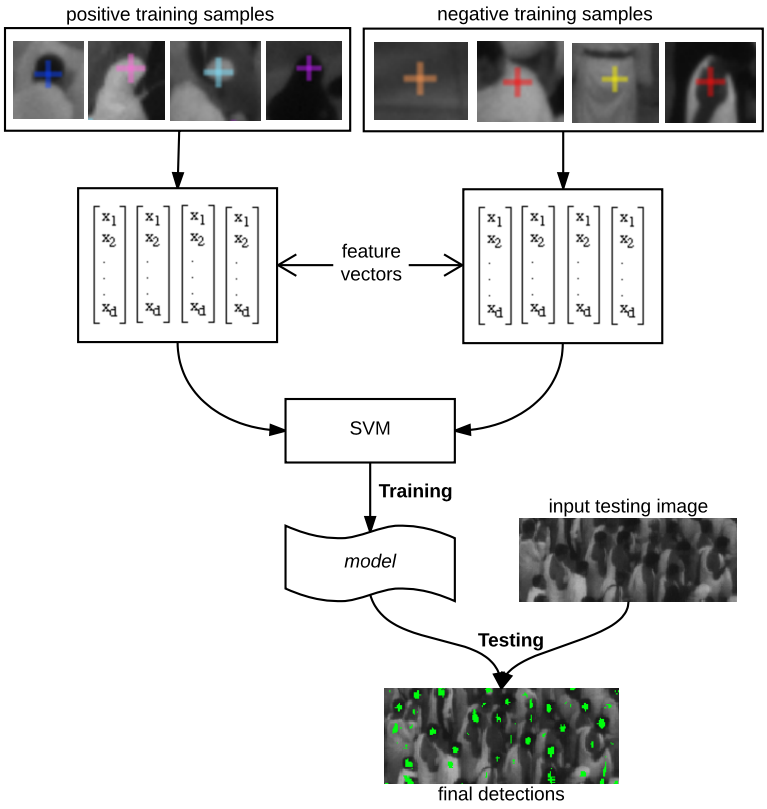


Figure 3.5: SVM learning overview.

3.3.2 SVM score calibration

In order to obtain class probabilities from SVM uncalibrated scores, i.e. sample distances to the hyperplane margin, a well established method proposed by Platt [211] consists in approximating the posterior probability by learning the optimal parameters of a logistic sigmoid function (cf. Sec. 2.2.3), relying on a calibration set independent from the training data.

In particular, given the training samples $\mathbf{x}^{(i)} \in \mathbb{R}^n$, $i = 1, ..., m$, labeled by $y^{(i)} \in \Omega = \{+1, -1\}$, defined as feature vectors derived from a head detector, the binary SVM computes a decision function such that $h(\mathbf{x}) = \text{sign}\big(f(\mathbf{x})\big)$ (cf. Eq. (2.29)) is used to predict the label of the unseen test sample $\mathbf{x}$. In order to obtain class probability $\mathrm{P}(y = 1|\mathbf{x})$, the method proposed by Platt [211] approximates the posterior probability by learning a logistic sigmoid function:

$$\mathrm{P}(y = 1|\mathbf{x}) \approx \sigma_{\lambda_0,\lambda_1}(f) = \frac{1}{1 + e^{\lambda_0 f + \lambda_1}}. \tag{3.14}$$

The optimal parameters $(\lambda_0^*, \lambda_1^*)$ are then determined by solving a regularized maximum likelihood problem, using a calibration set different from the SVM training one, or performing a n-fold cross-validation on the training set. Moreover, in order to cope with possibly imbalanced calibration sets, target values $y^{(i)} \in \Omega$ are replaced by $y^{(i)} \in \{y_+, y_-\}$ such that:

$$y_+ = \frac{\mathrm{N}_+ + 1}{\mathrm{N}_+ + 2}, \quad y_- = \frac{1}{\mathrm{N}_- + 2}, \tag{3.15}$$

where N_+ and N_- are the numbers of positive and negative samples in the calibration set respectively.

We have chosen to rely on a calibration set different from the training one, in order to avoid possible overfitting problems. Besides, our calibration set is an image patch,so that the calibration is performed on the real data distribution (this influences particularly the bias intercept λ_1). For each pixel of this image patch, its target value is directly derived from the related ground-truth,

without considering pixels which are on the contour of the heads to avoid confusion in the label assignment. Thus, starting from the ground-truth binary map indicating head centers, a dilation with a circular structuring element of radius r_1 is performed and the obtained 1-valued pixels are assigned to the positive class. In the same way, a dilation with $r_2 > r_1$ allows to assign the negative label to all the 0-valued pixels. Note that r_1 and r_2 values have been found empirically and should be adapted to the considered training set.

Platt suggested to use the Levenberg-Marquardt algorithm to optimize the parameters, but a Newton algorithm was later proposed and proven to be numerically more stable [166], so we adopted it. In the same way, the authors suggested a reformulation of the minimization of the negative log-likelihood problem in order to avoid numerical problems.

3.4 Single-descriptor results

3.4.1 SVM settings

We rely on a balanced training set composed by 2000 samples, i.e. 1000 positive and 1000 negative points, carefully manually selected in order to span as much as possible across the sample characteristics while at the same time remaining focused on the center of the heads, as explained in Appendix A.

Platt's calibration is performed on the basis of a calibration set to obtain probabilistic outputs. Considering that a head diameter in our dataset spans between 6 and 12 pixels, $r_1 = 2$ and Kernels adapted to each descriptor has been chosen in order to exploit their distinctive features at maximum. Cross-validation is performed to set the parameters of each descriptor and kernel, as well as to find the best C parameter (cf. Eq. (2.31)), and they are detailed in the following:

- *HOG* – We compute HOG descriptors in 24×24 windows, in order to include information about the immediate surrounding of the actual head while at the same time avoiding other targets. A L2-hys normalization is applied for each block. For learning, we rely on the HIK. C parameter is set to 0.01;

- *LBP* – Following the idea of [3], we subdivide the image in small regions from which histograms are extracted and then concatenated, in order to enhance the locality of the LBP. A $\mathrm{LBP}^{riu2}_{1,8}$ is used over 12×12 windows subdivided into four 6×6 blocks. The choice of the window size is sensitive, as larger windows result in wide detections, overflowing the actual heads. Stride between blocks has also been tested but it does not provide consistent improvement. Following the example of [3] which employs on a χ^2 distance as a dissimilarity measure, we rely on a χ^2 kernel function which has been shown to be positive definite and suited for data generated from histograms [162]. C parameter is set to 0.1;

- *Gabor Filter Banks (GABOR)* – To build the final feature vector, instead of just concatenating the raw responses of every filter in the bank, we subdivide the window around each pixel in several blocks, and then we compute their first and second order statistics. We use a Gabor filter bank of 5 scales and 4 orientations; a high number of scales is essential to obtain good results, while increasing the number of orientations does not provide an effective gain in performance. For each Gabor filter response image, we compute and concatenate mean and standard deviation over 4×4 blocks on 16×16 windows to obtain the GABOR feature vector. Then, a RBF kernel is considered for learning. C parameter is set to 10, while the σ parameter of the RBF kernel is set to 2;

- *DAISY* – We use a radius R = 8 from the center to the outer ring, with Q = 3 number of layers and T = 8 histograms of H = 8 bins at each layer. As for the HOG, the HIK is employed for SVM classification. C parameter is set to 1.

Table 3.3: Confusion matrix definition

	Ground-truth Positive	Ground-truth Negative
Predicted Positive	*True Positive (TP)*	*False Positive (FP)*
Predicted Negative	*False Negative (FN)*	*True Negative (TN)*

3.4.2 Evaluation method

Since our primary objective is pedestrian detection in the context of high-density crowds, we choose to perform an object-level analysis of the results. To this extent, we rely on Precision-Recall (PR) curves that are able to show, for each possible threshold value, the trade-off between precision and recall values which are defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.16}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.17}$$

where TP, FP and FN are defined by the confusion matrix shown in Table 3.3.

Precision is therefore the fraction of relevant instances among the retrieved instances, while Recall is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. Traditionally, the Area Under Precision-Recall Curve (AUPRC) is employed to evaluate the performance of an algorithm from its PR-curve.

Non-Maximum Suppression (NMS) is applied to obtain the detections at every threshold, setting the radius of a head to $r = 3$, with $2r + 1$ minimum distance between two maxima (head centers) in order to avoid overlapping detections.

3.4.3 Results

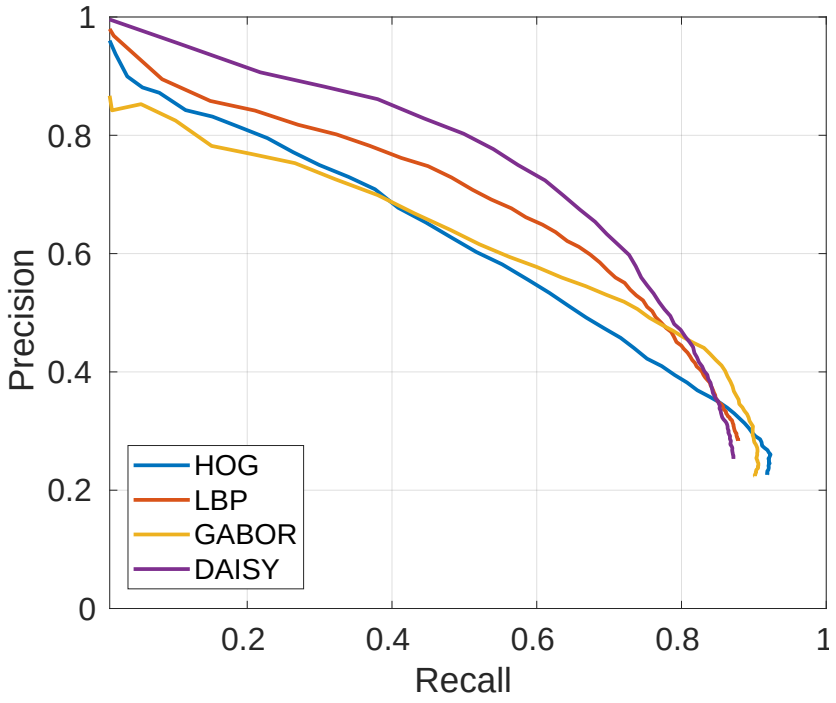


Figure 3.6: PR-curves of SVM trained with the different descriptors and kernels.

Table 3.4: Precision-Recall Break Even Point and Area Under Precision-Recall Curve with the different descriptors.

	HOG	LBP	GABOR	DAISY
PRBEP	0.57	0.63	0.59	0.67
AUPRC	0.58	0.63	0.58	0.68

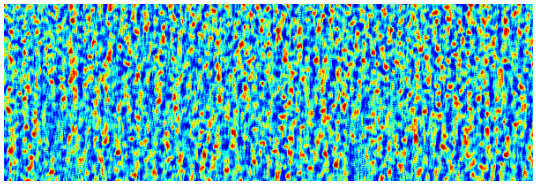
Figure 3.6 shows the PR-curves of SVM trained with different features obtained through the inspected descriptors, namely HOG, LBP, GABOR and DAISY. As it is possible to notice, HOG and GABOR provide lower precision values but are able to reach higher values of recall, whereas LBP and DAISY provide better precision but are not able to reach the same levels of recall as the other two descriptors. Overall, DAISY provides the best results and reveals to be a good choice for the given task, even though it has been proposed in the context of a different application (stereo matching), thanks to its intrinsic robustness to illumination and scene variation changes, along with its Gaussian smoothing that enforces spatial consistency.

Figure 3.7 shows the results of the dense classification performed on a testing image patch after SVM training with the different inspected descriptors. Results are shown in terms of colormaps of the probabilistic output map obtained after the calibration procedure (in the first column), and detections at the Precision-Recall Break Even Point (PRBEP) threshold, which is a useful operative threshold value corresponding to the threshold for which precision is equal to recall (cf. Table 3.4).

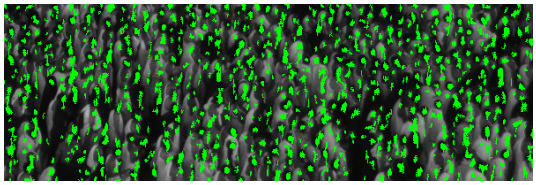
In line with the tendency already observed in the PR-curves corresponding to the different descriptors, HOG and GABOR provide noisier probability output maps and smaller output detections, which corresponds to lower values of precision (more false positives are generally present) while at the same time higher recall values (more heads are detected, even though with small blobs). These two descriptors seem to perform poorly when the background is cluttered with noisy edges. Conversely, LBP and DAISY provide larger and rougher results, which reflects in higher precision values since the number of false positives is smaller. LBP can indeed filter out noise using the concept of uniform pattern, while DAISY can provide very smooth detections, due to the Gaussian-based spatial sampling. Finally, note that HOG provides quite localized detections employing the largest window size, while LBP outputs large detection blobs relying on the smallest window size.

The results obtained using single descriptors however, looking also at the PRBEP and AUPRC values in Table 3.4, are not sufficient to perform a satisfactory analysis of the scene. They are nevertheless useful in that they underline the complementarity among the chosen descriptors, which is a desirable property in view of the design of a fusion strategy among them aimed at leveraging their peculiarities.

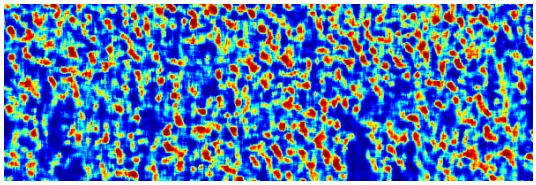
(a) Patch image



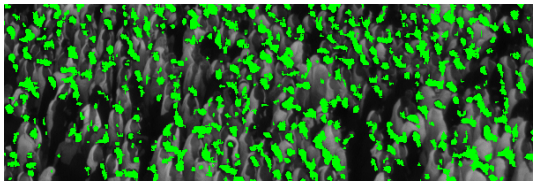
(b) Probabilistic output map - HOG



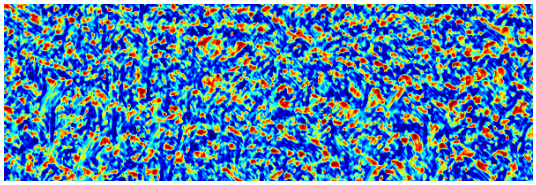
(c) Detection map - HOG



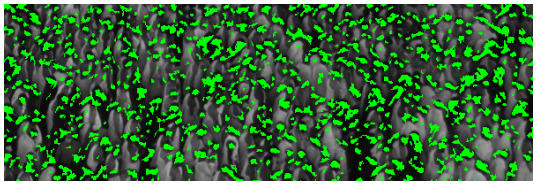
(d) Probabilistic output map - LBP



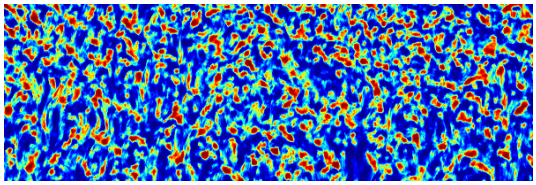
(e) Detection map - LBP



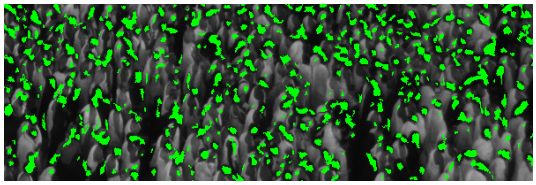
(f) Probabilistic output map - GABOR



(g) Detection map - GABOR



(h) Probabilistic output map - DAISY



(i) Detection map - DAISY

Figure 3.7: Visualization of the results of the dense classification performed on a testing image patch after SVM training with the different inspected descriptors, in terms of colormaps of the probabilistic output map obtained after the calibration procedure (first column) and detections at PRBEP threshold.

Chapter 4

Taking into account imprecision with Belief Function Framework

Contents

4.1 Motivation

In difficult applications such as high-density crowd analysis, an SVM model trained with features obtained through a single descriptor may not be enough to reach satisfactory levels of performance. Besides, the traditional learning method do not take into account possible imprecision that may exist due to the specific classifier and/or data used in the training and calibration processes. We therefore propose an ensemble composed of several SVM trained with different, complementary yet independent descriptors. To handle both the uncertainty provided by the classification and the related imprecision we propose a solution in the context of Belief Function Theory (BFT). After an introduction about fundamental concept of BFT, we present the proposed fusion strategy among the ensemble members.

4.2 Belief Function Theory

The Belief Function Theory (BFT) [59, 247], also known as Dempster-Shafer Theory (DST) [239] or evidence theory, is a formal framework for reasoning with partial (uncertain, imprecise) information or knowledge, representing thus a generalization of probability theory. It is also directly connected to possibility theory [291] and fuzzy sets [13, 292].

Solving a real-world problem in the context of Belief Function framework typically involves three different steps:

1. Representing and modeling the available pieces of information using BF;
2. Manipulating and combining the resulting BFs;
3. Making decisions based on the computation of some BF.

While many tools have been proposed for the two latter steps, including several combination rules and decision functions, modeling the initial (possibly partial, unreliable and/or imprecise) information using BFT is still challenging for many applications. We will therefore propose to this extent a BF definition for our ensemble of SVM classifiers.

We will now present an overview of BFT, with emphasis on the notions and operators useful for our study, before explaining the proposed information modelization along with the motivations behind it.

4.2.1 Belief representation

To handle both uncertainty and imprecision, Belief Functions (BFs) are defined on a larger hypothesis set than in the case of the probabilistic framework. Specifically, if Θ denotes the *discernment frame*, i.e. the set of mutually exclusive (*singleton*) hypotheses, BFs are defined on the set of the subsets of Θ, noted 2^Θ in reference to its number of elements: $2^{|\Theta|}$, where $|\Theta|$ is the cardinality of Θ. The power set 2^Θ is thus the set of all the possible disjunctions of the set of singleton hypotheses in Θ.

4.2.1.1 Mass function

Definition 4.2.1 *A **mass function** m^Θ, specifying a Basic Belief Assignment (BBA), is a function over the power set 2^Θ, $m^\Theta : 2^\Theta \rightarrow [0,1]$, which satisfies:*

$$\sum_{A \subseteq \Theta} m^\Theta(A) = 1. \tag{4.1}$$

Note that for conciseness the superscript indicating the discernment frame Θ can be omitted when there is no ambiguity about such set.

Definition 4.2.2 *A **focal element** (or focal set) of m is a subset $A \subseteq \Theta$ such that $m(A) > 0$.*

Definition 4.2.3 *A mass function is said to be **normalized** if:*

$$m(\emptyset) = 0, \tag{4.2}$$

where $\emptyset$ is the empty set representing the null hypothesis of the given discernment frame.

Normalization property is usually applied under the *closed-world assumption*, in which Θ is defined on an exhaustive set of hypotheses. Relaxing this hypothesis (*open-world assumption*) allows us to consider the existence of *unnormalized* BBAs, where the mass on the empty set can be interpreted as the degree of support of the hypothesis that the solution lies outside Θ.

Definition 4.2.4 *A mass function is said to be **vacuous** if it has Θ as unique focal element, i.e. $m(\Theta) = 1$.*

A vacuous mass function represents thus the total ignorance.

Definition 4.2.5 *A mass function is said to be **categorical** if it has only one focal element:*

$$\exists!\ A \subset \Omega \ s.t.\ m(A) = 1 \tag{4.3}$$

A categorical mass function conveys certain but possibly imprecise information.

Definition 4.2.6 *A mass function is said to be **Bayesian** if it has only singleton hypotheses as focal elements:*

$$\forall\ \mathrm{A} \subseteq \Omega\ s.t.\ m(\mathrm{A}) > 0 \Rightarrow |\mathrm{A}| = 1. \tag{4.4}$$

Bayesian BBAs represent thus precise information, as the masses on all the disjunctions representing partial ignorances or total ignorance (i.e. $m(\Theta)$) are null. In this sense, BFT is an extension of the probabilistic theory: a probability distribution boils down to a particular case of a mass function, when information is modeled through a Bayesian BBA.

4.2.1.2 Belief and Plausibility functions

Belief (Bel) and Plausibility (Pl) functions are in one-to-one relationship with m.

Definition 4.2.7 *Given a mass function m, **Belief** function is defined $\forall \mathrm{A} \subseteq \Theta$ as:*

$$Bel(\mathrm{A}) = \sum_{\mathrm{B} \subseteq \mathrm{A}, \mathrm{B} \neq \emptyset} m(\mathrm{B}). \tag{4.5}$$

Definition 4.2.8 *Given a mass function m, **Plausibility** function is defined $\forall \mathrm{A} \subseteq \Theta$ as:*

$$Pl(\mathrm{A}) = \sum_{\mathrm{B} \cap \mathrm{A} \neq \emptyset} m(\mathrm{B}). \tag{4.6}$$

The Belief of A can be interpreted as the degree to which the evidence *does not support* $\overline{\mathrm{A}}$, whereas the Plausibility of A can be seen as the degree to which the evidence *is consistent with* A.

It is important to notice that Bel and Pl functions may also be interpreted as upper and lower bounds of probability [239] as they check the duality property: $\forall \mathrm{A} \in 2^{\Theta}$, $\mathrm{Pl}(\mathrm{A}) = 1 - \mathrm{Bel}(\overline{\mathrm{A}})$, where $\overline{\mathrm{A}}$ denotes the complement of A with respect to Θ (equivalently, in the open-world assumption the property becomes $\mathrm{Pl}(\mathrm{A}) = \mathrm{Pl}(\Theta) - \mathrm{Bel}(\overline{\mathrm{A}})$).

From the definition $\mathrm{Pl}(\mathrm{A}) \geq \mathrm{Bel}(\mathrm{A}), \forall \mathrm{A} \subseteq \Theta$. The equality holds in the case of Bayesian mass functions, where the two representations are equivalent to a probability measure. The interval $[\mathrm{Bel}(\mathrm{A}), \mathrm{Pl}(\mathrm{A})]$ represents thus the amount of imprecision associated to the related BBA, which can span between 0 (for a Bayesian BBA) and 1 (for a vacuous BBA).

4.2.1.3 Discounting

The discounting operator can be employed in order to model the reliability of a source of information [239]. Over the years, several discounting procedures have been proposed (contextual discounting [184], contextual reinforcement [209] among the others), but the simplest one consists in reallocating some mass to Θ (representing ignorance) by means of a discounting factor.

Definition 4.2.9 *Given a discounting factor $\alpha \in [0,1]$, the **discounted mass function** ${}^{\alpha}m$ is defined as:*

$$\begin{aligned} {}^{\alpha}m(\mathrm{A}) &= \alpha m(\mathrm{A}), \quad \forall \mathrm{A} \subset \Theta \\ {}^{\alpha}m(\Theta) &= \alpha m(\Theta) + 1 - \alpha \end{aligned} \tag{4.7}$$

This operation has the effect of weakening all masses and reinforcing $m(\Theta)$ (i.e. the mass on the largest disjunction representing total ignorance). When $\alpha = 0$, the information is considered totally not reliable, and the resulting mass function is vacuous, whereas when $\alpha = 1$, the information is considered completely reliable and the mass function is not affected.

4.2.2 Belief Functions combination

Several combination rules have been proposed over the years. We report here the definition of the ones that will be used in the context of this work.

Definition 4.2.10 *Given two mass functions m_1 and m_2 derived from two independent pieces of evidence, the **conjunctive rule** of combination is defined as:*

$$m_1 \textcircled{\cap} m_2\,(\mathrm{A}) = \sum_{\mathrm{B}\cap\mathrm{C}=\mathrm{A}} m_1(\mathrm{B})\,m_2(\mathrm{C}),\ \ \forall \mathrm{A} \in 2^{\Theta}. \tag{4.8}$$

This combination, also referred to as *unnormalized Dempster's rule*, leads to a possibly non-null mass on the empty set, which corresponds to the *degree of conflict*, K. Specifically, the *degree of conflict* K between two mass functions is:

$$\mathrm{K} = \sum_{\mathrm{B}\cap\mathrm{C}=\emptyset} m_1(\mathrm{B})\,m_2(\mathrm{C}). \tag{4.9}$$

The normalized version of the conjunctive combination can be then defined, and it is often referred to as Dempster's rule.

Definition 4.2.11 *Given two mass functions m_1 and m_2 derived from two independent pieces of evidence, the **Dempster's rule** of combination is defined as:*

$$\begin{aligned} &m_1 \oplus m_2\,(\mathrm{A}) = \frac{1}{1-\mathrm{K}} \sum_{\mathrm{B}\cap\mathrm{C}=\mathrm{A}} m_1(\mathrm{B})\,m_2(\mathrm{C}), \forall \mathrm{A} \in 2^{\Theta} \setminus \{\emptyset\}, \\ &m_1 \oplus m_2\,(\emptyset) = 0. \end{aligned} \tag{4.10}$$

The Dempster's rule both in its normalized and unnormalized form comes with properties of *commutativity* and *associativity*. Moreover, when applied to Bayesian mass functions, it is equivalent to the probabilistic product rule, stressing once again the generalization performed by the evidential framework over the probabilistic one.

4.2.3 Decision making

Finally, after BBAs definition and combination, a decision can be taken. Several rules have been proposed in the literature. Most popular ones only consider singleton hypotheses (in order to avoid ambiguous decision) and are based on maximum of plausibility, credibility, or pignistic probability [247].

Pignistic probability in particular is one of the most commonly used, since it allows us to give a probabilistic interpretation to the BBA, by transforming a mass function into a probability measure with a *pignistic transformation*:

$$\forall \mathrm{A} \in \Theta,\ \mathrm{BetP(A)} = \frac{1}{1-m(\emptyset)} \sum_{\mathrm{B}\supseteq\mathrm{A}} \frac{m(\mathrm{B})}{|\mathrm{B}|}. \tag{4.11}$$

Alternatively, more recently the Plausibility Probability function [41] (Pl_P), and its unnormalized version called *contour function*, have been proposed. Pl_P also provides a probabilistic interpretation of the mass function, while being more consistent with the Dempster's rule, and is defined as:

$$\forall \mathrm{A} \in \Theta,\ \mathrm{Pl\_P(A)} = \frac{\mathrm{Pl(A)}}{\sum_{\mathrm{B}\in\Theta} \mathrm{Pl(B)}}. \tag{4.12}$$

4.3 Proposed BBA definition

In the context of SVM-based high-density crowd pedestrian detection, we consider that *imprecision* can arise in two different and complementary ways: in the derivation of posterior probability values from SVM decision scores, and later, from the spatial layout of the detections in the output image space.

We propose a mass allocation that is robust to possible imprecision of the calibration functions while at the same time taking into account the information coming from neighboring pixels in the image space. Besides, it allows for an amount of discounting that is different at every pixel of the classifier's output map, and it is not only a constant value that merely reflects the reliability of the detector, as is often done (e.g. [141]).

More specifically, let us explain better the origin and the modeling of these two different types of imprecision into the definition of a BBA.

Note that in our case, denoting by H and $\overline{\text{H}}$ the two singleton hypotheses, *"Head"* and *"Not Head"* respectively, the discernment frame is $\Theta = \left\{\text{H}, \overline{\text{H}}\right\}$, and the set of hypotheses is $2^{\Theta} = \left\{\emptyset, \text{H}, \overline{\text{H}}, \Theta\right\}$. Moreover, Belief and Plausibility boil down to: $\forall \text{A} \in \Theta$,

$$\begin{aligned} \text{Bel}(\text{A}) &= m(\text{A}), & (4.13)\\ \text{Pl}(\text{A}) &= m(\text{A}) + m(\Theta). & (4.14) \end{aligned}$$

4.3.1 BBA definition based on calibrated scores

The first source of imprecision that we take into account is related to the SVM score calibration process. Indeed, as explained in Sec. 3.3.2, in order to obtain class probabilities from SVM uncalibrated scores, a logistic sigmoid function $\sigma_{\lambda_0,\lambda_1}$ is learned (cf. Eq.(3.14)) through an optimization process that allows us to obtain the optimal sigmoid parameter pair $(\lambda_0^*, \lambda_1^*)$ on the basis of a calibration set different from the SVM training one.

Our system relying on multiple SVM classifiers, N different sigmoid parameter configurations are learned: $(\lambda_0^*, \lambda_1^*)_n$ refers to the sigmoid parameters estimated for classifier n, with $n = 1, \ldots, \text{N}$.

For each different test sample $\mathbf{x}$, given its score s_n, namely its distance to the hyperplane boundary defined by classifier n, we now define an associated Bayesian BBA $m_n^{\mathscr{B}}$ (i.e., BBA having only singleton focal elements), from the posterior probability given by its score calibration procedure:

$$\begin{aligned} m_n^{\mathscr{B}}(\text{H}) &= \sigma_{(\lambda_0^*,\lambda_1^*)_n}(s_n),\\ m_n^{\mathscr{B}}(\overline{\text{H}}) &= 1 - \sigma_{(\lambda_0^*,\lambda_1^*)_n}(s_n), & (4.15)\\ m_n^{\mathscr{B}}(\Theta) &= 0,\\ m_n^{\mathscr{B}}(\emptyset) &= 0. \end{aligned}$$

This initial Bayesian BBA is only able to model the uncertainty about the class the sample belongs to, relying on a calibration procedure that is assumed to be precise.

However, in difficult settings such as our application, a robust estimation of the sigmoid parameters is almost impossible to achieve, and few changes in the calibration set (cardinality or in the samples within it) can cause the sigmoid to appear very different. In presence of a steep transition between the two classes particularly, even a slight shift of the sigmoid may induce very different probability values and possibly different decisions for quite numerous samples, especially in case of strong overlap between the two classes. Now, with Belief Function framework we can naturally take into account the imprecision inherent to the sigmoid learning process. Instead of deriving a simple probabilistic value through logistic regression, we aim at associating a BBA to each unlabeled sample directly from its score and from the estimated sigmoid (from calibration process).

Xu et al. [285] proposed to extend the logistic calibration to derive a BBA that takes into account the number of samples per score value for calibration process. Such an approach is suitable especially when the number of samples is small and when there is no overlapping between the scores of the two considered classes. Otherwise, as shown in [150], in such difficult types of applications, it is hard for SVM to find a very large margin between the two classes and there can be a consistent overlap between samples with different labels for the same score. However, since the number of samples per score would be high, we would paradoxically not assign a high value of imprecision to them.

Then, as an alternative we consider the BBA allocation proposed by [17]. It relies on the observation that (fuzzy) erosion and dilation (respectively opening and closing) are dual with respect to complementation, and they can be interpreted as Belief and Plausibility functions: given a BBA m_0 derived from the output of a classifier, the following property holds: $\forall A \in 2^{\Theta}$,

$$\mathrm{Pl}(A) = 1 - \mathrm{Bel}(\overline{A}) \quad \leftrightarrow \quad \delta_v(m_0(A)) = 1 - \mathscr{E}_v(m_0(\overline{A})), \tag{4.16}$$

$$\leftrightarrow \quad \phi_v(m_0(A)) = 1 - \gamma_v(m_0(\overline{A})), \tag{4.17}$$

where δ_v and $\mathscr{E}_v$ are the dilation and erosion operators respectively, with structuring element v, while ϕ_v and γ_v are the closing and opening operators.

The amount and shape of the possible imprecision is thus modeled through a structuring element. Now, we propose to interpret the erosion operator as a discounting operator, in the sense that the obtained BBA will be less committed. Indeed, when applying erosion to $m_0(A)$ to derive $\mathrm{Bel}(A), \forall A \in \left\{H, \overline{H}\right\}$, the mass on Θ is increased by the sum of the differences between initial values and eroded values: $m_0(\Theta) = m_0(A) - \mathscr{E}_v(m_0(A)) + m_0(\overline{A}) - \mathscr{E}_v(m_0(\overline{A}))$.

In our case, the initial (Bayesian) BBAs $m_n^{\mathscr{B}}$ are provided by the learned sigmoid associated to each classifier n, through the probabilistic calibration.

Then, the application of erosion and dilation operations to this sigmoid, with a structuring element of width w defined as a segment line in the score domain, allows for the derivation of two new sigmoid functions that are interpreted as lower and upper bounds of probability with respect to the learned sigmoid, i.e. Bel and Pl functions of the obtained BBA. Due to the fact that we consider a flat structuring element and to the intrinsic monotonically increasing profile of the sigmoid function, considering classifier n, it is possible to easily derive:

$$\mathrm{Bel}_n(H) = \sigma_{(\lambda_0^*, \lambda_1^*)_n}(s_n - \frac{w}{2}), \tag{4.18}$$

$$\mathrm{Pl}_n(H) = \sigma_{(\lambda_0^*, \lambda_1^*)_n}(s_n + \frac{w}{2}). \tag{4.19}$$

Figure 4.1 shows an example of a sigmoid function learned on the calibration set, as well as the two derived sigmoid functions (for two structuring elements of different widths), that represent Bel and Pl functions and provide the interval of imprecision.

The interval between Bel and Pl functions embeds thus the amount of imprecision in the calibration step we have to cope with. It takes low values for points far from the hyperplane boundary for which the decision is already pretty sure, whereas on the contrary it takes high values in the area near to the hyperplane margin, where even a slight difference in the parameters of the sigmoid can change the decision. Then, previous BBA allocation allows us to model the fact that the calibration function may be not perfectly fitted due to the difficulty in the definition of a robust calibration set and to allocate large values of imprecision to the samples having their correspondent score within the SVM margin, in the overlapping area.

Table 4.1 proposes a toy example to illustrate the considered BBA allocation based on SVM scores. Let us suppose that for a given classifier the sigmoid's optimal parameters have been found to be $\lambda_0^* = -2$ and $\lambda_1^* = -0.05$ through Platt's calibration based on logistic regression on the calibration set. Then, considering two different test samples $\mathbf{x}_1$ and $\mathbf{x}_2$, such that $s_{\mathbf{x}_1} = -0.5$ and $s_{\mathbf{x}_2} = +2$

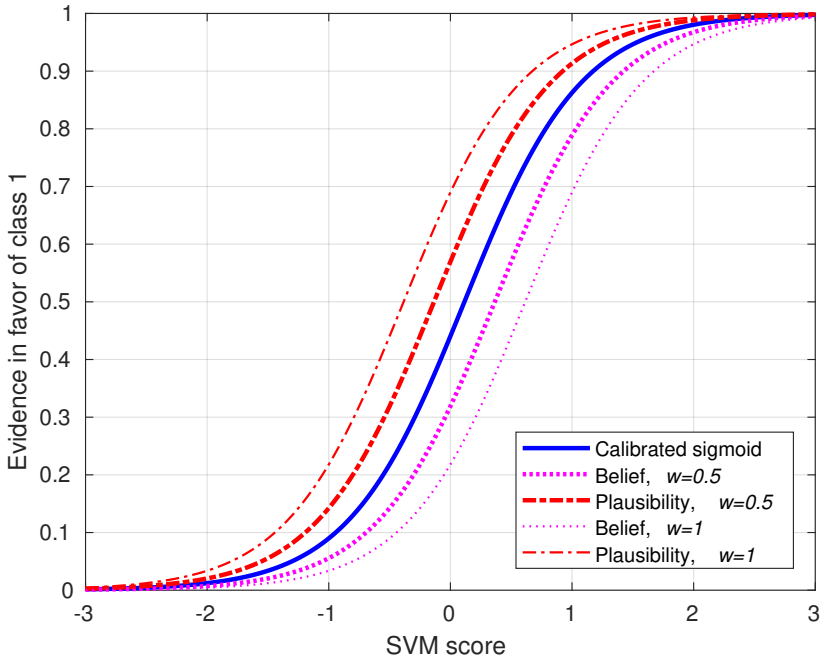


Figure 4.1: Example of a sigmoid function obtained with calibration, and derived Belief and Plausibility bounds at different structuring element w sizes. In our case, "class 1" corresponds to the H hypothesis.

are their SVM scores (i.e. their distances to the classification hyperplane), Eq. (3.14) provides the probability estimates $\mathrm{P}(y=1|\mathbf{x}_1)=0.28$ and $\mathrm{P}(y=1|\mathbf{x}_2)=0.98$. Note that for this example, where we consider only one classifier, subscripts refer to different samples $\mathbf{x}_1$ and $\mathbf{x}_2$ and the classifier index is omitted. Then, we can derive the associated Bayesian BBAs by simply assigning the probability estimate to the mass on H, and by computing the mass on $\overline{\mathrm{H}}$ accordingly. For example, considering sample $\mathbf{x}_1$, $m^{\mathscr{B}}_{\mathbf{x}_1}(\mathrm{H})=\mathrm{P}(y=1|\mathbf{x}_1)$ and $m^{\mathscr{B}}_{\mathbf{x}_1}(\overline{\mathrm{H}})=1-\mathrm{P}(y=1|\mathbf{x}_1)$. Then, by applying erosion with a flat structuring element of width w (in the example $w=1$) we can discount the mass on singleton hypotheses by an amount computed with Eqs. (4.18) and (4.19), as the difference between Bel and Pl. In this way we take into account the imprecision on the estimated sigmoid, and the smaller the distance of a sample to the SVM hyperplane, the higher the amount of imprecision that will be considered. In our example, sample $\mathbf{x}_1$ stands in the uncertain area between support vectors ($|s_{\mathbf{x}_1}<1|$), so that we know that a small change in the logistic optimal parameter estimation could possibly lead to a significant change in the probability estimate. On the contrary, sample $\mathbf{x}_2$ has

Table 4.1: Example of BBA allocation based on calibrated scores, assuming $\lambda_0^*=-2$, $\lambda_1^*=-0.05$ and erosion structuring element of width $w=1$. Only the focal elements are reported. In this example, where we consider only one classifier, subscripts refer to different samples $\mathbf{x}_1$ and $\mathbf{x}_2$ and the classifier index is omitted for clarity of notation.

Score	**Bayesian BBA**	**Imprecise score-based BBA**
$s_{\mathbf{x}_1}=-0.5$	$m^{\mathscr{B}}_{\mathbf{x}_1}(\mathrm{H})=0.28$ $m^{\mathscr{B}}_{\mathbf{x}_1}(\overline{\mathrm{H}})=0.72$	$\widetilde{m}_{\mathbf{x}_1}(\mathrm{H})=0.12$ $\widetilde{m}_{\mathbf{x}_1}(\overline{\mathrm{H}})=0.49$ $\widetilde{m}_{\mathbf{x}_1}(\Theta)=0.39$
$s_{\mathbf{x}_2}=+2$	$m^{\mathscr{B}}_{\mathbf{x}_2}(\mathrm{H})=0.98$ $m^{\mathscr{B}}_{\mathbf{x}_2}(\overline{\mathrm{H}})=0.02$	$\widetilde{m}_{\mathbf{x}_2}(\mathrm{H})=0.95$ $\widetilde{m}_{\mathbf{x}_2}(\overline{\mathrm{H}})=0.01$ $\widetilde{m}_{\mathbf{x}_2}(\Theta)=0.04$

an associated SVM score which is relatively high, and thus represents a test sample for which the classification is quite sure and will not easily change even in presence of calibration inaccuracy. With the proposed BBA allocation in the context of BF framework, we are therefore able to assign a higher value of imprecision to sample $\mathbf{x}_1$ with respect to $\mathbf{x}_2$.

4.3.2 BBA definition based on pixel neighborhood information

Regarding the second type of imprecision, namely the spatial one, it comes from the fact that in the context of high-density crowd pedestrian detection strong occlusions make the head of each pedestrian barely visible. Besides, due to the specific geometry of the recordings, each head corresponds to few pixels. The most effective head detectors are based on features computed in subwindows around the pixel of interest, which further increases the spatial imprecision of the detection. For this reason, we model the spatial imprecision due to the close resolutions of object (head) and descriptor respectively by performing opening operation in the spatial domain to discount the BBA taking into account the neighborhood heterogeneousness.

In particular, the BBA allocation proposed is able to take into account both types of imprecision, aiming to be more robust to possible imperfections of the learned sigmoid from which the mapping from SVM scores to probability values is made, while at the same time taking into account the information coming from neighboring pixels in the image space. Practically, we process two successive discounting steps on the initial Bayesian BBA derived from the learned sigmoid. Firstly, having learned the sigmoid of classifier n by logistic regression, we define BBAs to model the imprecision due to possible errors in the calibration, by applying an erosion operator in the 2D space where SVM calibration scores are projected with respect to their label. Then, we increase the mass on Θ discounting the previous BBA by performing a morphological opening operation, this time in the image space, to take into account neighbor pixels information based on the assumption that they are likely to belong to the same class.

More in detail, dilation δ_w and erosion $\mathscr{E}_w$ operators depending on the structuring element of width w are composed with the calibrated sigmoid σ_n relative to classifier n, in order to derive the two different sigmoid functions, denoted $(\delta_w \circ \sigma_n)$ and $(\mathscr{E}_w \circ \sigma_n)$, representing Pl_n and Bel_n functions (cf. Fig. 4.1) which can be evaluated on every score $s_{\mathbf{x},n}$ relative to sample $\mathbf{x}$ and classifier n. This takes into account the imprecision of the calibration step.

Since we perform a dense classification of an entire testing image, our test samples coincide with the pixels set belonging to the pixel domain, noted $\mathscr{P}$. Now, to stress this dualism yet emphasizing the fact that we are considering the image space, we privilege from now on the notation "pixel" $\mathbf{x}$ instead of "test sample" (used so far), although the two terms could be possibly used interchangeably.

Then, for each pixel $\mathbf{x}$ and classifier n independently, we derive the 'one-time' discounted BBA $\widetilde{m}_{\mathbf{x},n}$, so that at the end of this step we get a map (image) of BBAs $\left\{\widetilde{m}_{\mathbf{x},n}, \mathbf{x} \in \mathscr{P}\right\}$, where $\mathscr{P}$ is the pixel domain. This image $\widetilde{\mathscr{M}_n}$ is composed by four layers corresponding to the mass values of any hypothesis in $\left\{\emptyset, \mathrm{H}, \overline{\mathrm{H}}, \Theta\right\}$, respectively. Then, applying an opening to $\widetilde{\mathscr{M}_n}$ second and third layers (i.e., the ones corresponding to singleton hypotheses), and increasing the Θ layer values accordingly, the map $\mathscr{M}_n$ of the final BBAs $m_{\mathbf{x},n}$ is derived. This allows us to model the spatial imprecision of the detectors.

Specifically, with $s_{\mathbf{x},n}$ being the SVM score given by classifier n associated to pixel $\mathbf{x}$, we have:

$$\begin{cases} \forall \mathbf{x} \in \mathscr{P}, \widetilde{m}_{\mathbf{x},n}(\mathrm{H}) &= (\mathscr{E}_w \circ \sigma_n)(s_{\mathbf{x},n}), \\ \forall \mathbf{x} \in \mathscr{P}, \widetilde{m}_{\mathbf{x},n}(\overline{\mathrm{H}}) &= 1 - (\delta_w \circ \sigma_n)(s_{\mathbf{x},n}), \\ \forall \mathbf{x} \in \mathscr{P}, \widetilde{m}_{\mathbf{x},n}(\Theta) &= 1 - \widetilde{m}_{\mathbf{x},n}(\mathrm{H}) - \widetilde{m}_{\mathbf{x},n}(\overline{\mathrm{H}}). \end{cases} \tag{4.20}$$

where σ_n is the learned sigmoid for classifier n, while $(\mathscr{E}_w \circ \sigma_n)$ and $(\delta_w \circ \sigma_n)$ its eroded and dilated results respectively with a (flat) structuring element of width w, applied in the score space. Then, in the image space,

Table 4.2: Neighborhood spatial arrangement for samples $\mathbf{x}_1$ and $\mathbf{x}_2$. Corresponding mass allocations are reported in Table 4.3.

neighborhood of sample $\mathbf{x}_1$:			neighborhood of sample $\mathbf{x}_2$:		
	$\mathbf{x}_{11}$			$\mathbf{x}_{21}$	
$\mathbf{x}_{14}$	$\mathbf{x}_1$	$\mathbf{x}_{12}$	$\mathbf{x}_{24}$	$\mathbf{x}_2$	$\mathbf{x}_{22}$
	$\mathbf{x}_{13}$			$\mathbf{x}_{23}$	

Table 4.3: Example of proposed BBA allocation after discounting based on SVM scores, for neighborhood of samples $\mathbf{x}_1$ and $\mathbf{x}_2$ spatially arranged as reported in Table 4.2. BBA allocation for samples $\mathbf{x}_1$ and $\mathbf{x}_2$ is already reported in Table 4.1.

	$\widetilde{m}_{\mathbf{x}_{11}}$	$\widetilde{m}_{\mathbf{x}_{12}}$	$\widetilde{m}_{\mathbf{x}_{13}}$	$\widetilde{m}_{\mathbf{x}_{14}}$	$\widetilde{m}_{\mathbf{x}_{21}}$	$\widetilde{m}_{\mathbf{x}_{22}}$	$\widetilde{m}_{\mathbf{x}_{23}}$	$\widetilde{m}_{\mathbf{x}_{24}}$
H	0.8	0.2	0.7	0.01	0.95	0.94	0.98	0.95
$\overline{\text{H}}$	0.19	0.4	0.2	0.8	0.04	0.03	0.01	0.03
Θ	0.01	0.4	0.1	0.19	0.01	0.03	0.01	0.02

$$\begin{cases} \mathscr{M}_n(\emptyset) = \{0\}_{\mathbf{x}\in\mathscr{P}}, \\ \forall \text{A} \in \left\{\text{H}, \overline{\text{H}}\right\}, \mathscr{M}_n(\text{A}) = \gamma_a\left(\widetilde{\mathscr{M}_n}(\text{A})\right), \\ \mathscr{M}_n(\Theta) = \{1\}_{\mathbf{x}\in\mathscr{P}} - \mathscr{M}_n(\text{H}) - \mathscr{M}_n(\overline{\text{H}}), \end{cases} \tag{4.21}$$

where $\mathscr{M}_n(\text{A})$ is the layer image associated to hypothesis A, $\forall \text{A} \in 2^\Theta$, and γ_a is the opening operator of parameter a applied in the image domain. A spatial Gaussian structuring element fitted in a window of radius a is used, to better take into account the prior on the spatial consistency.

Note that the two morphological operations described are not commutative, since they are applied in two different spaces, i.e. score and image domains, and we find it more natural to firstly consider the imprecision due to the calibration step and later consider the imprecision in the spatial context.

Let us continue with the toy example proposed in the previous section. As in the previous example, the subscript related to the classifier is omitted as there is no ambiguity since only one classifier is considered. Table 4.2 shows the spatial arrangement of neighbor samples around the

Table 4.4: Example of BBA allocation for samples $\mathbf{x}_1$ and $\mathbf{x}_2$. From the BBAs based on imprecise score we derive the final BBAs applying a second discounting based on neighboring pixel heterogeneity (in this example, with flat 4-connectivity structuring element).

Sample	**Imprecise score-based BBA**	**Final BBA**
$\mathbf{x}_1$	$\widetilde{m}_{\mathbf{x}_1}(\text{H}) = 0.12$	$m_{\mathbf{x}_1}(\text{H}) = 0.01$
	$\widetilde{m}_{\mathbf{x}_1}(\overline{\text{H}}) = 0.49$	$m_{\mathbf{x}_1}(\overline{\text{H}}) = 0.19$
	$\widetilde{m}_{\mathbf{x}_1}(\Theta) = 0.39$	$m_{\mathbf{x}_1}(\Theta) = 0.8$
$\mathbf{x}_2$	$\widetilde{m}_{\mathbf{x}_2}(\text{H}) = 0.95$	$m_{\mathbf{x}_2}(\text{H}) = 0.94$
	$\widetilde{m}_{\mathbf{x}_2}(\overline{\text{H}}) = 0.01$	$m_{\mathbf{x}_2}(\overline{\text{H}}) = 0.01$
	$\widetilde{m}_{\mathbf{x}_2}(\Theta) = 0.04$	$m_{\mathbf{x}_2}(\Theta) = 0.05$

Table 4.5: Example of probability of H in **x** neighborhood, **x** being the central pixel, given by four different classifiers after score calibration.

classifiers 1 to 3:			classifier 4:		
	0.7			0.5	
0.5	**0.6**	0.5	0.5	**0.1**	0.5
	0.5			0.5	

considered $\mathbf{x}_1$ and $\mathbf{x}_2$. Let us suppose that neighbors have associated BBAs reported in Table 4.3 after BBA allocation based on SVM scores. BBA allocation for central samples $\mathbf{x}_1$ and $\mathbf{x}_2$ is already reported in Table 4.1. Note that the spatial arrangement of the samples is fully independent from their position in the score space. It is evident in the example that $\mathbf{x}_2$ has a more homogeneous neighborhood with respect to $\mathbf{x}_1$. This reflects in a higher discounting for sample $\mathbf{x}_1$ (for simplicity, in the example applying erosion with a flat 4-connectivity structuring element). Note that with the Bayesian allocation we would have assigned to $\mathbf{x}_1$ a high mass on $\overline{\mathrm{H}}$, while taking into account the two types of imprecision we end up with a BBA having a high value of ignorance, that will not contribute a lot in the conjunctive combination with the other classifiers. On the contrary, the final BBA allocation of $\mathbf{x}_2$ reflects its Bayesian counterpart, since its calibrated score is quite reliable and its neighborhood is homogeneous.

4.3.3 BBAs combination

Considering the N different descriptors, N BBAs are defined as explained for every test sample, i.e. pixel, $\mathbf{x} \in \mathscr{P}$. According to the BBA obtained from descriptor n, the uncertainty of a head presence in $\mathbf{x}$ ranges between $\mathrm{Bel}_{\mathbf{x},n}(\mathrm{H}) = m_{\mathbf{x},n}(\mathrm{H})$ and $\mathrm{Pl}_{\mathbf{x},n}(\mathrm{H}) = m_{\mathbf{x},n}(\mathrm{H}) + m_{\mathbf{x},n}(\Theta)$, so that $m_{\mathbf{x},n}(\Theta)$ represents the imprecision on the uncertainty value provided by n^{th} descriptor for the given sample. In the proposed model, the uncertainty comes from the binary classifier score, whereas the imprecision comes both from the initial score calibration and from spatial heterogeneity of uncertainty values within the considered structuring element.

Finally, the combination between BBAs can be performed. As the descriptors are considered *cognitively* independent, the Dempster's rule or its unnormalized version, the conjunctive combination rule, are well-suited for this task.

In our case where $|\Theta| = 2$, and considering $m_{\mathbf{x},n}$ BBAs allocation, the analytic result of the conjunctive combination rule may be derived:

$$\begin{cases} m_{\mathbf{x}}(\mathrm{A}) &= \sum\limits_{\substack{(\mathrm{B}_1,\ldots,\mathrm{B}_\mathrm{N}) \in \{\mathrm{A},\Theta\}^\mathrm{N}, \\ \exists n \in [1,\mathrm{N]} s.t. \mathrm{B}_n = \mathrm{A}}} \prod_{n=1}^{\mathrm{N}} m_{\mathbf{x},n}(\mathrm{B}_n), \forall \mathrm{A} \in \left\{\mathrm{H}, \overline{\mathrm{H}}\right\}, \\ m_{\mathbf{x}}(\Theta) &= \prod_{n=1}^{\mathrm{N}} m_{\mathbf{x},n}(\Theta), \\ m_{\mathbf{x}}(\emptyset) &= 1 - m_{\mathbf{x}}(\mathrm{H}) - m_{\mathbf{x}}\left(\overline{\mathrm{H}}\right) - m_{\mathbf{x}}(\Theta). \end{cases} \tag{4.22}$$

The result is thus a single four-layer map $\mathscr{M}$ of BBAs $m_{\mathbf{x}}$, where the overall ignorance is reduced as a result of the combination, but at the same time a conflict component may appear in each pixel.

4.3.4 Final decision from the obtained BBA

Finally, for every sample, the decision is taken from its corresponding $m_{\mathbf{x}}$ ((7.8)).

Binary decision can be taken considering singleton hypotheses, e.g. from maximum of Belief or Plausibility. To illustrate the interest of modeling imprecision in addition to uncertainty,

Table 4.6: Mass allocation (both Bayesian and proposed one), combination (both with conjunctive and Dempster's rules) and decision (in bold) considering the example probability maps reported in Table 4.5. For example simplicity, erosion with a flat 4-connectivity structuring element is used in the BBA allocation; for comparison, normalized product of probability values is shown.

	H	$\overline{H}$	$\{H, \overline{H}\}$	$\emptyset$
$m^{\mathscr{B}}_{\mathbf{x},1} = m^{\mathscr{B}}_{\mathbf{x},2} = m^{\mathscr{B}}_{\mathbf{x},3}$	0.6	0.4	/	/
$m^{\mathscr{B}}_{\mathbf{x},4}$	0.1	0.9	/	/
$m_{\mathbf{x},1} = m_{\mathbf{x},2} = m_{\mathbf{x},3}$	0.5	0.3	0.2	0
$m_{\mathbf{x},4}$	0.1	0.5	0.4	0
$m_{\mathbf{x},\mathbb{O}^4_1}$	**0.168**	0.109	0.003	0.72
$m_{\mathbf{x},\oplus^4_1}$	**0.6**	0.389	0.011	0
$\prod_{n=1}^{4} P_{\mathbf{x},n}(\widetilde{H})\ \forall \widetilde{H} \in \{H, \overline{H}\}$	0.28	**0.72**	/	/

let us consider the following example with a pixel belonging to a head and four different classifiers, i.e. *sources*, available to detect it. After the SVM training and Platt's probabilistic calibration procedures, as shown in Table 4.5 three of them provide a probability of H equal to 0.6 ($P_{n\in\{1,2,3\}}(\overline{H}) = 0.4$); however punctual noise present in the fourth source leads to $P_4(\overline{H}) = 0.9$ (and $P_4(H) = 0.1$) so that decision based on product of probability values (normalized, so that the sum of all the hypotheses is 1) would lead to the wrong label, $\overline{H}$ (note that the normalization does not have any impact on the final decision but it is simply performed in order to obtain a well-defined probability mass function). Now, using the proposed evidential approach, information coming from the calibration process as well as from the pixel's neighborhood in the image space are taken into account to perform a tailored discounting of unreliable classification responses. In the context of this example, for visualization purposes, we apply only the second discounting based on spatial information. Table 4.6 shows that the evidential combination leads to the right decision, H, both using the conjunctive combination rule and the Dempster's rule.

However rather than obtaining a binary decision we would like to maintain soft decisions, in order to derive more accurate statistics and to not lose available information for successive processing. To this extent, pignistic probability (Eq. 4.11) associated to each pixel $\mathbf{x}$ in our setting is computed as: $\forall A \in \Theta$,

$$\mathrm{BetP}_{\mathbf{x}}(A) = \frac{1}{1 - m_{\mathbf{x}}(\emptyset)} \cdot \left(m_{\mathbf{x}}(A) + \frac{m_{\mathbf{x}}(\Theta)}{2} \right). \tag{4.23}$$

This allows us to assign a probabilistic interpretation to the resulting BBA associated to each test sample, i.e. pixel.

In particular, we compute a single-layer BetP(H) image map where at every pixel the $\mathrm{BetP}_{\mathbf{x}}(H)$ value will be differently normalized on the basis of the conflict value included in $m_{\mathbf{x}}$, represented by the mass on the empty set.

4.4 Experimental results

In order to perform the experiments regarding the proposed BBA allocation and combination, we employ an ensemble of SVM classifiers obtained using the different descriptors highlighted in Chapter 3, namely HOG, LBP, GABOR and DAISY.

Figure 4.2 shows the sigmoid functions obtained through the calibration step, for each descriptor considered, and the Bel and Pl functions that define the interval obtained by erosion using a structuring element of size $w = 0.5$. Two important considerations can be made. Firstly, it be-

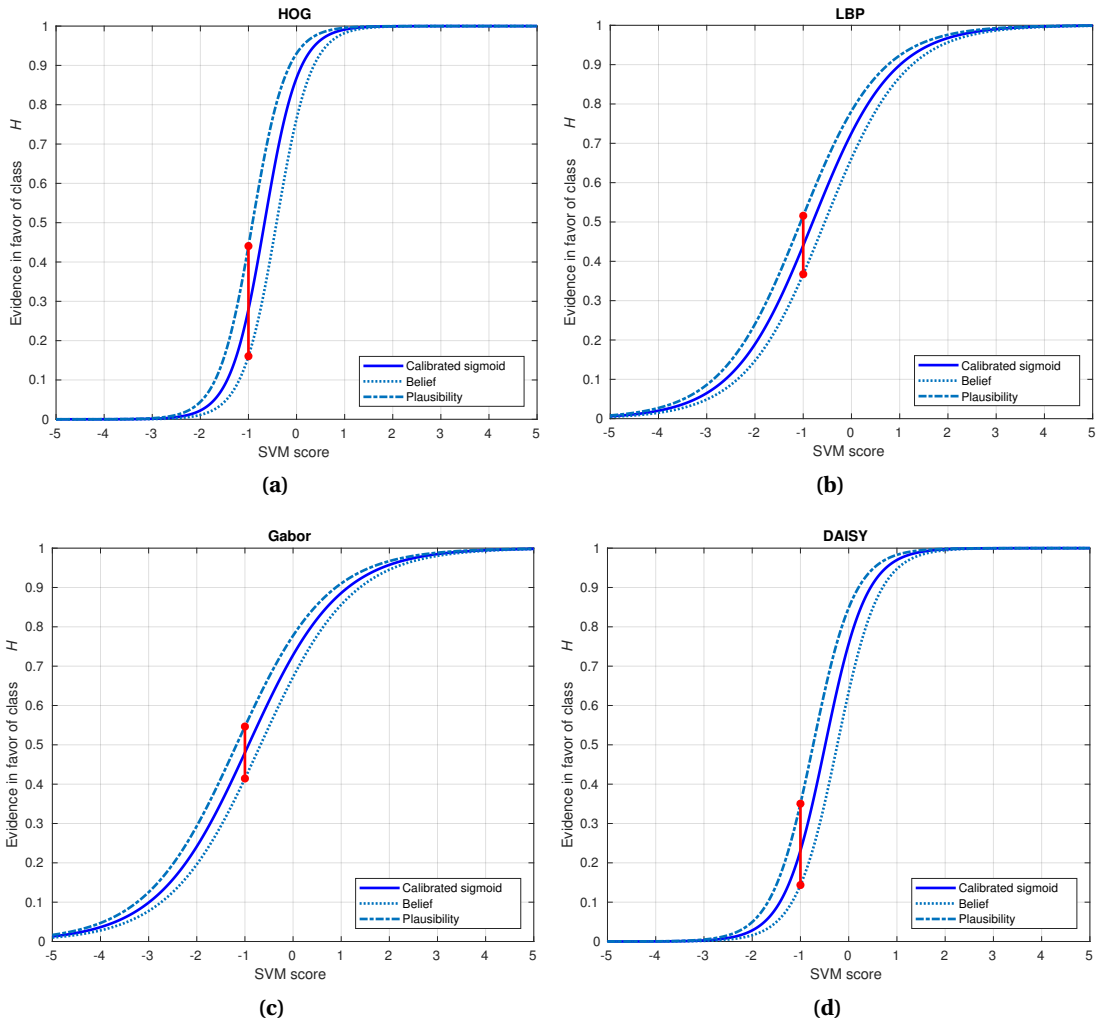


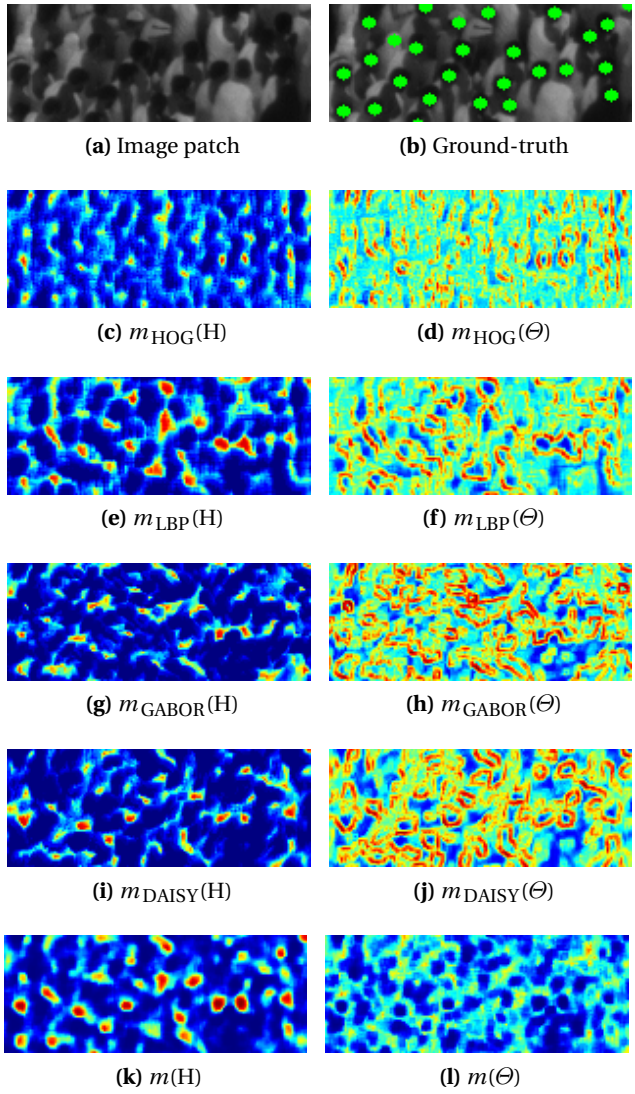
Figure 4.2: Sigmoid functions obtained with the calibration step and associated Bel and Pl sigmoids, with $w = 0.5$ size of structuring element. In red: example of the imprecision interval at SVM $score = -1$.

comes clear that with a larger structuring element we introduce more imprecision, allowing for less committed BBAs. On the contrary, a smaller structuring element introduces less imprecision. Secondly, considering the same size of the structuring element w, the interval between Bel and Pl functions, namely the imprecision we took into account, is bigger in presence of the steeper sigmoid functions like HOG and DAISY ones, for which even a small shift in the location would possibly cause the final decision to be different for the same score, resulting in less committed BBAs associated to each possible score. Conversely, scores from LBP and GABOR sigmoid functions have higher absolute values and less overlap, meaning that the decision about them is pretty stable, resulting in a smaller imprecision interval and thus in more committed BBAs.

Considering the effective choice of w value, we noticed that increasing the size of the structuring element w, the precision in the fusion results consistently increases, stressing the importance of the introduction of imprecision during calibration. However, increasing too much the size of the structuring element is detrimental for the detections, because they become too much uncertain and thus recall tends to decrease. For the experiments, we set $w = 2$, a good compromise that allows us to reach high precision without harming the overall recall.

After obtaining BBAs out of the calibration procedure, the second discounting in the image space is applied. For the experiments, we employ a spatial Gaussian structuring element of size $a = 2$, which allows us to better model the desired spatial consistency with respect to a crisp structuring element.

Figure 4.3 shows an example patch from an image of the dataset with its relative ground-truth,



(a) Image patch **(b)** Ground-truth

(c) $m_{\mathrm{HOG}}(\mathrm{H})$ **(d)** $m_{\mathrm{HOG}}(\Theta)$

(e) $m_{\mathrm{LBP}}(\mathrm{H})$ **(f)** $m_{\mathrm{LBP}}(\Theta)$

(g) $m_{\mathrm{GABOR}}(\mathrm{H})$ **(h)** $m_{\mathrm{GABOR}}(\Theta)$

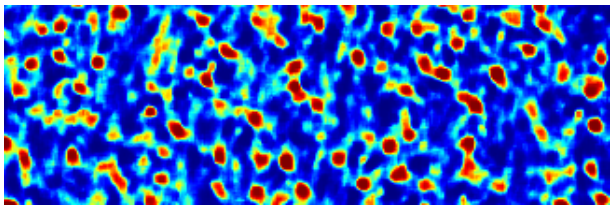
(i) $m_{\mathrm{DAISY}}(\mathrm{H})$ **(j)** $m_{\mathrm{DAISY}}(\Theta)$

(k) $m(\mathrm{H})$ **(l)** $m(\Theta)$

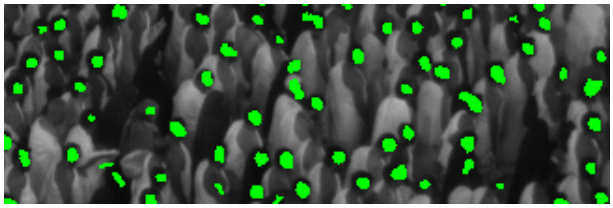
Figure 4.3: Example of an image patch with its associated ground-truth, BBAs allocations for each different detector and result after their combination.

where heads are highlighted in green. Then, for each different descriptor, BBAs allocation is shown through the mass on the head hypothesis (which corresponds to Bel(H)) and total ignorance arising after discounting both in the calibration and image space. The last row presents the results after the conjunctive combination rule. We notice again, as previously highlighted in Chapter 3, that each source has a specific behavior, a fact which underlines their complementarity. HOG and GABOR provide more localized detections, visible in the mass on H hypothesis, but more noise is present. On the contrary, LBP and DAISY provide larger and rougher results. Each descriptor then has higher values of ignorance in the pixels corresponding to the border of the heads, since their neighborhood is not homogeneous in the image space, and their correspondent score is probably in the area of uncertainty during calibration. With the conjunctive combination rule however, we are able to consistently reduce the total ignorance as shown in Figure 4.3l, and the shape of the head detections becomes very clear, as depicted in Figure 4.3k.

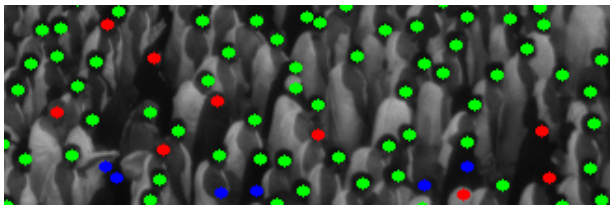
Figure 4.4 shows an example of the classification result on the basis of BetP(H) value at every pixel obtained with the conjunctive combination rule after the proposed BBA allocation based on the two consecutive discounting operations in the score and image spaces, hence the name Fusion SIS, i.e. “Fusion (after BBA allocation) in Score and Image Space”. Results are shown both in terms of output map and detections at a reasonable threshold ($th = 0.8$). This particular threshold choice has been made in order to be able to recover the most confident detections while at the same time



(a) Output BetP(H) map after Fusion SIS



(b) Detection map



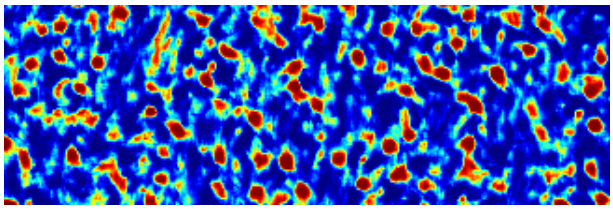
(c) NMS result map

Figure 4.4: Fusion SIS results in terms of BetP(H) colormap, detections at a given threshold and non-maximum suppression.

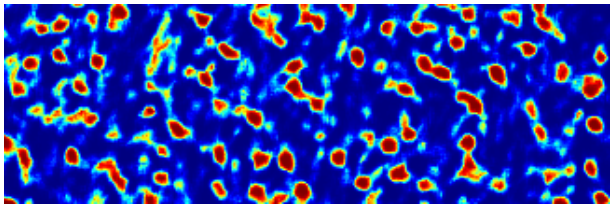
keeping them localized at the center of the head. Nevertheless, the learning process works at pixel level, with a balanced training set, while in the testing image the number of pixels corresponding to the two classes is unbalanced, hence the need for a quite high threshold (moreover, it is close to the PRBEP threshold value). Results after NMS are then presented in Figure 4.4c, setting the radius of a head $r = 3$, with $2r + 1$ minimum distance between two maxima (head centers) in order to avoid overlapping detections, highlighting in green TPs, in red FNs and in blue FPs. Most of the heads are correctly detected even in this condition of extreme density, while the number of false detections is kept low. False negative heads can be explained by the presence of dark heads or low contrast at the border.

Figure 4.5a and 4.5b provide a visual comparison of the fusion results obtained taking into account only spatial imprecision in the calibration or in the image domain *separately*, called Fusion SS, i.e. "Fusion (after BBA allocation in) Score Space" and Fusion IS, i.e. "Fusion (after BBA allocation in) Image Space" respectively. The detections obtained with the two approaches are similar in their locations, but are a bit larger taking into account imprecision during the calibration, while they are spatially more consistent considering imprecision in the spatial domain. Problematic areas are in both cases mostly at the boundary of the detections, that corresponds to pixels having their related score at lower distance from the hyperplane in the first case, and to pixels on which neighborhood disagrees the most in the second case. Considering the proposed BBA definition that takes into account both types of imprecision, namely Fusion SIS, whose result has been previously shown in Figure 4.4a, we are able to take the best out of the two approaches, obtaining larger while at the same time spatially homogeneous detections.

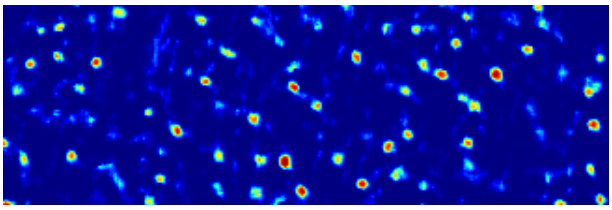
Figure 4.5c provides a visual comparison with a straightforward fusion solution which simply performs the product of the probabilities given by each independent detector at every pixel, without considering imprecision from calibration nor neighborhood information. There are just few heads for which the detection is pretty sure, and the size of the detections is always underestimated, since in order to have a confident detection all the sources must agree. Instead, taking into account possible sources of imprecision as proposed, we obtain more committed and smoother



(a) Output BetP(H) map after Fusion SS



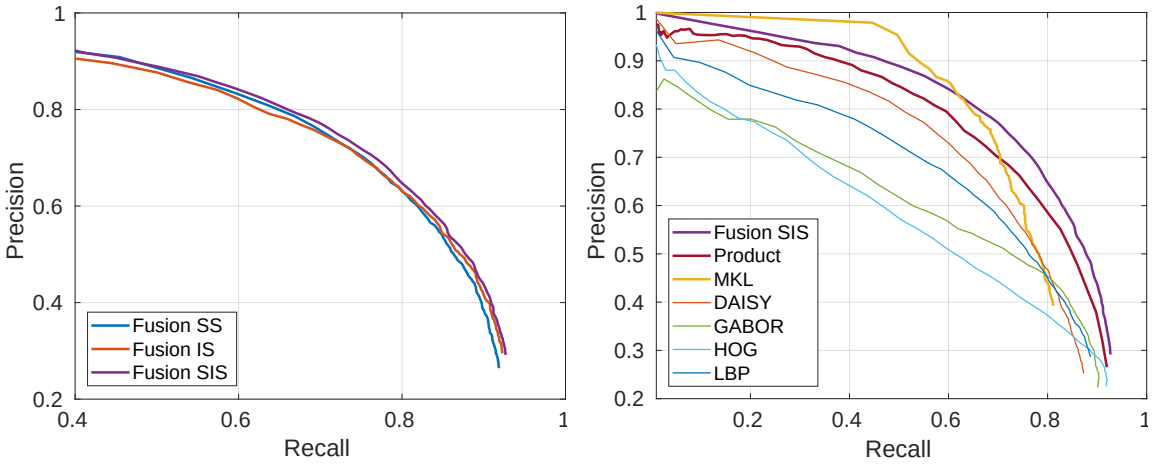
(b) Output BetP(H) map after Fusion IS



(c) Product output map

Figure 4.5: Comparison between fusion results after Fusion SS and Fusion IS in terms of BetP(H), and simple product of probabilities.

detections, that can be more useful as starting point for later stages such as tracking applications.



(a) **(b)**

Figure 4.6: (a) Fusion results in terms of PR-curves, after conjunctive combination rules with the three investigated BBA allocations; (b) PR-curves of the comparison of the proposed Fusion SIS, product of probabilities, MKL and the original four detectors.

A quantitative study using the three different proposed approaches separately is given by Figure 4.6a, where PR-curves are derived after NMS. Even if the three curves are similar, as the results are obtained after the NMS operation that flattens the already highlighted visible differences between the methods, the plot stresses the complementarity between the two approaches that take into account only a single source of imprecision (i.e. score or image space). The fusion after the proposed BBA allocation that performs two successive discounting operations gives indeed the best result. Indeed, we are able to tackle the problem of sparse false positives due to the unreli-

Table 4.7: PRBEP and AUPRC with the different fusion strategies.

	Fusion SS	Fusion IS	Fusion SIS	Product	MKL
PRBEP	0.73	0.73	**0.74**	0.69	0.70
AUPRC	0.77	0.77	**0.78**	0.74	0.72

ability of the descriptors while at the same time increasing the number and homogeneity of the detections.

Figure 4.6b shows a comparison of the proposed fusion approach with respect to the simple product of probabilities, MKL [90] and the original four sources, i.e. detectors. The results obtained with the proposed method provides overall better values both for precision and recall, highlighting once again the importance of considering imprecision both in the calibration and in the image space. Product of probabilities, while being better than the original sources, is not as good as the evidential fusion that is able to take into account the imprecision in addition to the uncertainty of each classifier. Regarding MKL, although it is able to reach higher values of precision thanks to the learned weights associated to each kernel, it does not exploit the available information in such a way to not lose detections, being not able to reach high values of recall.

Still in the context of a quantitative analysis, Table 4.7 provides quantitative values for PRBEP and AUPRC with the different fusion strategies. Again, the proposed evidential fusion clearly outperforms product of probabilities and MKL, and show the importance of taking into account possible sources of imprecision both in the score and image space.

These results conclude a first part of the work which aims to exploit the complementary nature of different descriptors providing orthogonal views of complex data. In the following Chapter, we will focus our attention on building competitive ensemble of classifiers using a limited amount of labeled training instances.

Chapter 5

Evidential QBC Active Learning

Contents

5.1 Motivation

Until this point, results have been shown on the basis of different SVM models trained with the same training set composed of manually carefully selected samples. For instance, positive samples have been selected in order to span across the greatest number of diverse heads appearance, e.g. for our datasets roughly the same number of dark-veiled women, white-veiled women, men with and without a hat, while at the same time presenting a "good" shape, i.e. only slightly occluded and with a rather well-contrasted background, in order to not employ very specific points for the learning that could lead to overfit being not very representative of the actual class distribution. In the same way, negative samples have been carefully selected as "difficult" points, e.g. shoulder's centers whose shape remind that of a head, casual circular structures, hands, clothes, floor (even though seldom visible), and more generally any point which could lead to a high response in terms of gradient or texture.

To illustrate the necessity of a well-defined training set, Fig. 5.1 compares the results obtained in the previous Chapter using the manually carefully selected samples, with the results obtained using a random training set (enforcing a minimum distance between selected points in the image for fairness). As we see, the classifiers are really sensitive to the choice of the samples, providing very different levels of performance after the fusion among them with the two different training sets. Thus, it is important to define a training set which spans over all the possible shades of sample characteristics while at the same time remaining focused on the specific targets.

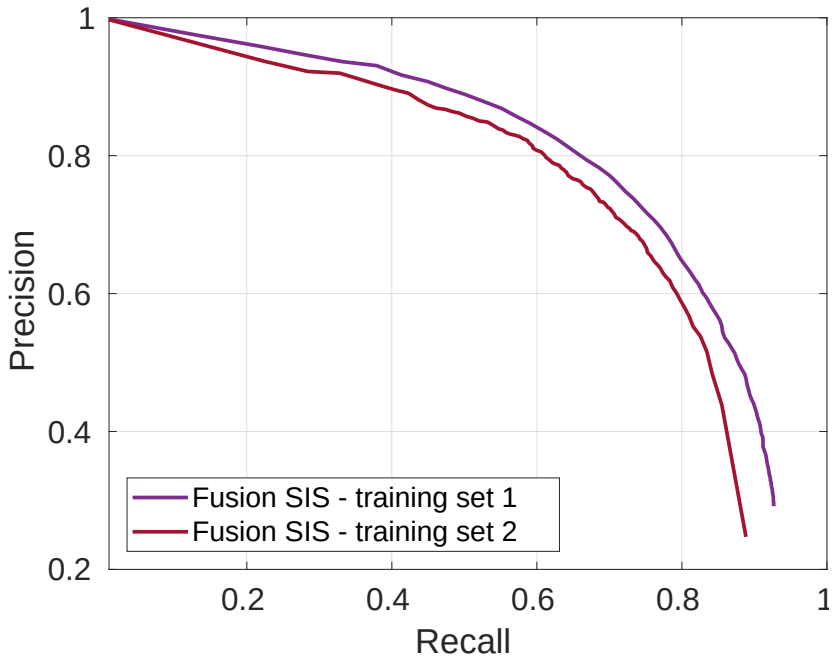


Figure 5.1: PR-curves of Fusion SIS in presence of two different training sets used to train the SVM classifiers (based on different descriptors) that compose the ensemble.

However, a somehow clear distinction among head types is generally impossible to be performed. Moreover, the notion of "good" positive samples and the notion of how "difficult" a point could be in order to be classified are quite subjective. Nevertheless, there is still a gap between which samples a human could *think* the classifier needs, and which ones are actually the most *informative* according to it.

To this extent, Active Learning (AL) has been proposed [42]. It relies on the assumption that if a learning algorithm is allowed to choose data from which to learn, it will reach better levels of performance with less training data [236]. AL algorithms work by posing "queries" to an *oracle* (e.g. a human annotator) about instances to be labeled. This approach is well-motivated by many machine learning applications, where unlabeled data may be abundant, but manually labeling is expensive, difficult and very time-consuming, from text classification [287] to robotics [35] and medical image classification [109] among others.

For this reason, we find that AL suits perfectly our needs. Instead of carefully labeling positive and negative instances, hard and laborious task which does not even guarantee that the chosen samples will be the best ones according to the classifiers, we exploit the ground-truth maps as oracle to obtain the right label, letting the algorithm choose which points to add to the training set.

Note that having ground-truth maps at our disposal, we could simply use all the pixels of an image as different training samples. However, this approach is not feasible for multiple reasons. Firstly, the training data would be extremely unbalanced between positive and negative examples. Secondly, for computational reasons. SVM are indeed rather computationally expensive. The complexity depends on the number of support vectors and algorithm used to solve the quadratic optimization, but regardless of the exact algorithm used the asymptotic computational cost of solving the SVM QP problem grows at least like $O(m^2)$ where m is the number of training samples when C is small, and like $O(m^3)$ when C gets larger [20]. Moreover, space complexity (which is $O(m^2)$) would become an issue as well, since the kernel matrix would become too large to be stored entirely in memory.

Lastly, it is hard to obtain precise ground-truth maps which are able to discriminate between head and background at pixel level: due to the high visual homogeneity of the images, it may be difficult to assign to a specific class pixels which are on the head's contours. Even worse, if

inserted in the training set the contour pixels could possibly confuse the classifier since their exact label is not immediately clear even for a human annotator. Redundant information coming from neighboring pixels is then useless since only the support vectors contribute in the definition of the separation hyperplane.

For all these reasons, it is crucial to be able to automatically select a relatively small training set composed by *informative* samples, where the measure of informativeness is directly given by the classifier (or ensemble of classifiers like in our case), with respect to some metrics.

5.2 Active Learning overview

There are two main scenarios in which the AL algorithm may ask queries about the samples to label:

- *Stream-based (selective) sampling* – The learner obtains one unlabeled instance at a time, in a sequential way from some streaming data source, and it must decide on the fly whether to query for the label and add it to the training set, or simply discard it. This decision can be taken in several ways. For example, one approach is to define a measure of utility or information content, such that instances with higher utility are more likely to be queried [50]. Another approach is to explicitly compute parts of the instance space that are still ambiguous, i.e. *regions of uncertainty*, and query only the incoming instances that fall within it [43];

- *Pool-based sampling* – Pool-based AL [157] relies on an initial small set of labeled instances, $\mathscr{L}$, and a larger set of unlabeled ones, $\mathscr{U}$. Batches of *informative* training samples are iteratively selected from $\mathscr{U}$ and added to $\mathscr{L}$, with respect to some heuristics, after a query about their actual label. Contrarily to stream-based approaches, it evaluates and ranks the entire collection of unlabeled data (or a selected part of it) before choosing the best sample to add to the training set on the basis of the current model.

Pool-based sampling appears to be the most popular method employed for applied research in AL, while stream-based selective sampling is more common in the theoretical literature. Indeed, situations where a large pool of unlabeled samples is available are common and the cost of labeling remains an issue for many applications.

To select the right samples to query, many strategies have been proposed. The most popular ones are *uncertainty sampling* and *Query by Committee* (QBC), with many variants in order to balance exploitation of the current classifiers and exploration of the feature spaces [31]. They can be applied both in the context of stream-based and pool-based learning, but since in this work we rely on a pool of unlabeled samples in the following we will describe these two strategies mainly for this context.

5.2.1 Uncertainty sampling

Uncertainty sampling [156] consists in iteratively requesting labels for training instances whose class remains uncertain, despite the information provided by the previously labeled instances. In this way the learning algorithm can focus its attention on the examples it finds confusing, selectively adjusting the boundary between classes. Popular strategies consist in querying the instance whose predicted output is the least confident or with maximum entropy. In the context of SVM classification the prevailing method is to select the samples which are closer to the separation hyperplane margin [234, 259].

More recently, DUAL [67] and QUIRE [116] methods have been proposed. The former is based on a density weighted uncertainty sampling based on estimated future residual error reduction after each actively sampled point, while the latter aims at selecting both informative and representative examples on the basis of a prediction of the uncertainty for the yet unlabeled samples.

The authors of [25] consider instead the *diversity* between samples, proposing a selection strategy which aims at reaching a trade-off between the minimum distance from the hyperplane margin and the maximum angle between the hyperplanes defined by each sample feature. In the context of image classification, diversity among the selected samples can be reached using spatial information as well, such as in [203], where the authors propose three criteria to favor samples distant from the ones already present in the training set, namely an Euclidean distance, a distance based on the Parzen window method applied in the spatial domain, and a distance that maximizes the spatial entropy variation value to distribute spatially the training samples as widely as possible.

Although uncertainty sampling offers an intuitive and flexible solution for augmenting the training set, this framework is suited in its standard form for relying on a single classifier and is therefore not really useful to be investigated further in the context of this work which is in fact based on ensembles. Besides, precisely because it is based on a single classifier, it has been criticized of being quite shortsighted, as the utility scores are based on the output of just a single hypothesis that, trained by definition on little data tends to bias the active learning sampling strategy. The use of multiple classifiers is useful to mitigate or circumvent this issue, and will be explored further in the context of our application.

5.2.2 Query-By-Committee

On the other hand, QBC [237] exploits a committee of classifiers and operates by asking for the label of the sample on which the ensemble disagrees the most. This approach is better suited for more complex classification tasks which benefit from multiple classifiers providing different interpretations of the input data, such as the application we consider here.

When deploying a QBC algorithm three questions may arise:

1. *How to build the committee set?* – Usually generic ensemble learning algorithms explained in Sec. 2.3.2.2 are used for the construction of the committee. Query-by-bagging [23] or query-by-boosting [80] can be used to train weak classifiers on (weighted) randomly sampled variations of the training data set. Alternatively, a single model can be exploited and many variations of it can be derived, e.g. changing its intrinsic parameters, like in [180] for naive Bayes, using the Dirichlet distribution over model parameters;

2. *How to quantify the disagreement among committee members in order to define a strategy to select the new samples?* – There exists a variety of heuristics to measure the disagreement among a classifier ensemble, but surely the most popular ones are (Soft) Vote Entropy (SVE) [236], and Kullback-Leibler (KL) divergence [146]. Other measures include Jensen-Shannon divergence [182], a smoothed version of KL divergence, and F-compliment [192], based on the F1-measure. A combination between Vote Entropy and KL divergence is proposed in [303] in the specific context of stream-based QBC, where a continuous stream of samples is given as input and the active learner must decide if it is worth or not asking for the true label. Recently, [133] proposed an interesting method to merge diversity and density measures in the instance selection, to ensure variety within the batch and in the whole training set;

3. *How to obtain a final robust classification?* – Finally, the classification is usually performed at every iteration on the basis of the committee member responses. Common combination methods have been discussed in Sec. 2.3.2.1. Typically in the context of QBC a (weighted) average among the various results is performed, or the model that provides the best performance (according to a given metric, e.g. accuracy) is simply retained.

However, a clear limitation of traditional QBC approaches is that the selection of the new samples to be added to the training set is performed independently from the (optional) committee member combination, that is only used to derive statistics for evaluation purposes. The possible information arising from the combination of the committee members is not exploited.

From our part, the definition of the fusion strategy based on BF framework presented in Chapter 4 allows us to naturally have at our disposal several clues to quantify the disagreement between committee members. The result of the source combination indeed is a BBA associated to every unlabeled sample, that intrinsically contains conflict and ignorance components. For this reason once again we find it appropriate to work in the evidential domain: from the one hand, through the definition of appropriate BBAs we can model the imprecision over the uncertainty value provided by each classifier; on the other hand, the BF framework directly provides indicators to quantify the disagreement between committee members.

5.3 Evidential QBC Active Learning

We thus propose a QBC algorithm that takes a committee of models which are all trained on the same training set, but representing competing hypotheses supported by different SVM classifiers based on gradient, texture and orientation descriptors described in Section 3.2, so that we find it natural to build a set of classifiers with them. Firstly, we use BF framework to perform fusion between the different pedestrian detectors, as explained in Sec. 4.3, and then we propose and investigate different evidential-based measures for the selection of the batch of new training samples, in a pool-based sampling setting.

The evidential framework is therefore not only involved in the combination of the sources to obtain a robust decision, but it plays at the same time an original role in the definition of new sample selection strategies at each iteration of the AL process.

After having built $\mathscr{C}$, the committee of classifiers of cardinality $|\mathscr{C}| = \mathrm{N}$ sources, QBC relies on some heuristics to measure the disagreement among them, in order to find the most informative samples to add to the training set $\mathscr{L}$. We moreover ensure *diversity* among samples in two different and complementary ways: firstly in the feature space, following the work of [25], by a maximum angle between the hyperplanes defined by each sample feature; secondly, in the image space, by a minimum Euclidean distance applied in the spatial domain between instances already in the training set and in the current batch.

In the following, we investigate traditional disagreement metrics such as Soft Vote Entropy and KL divergence, as well as new evidential-based disagreement measures.

5.3.1 Traditional disagreement measures in QBC and their limitations

Specifically, given the set of mutually exclusive hypotheses $\Theta = \left\{\mathrm{H}, \overline{\mathrm{H}}\right\}$ and the committee $\mathscr{C}$ of classifiers of cardinality N, Soft Vote Entropy asks for the label of the unlabeled sample such that:

$$\mathbf{x}^*_{\mathrm{SVE}} = \underset{\mathbf{x}\in\mathscr{U}}{\operatorname{argmax}} \sum_{y\in\Theta} \mathrm{P}_{\mathscr{C}}\left(y \mid \mathbf{x}\right) \log\left(\frac{1}{\mathrm{P}_{\mathscr{C}}\left(y \mid \mathbf{x}\right)}\right), \tag{5.1}$$

where $\mathscr{U}$ is the set of unlabeled samples ($\mathscr{U} \subset \mathscr{P}$), and

$$\mathrm{P}_{\mathscr{C}}\left(y \mid \mathbf{x}\right) = \frac{1}{\mathrm{N}} \sum_{n=1}^{\mathrm{N}} \mathrm{P}_n\left(y \mid \mathbf{x}\right), \tag{5.2}$$

is the average or *consensus* probability that y is the correct label according to the committee. Soft Vote Entropy is thus essentially an ensemble generalization of entropy-based uncertainty sampling. The log function, here and from now on, represents the logarithm to the base 2.

On the other hand, the KL divergence strategy adds a sample to the training set such that:

$$\mathbf{x}^*_{\mathrm{KL}} = \underset{\mathbf{x}\in\mathscr{U}}{\operatorname{argmax}} \frac{1}{\mathrm{N}} \sum_{n=1}^{\mathrm{N}} \mathscr{D}_{\mathrm{KL}}\left(\mathrm{P}_{\mathbf{x},n} \| \mathrm{P}_{\mathbf{x},\mathscr{C}}\right), \tag{5.3}$$

where $\mathrm{P}_{\mathbf{x},n} = \mathrm{P}_n\left(y \mid \mathbf{x}\right)$ and $\mathrm{P}_{\mathbf{x},\mathscr{C}} = \mathrm{P}_{\mathscr{C}}\left(y \mid \mathbf{x}\right)$ for simplicity of notation, while $\mathscr{D}_{\mathrm{KL}}$ is the KL divergence that quantifies the disagreement as the average divergence between the prediction of each classifier n in the committee and the consensus $\mathrm{P}_{\mathscr{C}}$, and is defined as:

$$\mathscr{D}_{\mathrm{KL}}\left(\mathrm{P}_{\mathbf{x},n} \| \mathrm{P}_{\mathbf{x},\mathscr{C}}\right) = \sum_{y \in \Theta} \mathrm{P}_n\left(y \mid \mathbf{x}\right) \log\left(\frac{\mathrm{P}_n\left(y \mid \mathbf{x}\right)}{\mathrm{P}_{\mathscr{C}}\left(y \mid \mathbf{x}\right)}\right). \tag{5.4}$$

The conceptual difference behind SVE and KL resides in the way they quantify the *disagreement*. Considering a committee of classifiers, the consensus probability $\mathrm{P}_{\mathscr{C}}\left(y \mid \mathbf{x}\right)$ between them could be uniform in two different cases. Firstly, all the classifiers have an uniform distribution among the hypotheses, so that the consensus distribution is also uniform. Secondly, the classifiers strongly disagree between them, but since the consensus is an average between their responses, it ends up being uniform among all the hypotheses as well. In the first case, all the classifiers agree that the label is uncertain, while in the second case they strongly support a different label. Since SVE only considers consensus, it cannot distinguish between the two cases. On the other hand, KL divergence would favor samples with uncertain consensus because of conflicting predictions given by the classifiers.

Besides these highlighted limitations, the mentioned measures do not exploit the possible information arising from the combination among the committee members, and the final result on which evaluation is performed is not taken into account in the selection of the new samples based on their maximization.

5.3.2 Proposed evidential disagreement measures

On our side, after having performed the combination between the various sources in the BF framework, the result is the map $\mathscr{M}$ where at each sample, i.e. pixel $\mathbf{x}$ of the image corresponds a BBA $m_{\mathbf{x}}$ that incorporates a different evidence of belonging to a certain class (head or not head), as well as a component of ignorance that remains after the combination, and conflict between the sources, i.e. the masses on Θ and $\emptyset$ respectively that come from the conjunctive combination. We can therefore extend the concept of Soft Vote Entropy to the evidential framework, to define new evidential measures of disagreement among committee members. The Maximum Entropy (ME) strategy will add to the training set sample such that:

$$\mathbf{x}^*_{\mathrm{ME}} = \underset{\mathbf{x} \in \mathscr{U}}{\operatorname{argmax}}\, \mathrm{H}(m_{\mathbf{x}}), \tag{5.5}$$

where in our case $m_{\mathbf{x}}$ is the BBA associated to the unlabeled sample $\mathbf{x}$, obtained after the explained BBA allocations and conjunctive combination (cf. Sec. 4.3), and $\mathrm{H}(\cdot)$ is a definition of the entropy function in the evidential domain.

Several definitions of *evidential entropy* have been proposed over the past decades, with the aim of measuring the degree of total uncertainty of a BBA, but a formulation satisfying all the desired properties still remains an open issue.

Table 5.1 summarizes some popular definitions, that we intend to investigate as heuristics to measure the disagreement among the committee members and therefore select the new training points. Some of them, like Höhle [108], Yager [288] and Nguyen [193] definitions are only able to measure the conflicting portion of uncertainty. Pal definition [200, 201] is an extension of Nguyen's one, taking into account also the cardinality of each focal element. The definition given by Dubois and Prade [68], on the contrary, captures only the non-specificity portion of uncertainty, quantifying how a BBA is imprecise. The most non-specific BBA is given by the vacuous BBA having $m(\Theta) = 1$, while the most specific BBAs are the Bayesian ones, so that non-specificity is a measure of how a BBA is committed among the various hypotheses. The formulation given by Lamata and Moral [153] and the more recent Deng [57] and Jiroušek and Shenoy [126] ones, combine both conflicting and non-specificity components in different ways. Regarding the conflicting part, Lamata et al. use Yager's definition which relies on the plausibility function, Deng uses Nguyen's formulation while Jiroušek et al. interpret it in a completely different way, as the Shannon's entropy of the plausibility probability function Pl_P [41], an alternative representation to pignistic probability for translating BBAs into probabilistic framework. Regarding the non-specificity component, Lamata

Table 5.1: Evidential entropy definitions given BBA m with discernment frame Θ

Reference	Entropy formulation
Höhle [108]	$\mathrm{H_O}(m) = \sum_{\mathrm{A}\in 2^\Theta} m(\mathrm{A}) \log\left(\frac{1}{\mathrm{Bel(A)}}\right)$
Yager [288]	$\mathrm{H_Y}(m) = \sum_{\mathrm{A}\in 2^\Theta} m(\mathrm{A}) \log\left(\frac{1}{\mathrm{Pl(A)}}\right)$
Nguyen [193]	$\mathrm{H_N}(m) = \sum_{\mathrm{A}\in 2^\Theta} m(\mathrm{A}) \log\left(\frac{1}{m(\mathrm{A})}\right)$
Pal et al. [200, 201]	$\mathrm{H_P}(m) = \sum_{\mathrm{A}\in 2^\Theta} m(\mathrm{A}) \log\left(\frac{\lvert\mathrm{A}\rvert}{m(\mathrm{A})}\right)$
Dubois and Prade [68]	$\mathrm{H_{DP}}(m) = \sum_{\mathrm{A}\in 2^\Theta} m(\mathrm{A}) \log\left(\lvert\mathrm{A}\rvert\right)$
Lamata and Moral [153]	$\mathrm{H_{LM}}(m) = \mathrm{H_Y}(m) + \mathrm{H_{DP}}(m)$
Deng [57]	$\mathrm{H_D}(m) = \mathrm{H_N}(m) + \sum_{\mathrm{A}\in 2^\Theta} m(\mathrm{A}) \log\left(2^{\lvert\mathrm{A}\rvert} - 1\right)$
Jiroušek and Shenoy[126]	$\mathrm{H_{JS}}(m) = \sum_{\mathrm{A}\in\Theta} \mathrm{Pl\_P(A)} \log\left(\frac{1}{\mathrm{Pl\_P(A)}}\right) + \mathrm{H_{DP}}(m)$
Jousselme et al. [128]	$\mathrm{H_J}(m) = \sum_{\mathrm{A}\in\Theta} \mathrm{BetP(A)} \log\left(\frac{1}{\mathrm{BetP(A)}}\right)$

et al. and Jiroušek et al. rely on Dubois and Prade definition, while Deng provides a brand new formulation. Alternatively, Jousselme et al. [128] firstly perform a pignistic transformation from BBA to probability mass function through BetP, and then apply Shannon's entropy on it. A similar definition, called pignistic entropy, appears in [63], in the context of the Dezert-Smarandache Theory (DSmT) [61, 62], that is a variant of the classical Dempster-Shafer Theory (DST). Since we indeed rely on DST, we refer in the following to Jousselme's definition. The advantage of such a formulation for our application is that since it is based on the BetP function, there is a direct link between it and the final map we use for decision and, possibly, crowd density evaluation application, which will be further explored as a possible application in the context of crowd macro-analysis.

Besides entropy-based criteria, the masses on Θ and $\emptyset$ can be directly exploited as indicators for the selection of the new samples. It is possible to directly derive two simple strategies, based on Maximum Ignorance (MI) and Maximum Conflict (MC) respectively:

$$\mathbf{x}^*_{\mathrm{MI}} = \underset{\mathbf{x}\in\mathscr{U}}{\operatorname{argmax}}\, m_{\mathbf{x}}(\Theta), \tag{5.6}$$

$$\mathbf{x}^*_{\mathrm{MC}} = \underset{\mathbf{x}\in\mathscr{U}}{\operatorname{argmax}}\, m_{\mathbf{x}}(\emptyset). \tag{5.7}$$

where in our case $m_{\mathbf{x}}$ is the BBA associated to the unlabeled sample $\mathbf{x}$, obtained after the explained BBA allocations and conjunctive combination (cf. Sec. 4.3).

Equation (5.6) favors the selection of new points for which all the classifiers do not have enough information to decide about their actual class, i.e. samples with maximal mass on the compound hypothesis. On the contrary, Eq. (5.7) supports the selection of points on which the classifiers disagree the most about their actual label, i.e. samples with maximal mass on the empty set. In Eq. (5.7) we choose to use a measure of total conflict derived from the conjunctive combination rule as disagreement measure. In [53, 54] total conflict is separated into internal and external components. Internal conflict quantifies the (self-)inconsistency of the n^{th} source, while external conflict is only based on the interaction between sources and does not integrate any self-inconsistency. The authors of [60] in particular agree with this subdivision, and they propose conflict measurements based on contour functions, making no a-priori assumptions regarding the possible dependence between sources.

The concepts of *conflict* and *ignorance* have already been used in the context of single classifier uncertainty sampling-based AL in [173], but with totally different meanings from those in the BF framework. In their work, conflict models the extent to which a new query point lies in the conflict region between two or more classes (whereas for us it refers to conflicting beliefs from different

Table 5.2: Example of the computation of evidential-based disagreement measures based on different mass allocations $m_{\mathbf{x}_1}, \ldots, m_{\mathbf{x}_4}$. The sample related to the bold value in each column is the one that is chosen to be added to the training set according to MC, MI and ME criteria with the related entropy measures.

	$\emptyset$	H	$\overline{\text{H}}$	Θ	BetP(H)	H_O	H_Y	H_N	H_P	H_DP	H_LM	H_D	H_RP	H_J
$m_{\mathbf{x}_1}$	0.01	0.1	0.1	**0.79**	0.5	0.67	0.03	0.93	1.72	**0.8**	0.83	2.19	**1.8**	**1**
$m_{\mathbf{x}_2}$	**0.79**	0.1	0.1	0.01	0.5	**1.02**	**0.88**	1.23	1.27	0.05	**0.94**	1.3	1.05	**1**
$m_{\mathbf{x}_3}$	0.4	0.1	0.1	0.4	0.5	0.86	0.09	**1.25**	**1.92**	0.66	0.75	**2.31**	1.66	**1**
$m_{\mathbf{x}_4}$	0.1	0.79	0.01	0.1	0.93	0.24	0.05	0.59	0.7	0.11	0.16	0.76	0.61	0.35

classifiers), while ignorance represents the distance of a new query point from the training samples seen so far, so that it is higher in areas of the version space not represented yet (while for us it is higher when for all the classifiers the point resides in their uncertainty area - in a sense, the two definitions are completely the opposite). Always in the different context of uncertainty sampling, in [242] there is a distinction between insufficient-evidence and conflicting-evidence uncertainties, but the concept of *evidence* does not refer to BF framework, but it is rather measured as a weighted similarity of a given sample to the support vectors.

We expect that the inclusion of samples with high ignorance or conflict (defined in the BF framework) will be beneficial for the learning process, respectively in order to sharpen the decision boundaries between the classes for all the classifiers and to reduce the overall conflict between the various sources. However, the former strategy exploits examples which are near the current decision margins in all the feature spaces, and it is not able to solve possible conflicts but it just adjusts the boundaries, while the latter allows for an exploration of the version spaces to select points which are not yet represented by the current models, but it could be prone to outlier selection. In this sense they are complementary strategies, and they should be used in conjunction with a criterion able to balance them. Alternatively, we expect entropy-based disagreement to be able to naturally find a trade-off between them as a measure of information gain.

Table 5.2 shows an example of four BBAs associated to different samples, and the decision about which sample to query based on the different evidential entropy criteria (i.e. the sample related to the bold value in each column). In particular, $m_{\mathbf{x}_1}$ has a high component of ignorance, $m_{\mathbf{x}_2}$ is a very conflicting BBA, $m_{\mathbf{x}_3}$ is not committed about any singleton hypothesis and at the same time has a high amount of both ignorance and conflict, while $m_{\mathbf{x}_4}$ is very committed about H hypothesis. The value of BetP(H) is also shown, to highlight the fact that the probabilistic framework assigns the same value to the first three BBAs even if they are intrinsically very different one from the others. As we expect, no measure selects $m_{\mathbf{x}_4}$ to be added to the training set, since it is quite committed while not conflicting and it would not provide much information. On the contrary, the first three BBAs are selected based on the different measures. A clear limitation of MI and MC criteria is that they fail detecting BBAs with relatively high values of both conflict and ignorance: MI selects $m_{\mathbf{x}_1}$ while MC selects $m_{\mathbf{x}_2}$, but they do not consider $m_{\mathbf{x}_3}$ at all, even if it represents a potentially interesting sample to add to the training set. Conversely, entropy-based criteria are able to better consider the relative allocation of masses through the various hypotheses. Using Höhle and Yager definitions of entropy, $m_{\mathbf{x}_2}$ is selected, highlighting their tendency to detect conflicting instances. Nguyen and Pal favor the selection of $m_{\mathbf{x}_3}$, prioritizing samples which are both not very committed and conflicting, even if Nguyen is more sensitive to conflict while Pal gives more importance to the ignorance component. Dubois and Prade's formulation of entropy favors samples with high ignorance, not being able to capture the conflict component. Among the three composite formulations that aim at taking into account both conflict and non-specificity (i.e., Lamata and Moral, Deng, Jiroušek and Shenoy), we can notice that they all prioritize different samples, but there is only a slightly difference among the entropy values associated to the first three BBAs. This suggests the fact that they would probably select the three of them to be part of the same batch. In the same way, Jousselme's definition based on BetP(H) encourages a diversity in terms of BBAs in

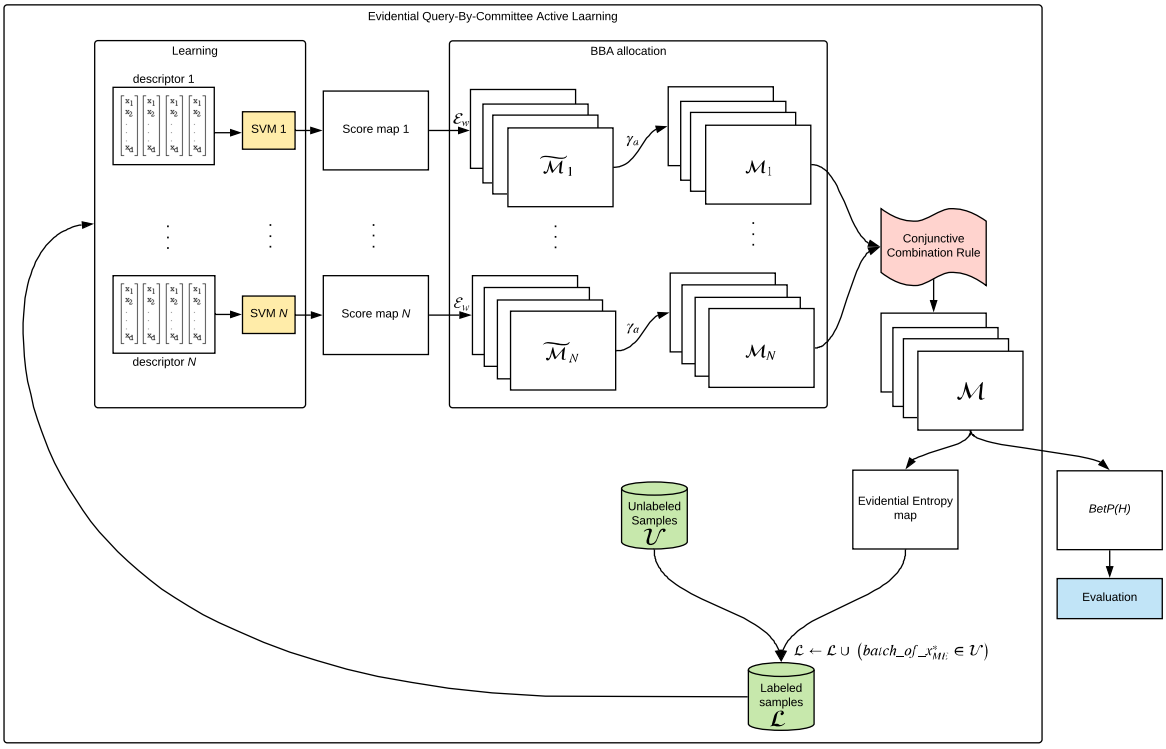


Figure 5.2: Evidential Query-By-Committee Active Learning flowchart.

the same batch, allowing to tackle different types of issues at the same time (i.e., conflicting and/or not committed BBAs).

5.3.3 Global overview of the proposed evidential QBC process

Figure 5.2 shows the complete flowchart of the proposed evidential QBC method, in opposition to Fig. 5.3 which shows the flowchart of the traditional approach. After the learning step, BF framework is involved in three operations, namely in the BBA allocation procedure through successive discounting, in the combination of sources that allows us to obtain a BetP(H) map used for evaluation, and in the derivation of evidential entropy map which guides the selection of the most informative samples to add to the training set for the subsequent iteration of the active learning procedure.

The proposed evidential QBC differs from the traditional one. First of all, from the score maps given by SVM classifications we do not derive probabilistic maps through logistic regression, but we perform a BBA allocation that takes into account two possible sources of imprecision, namely in the estimation of the sigmoid parameters to perform logistic regression and, later, in the image space. Then, the conjunctive combination rule is able to take into account the information provided by the different sources, discounted accordingly to their pixel-wise evaluated reliability. At this stage, the obtained BBA map $\mathscr{M}$ can be used either for evaluation, through the computation of the BetP(H) map, or to compute the evidential entropy map, from which the samples with maximum entropy are extracted and added to the labeled samples set $\mathscr{L}$. Note that in case of Maximum Ignorance or Maximum Conflict criteria, the evidential entropy map would not be computed, and the samples would directly be chosen maximizing ignorance and conflict channels, $\mathscr{M}(\Theta)$ and $\mathscr{M}(\emptyset)$ respectively.

The conjunctive use of BF both in the combination and in the derivation of the disagreement measures in the AL process allows us to overcome the limitation of traditional QBC where (optional) combination and disagreement computation are performed independently.

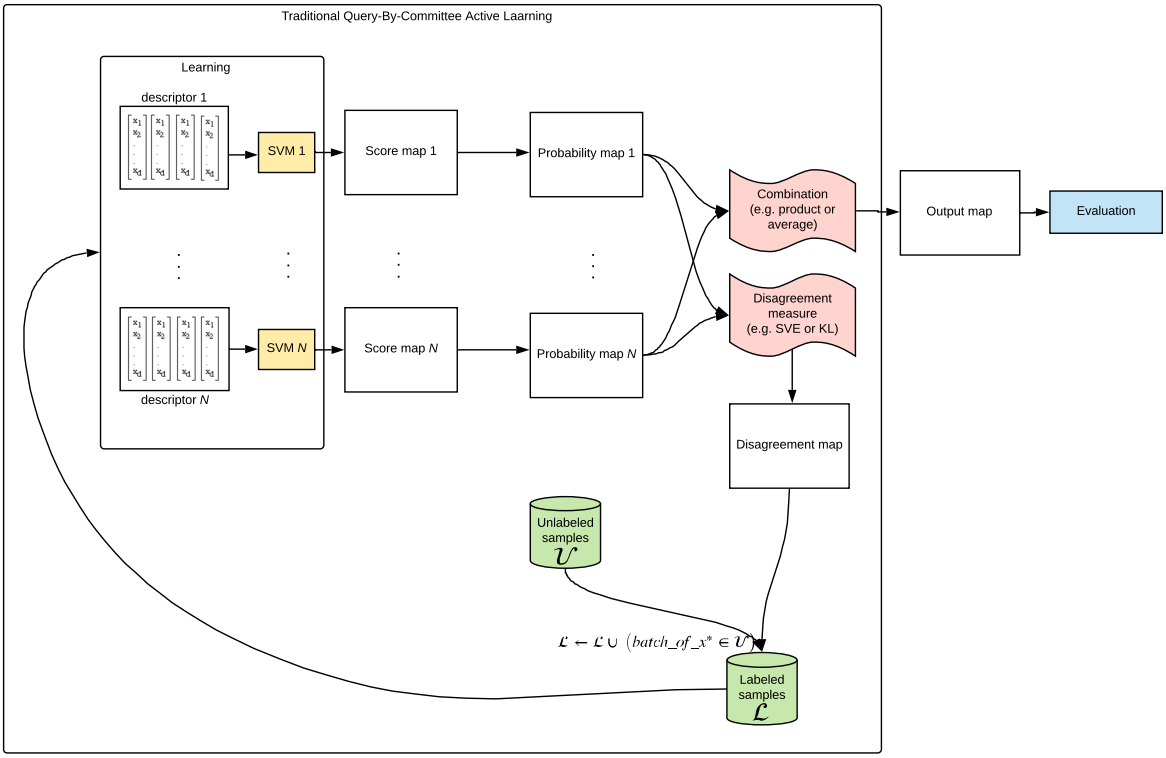


Figure 5.3: Traditional Query-By-Committee Active Learning flowchart.

In the following section, we will investigate all the proposed evidential-based disagreement measures as well as the traditional ones in the context of our application.

5.4 Experimental results

For the QBC algorithm, we thus build the committee $\mathscr{C}$ of classifiers with the four SVM pedestrian detectors, namely HOG, LBP, GABOR and DAISY, explained in Sec. 3.2. Such a committee is quite heterogeneous since each classifier contributes providing a different view of the data, so that the explained fusion strategy is applied at every iteration, both to obtain the image map of the BetP(H) on which we compute statistics, and to choose the samples to add to the training set on the basis of the different evidential-based proposed heuristics.

In the context of AL, the choice of the evaluation metrics is not trivial. The recent study carried out by [220] indeed have pointed out that most of the evaluations of AL approaches in the literature have focused on a single performance measure, and have shown that the improvements provided by AL for one performance measure often comes at the expense of another measure. Besides this, the most used metric is accuracy, which intrinsically depends on the choice of a threshold so that a question arises about how much of the observed improvement is due to the effective learning and how much of it is simply due to a shift in the optimal decision threshold. Moreover, accuracy metric is not relevant in presence of highly imbalanced data. To solve this last problem, popular measures are Precision, Recall, and F1 score, but they still require a threshold.

For all these reasons, we choose to evaluate our method on the basis of two different measures, which do not depend on a threshold and at the same time are suited in the presence of imbalanced data, namely AUPRC and PRBEP (i.e., the value corresponding to the point of the curve where Precision is equal to Recall). These two metrics are computed on the BetP(H) map, applying non maxima suppression (NMS) at every threshold to identify the targets (as done already in Chapter 4).

We conducted our tests starting from a random training set of 500 samples arriving up to 2000 samples, with a batch size of 100 samples per iteration added on the basis of the discussed disagreement measures. The pool of unlabeled samples $\mathscr{U}$ from which the active learning solution can choose the samples to add to the training set is composed by more than 760K samples, which are labeled according to the ground-truth map of the corresponding image.

5.4.1 Comparison between the proposed evidential disagreement measures

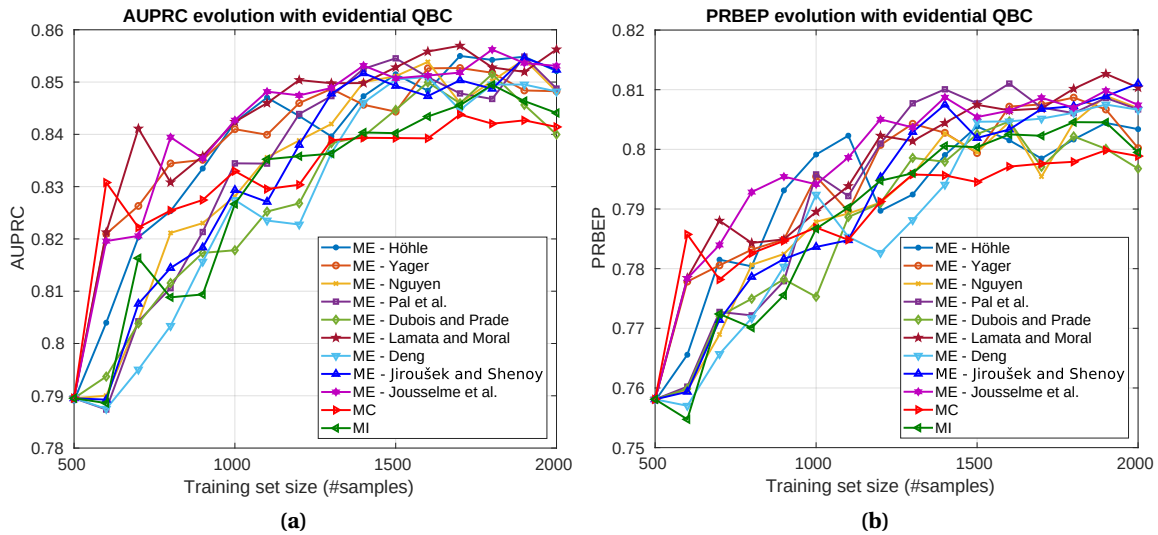


Figure 5.4: AUPRC and PRBEP at every iteration using ME criterion with different evidential entropy disagreement measures, MC and MI criteria.

Figure 5.4 shows the AUPRC and PRBEP for every iteration using the proposed Maximum Entropy (ME) with the different evidential entropy definitions, Maximum Conflict (MC) and Maximum Ignorance (MI) criteria. It is possible to see an improvement of both metrics with all the

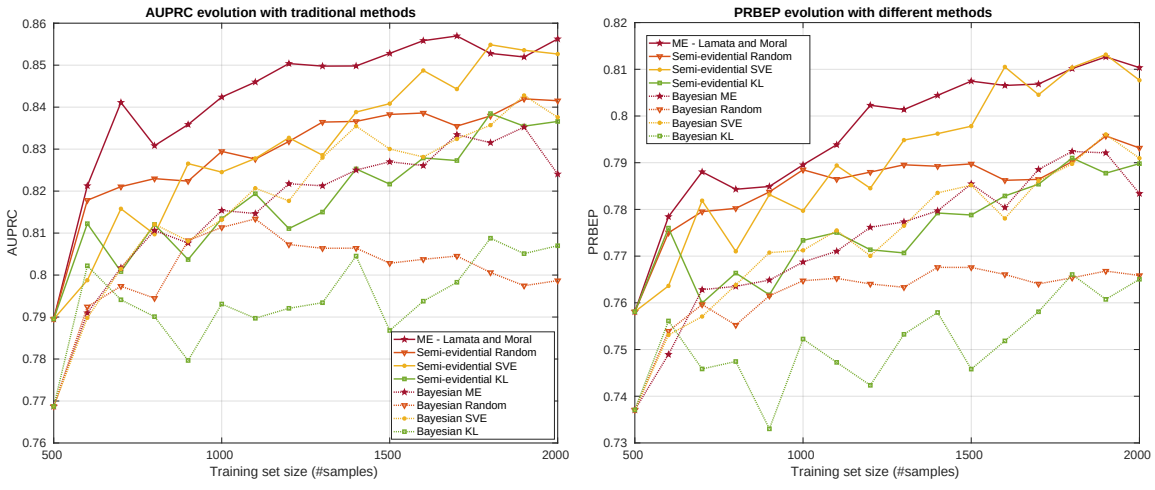
investigated disagreement measures, stressing the robustness of the method and the fact that the approach is well-suited to our application. All the curves tend to flatten towards the end of the process, which means that the final number of samples represents a suitable training set size.

We also note that some curves have higher performance even when relying on a small size of the training set (i.e., are faster to converge). This means that those query strategies are immediately able to select the most informative samples to add to the training set. There are indeed some differences among the results achieved using the various definitions. It is clear that entropy formulations which focus on conflict (e.g., the Yager one) provide better results with respect to Dubois-Prade definition which focus only on the non-specificity, already in presence of a small training set size. Moreover, considering both imprecision and conflict components seems to be beneficial, in particular using Lamata and Moral's composite definition. Note that also that the simple Jousselme's entropy-based criterion appears quite beneficial both in terms of AUPRC and PRBEP. In general, the best strategies appear to be the ones that encourage *diverse* samples inside the same batch in terms of BBA structure, that is to say, both conflict and ignorance components have to be taken into account, with a slight preference for samples with conflicting BBAs.

Considering the results obtained with the two simple evidential criteria solely based on conflict and ignorance, these approaches do not reach the performance of entropy-based disagreements. As expected, selecting the samples on the basis of maximum conflict allows for a steeper improvement at the beginning, where exploration of the version space is very important, but after some iterations the curves tend to flatten. On the contrary, the samples with high values of ignorance are mostly useful when the size of the training set begins to be consistent, and it becomes important to exploit the current feature spaces to adjust the boundaries. This behavior reflects the importance to pass from an initial exploration to a final exploitation of the data. To this extent, evidential QBC based on Maximum Entropy criterion is able to naturally find a trade-off between the two necessities.

In the following, we choose Lamata and Moral's entropy definition as the more competitive criterion among the evidential entropy formulations. Indeed, it outperforms other formulations when considering AUPRC metric, which is a key indicator since it takes into account the whole Precision-Recall curve, and at the same time has good performance in terms of PRBEP.

5.4.2 Comparison with traditional approaches



(a) **(b)**

Figure 5.5: AUPRC and PRBEP at every iteration. Comparison of evidential-based disagreement measures with traditional ones.

In order to evaluate the benefit for the active learning procedure of the proposed BBA allocation used in conjunction with evidential disagreement measures, Fig. 5.5 reports the AUPRC and

PRBEP curves related to two different levels of comparison.

Firstly, we evaluate the difference with respect to a result reached using purely probabilistic reasoning. We perform the Bayesian BBA allocation from the output of each classifier after Platt's regression, without applying any discounting neither in the score space nor in the image space, and we apply the normalized conjunctive combination rule (i.e. Dempster rule): in this way, the classifiers combination boils down to simple product of probabilities. Then, on the resulting probabilistic map, we apply the traditional SVE and KL disagreement measures, as well as a baseline that simply adds randomly drawn samples at every iteration. Moreover, to quantify exactly the benefit of the proposed BBA allocation over the Bayesian one, we aim at converting the proposed evidential disagreement measures to the Bayesian framework. MC and MI do not apply, since Bayesian BBAs have null masses on conflict and ignorance respectively. Transposing the evidential entropy definitions to the Bayesian framework, we notice that all the formulations (except Dubois-Prade's one which is always null being mostly related to ignorance, and Jiroušek and Shenoy's one) boil down to:

$$\mathrm{H}(m) = m(\mathrm{H}) \log\left(\frac{1}{m(\mathrm{H})}\right) + m(\overline{\mathrm{H}}) \log\left(\frac{1}{m(\overline{\mathrm{H}})}\right). \tag{5.8}$$

Curves related to this first comparison are the ones referred to as "Bayesian" in the plots' legend of Fig. 5.5. Clearly, evidential approach based on ME criterion (cf. "ME - Lamata and Moral" curves) outperforms all the probabilistic ones with respect to both AUPRC and PRBEP. Besides, the fact that there is a consistent gap between the proposed evidential Maximum Entropy and the corresponding curve in the Bayesian framework (Bayesian ME) indicates that the detector combination with the proposed BBA allocation is significantly superior to a simple product of probabilities.

In order to show that the performance gain is not only due to the relevant BBA allocation, but also to the good choice of disagreement measure for active learning, we propose a second type of comparison. Now, we perform indeed the proposed evidential BBA allocation, obtaining a BetP(H) map that we interpret in this case as a probability map to compute SVE, KL and the random baseline in a probabilistic framework. This allows us to focus on the benefit of the BF framework vs. probabilistic one only with respect to the new sample selection step, to see exactly the impact of evidential measures in the selection of the new samples being not biased by the detector combination result. The related curves are referred to as "Semi-evidential" in Fig. 5.5, since the BF framework is only involved in the BBA allocation and combination but not in the sample selection step.

Entropy-based criteria, namely Semi-evidential SVE and the proposed evidential ME, outperform the others, both in terms of AUPRC and PRBEP. However, although reaching almost the same performance as the evidential ME at the very end of the process, SVE is not able to select the most informative samples from the beginning, showing a much slower convergence. In particular, entropy-based evidential criterion results to be the best one, due to the ability of BF framework to model in a finer way the actual information contained in each sample, highlighting the importance of the coupling between the fusion of the classifiers and the definition of the disagreement measures.

We can notice how KL strategy, which was expected to select conflicting samples based on the consensus probability, does not seem to be very efficient in this context, performing even worse than random sampling both in the semi-evidential and in the Bayesian comparisons. This is against what we observed in the comparison of the various evidential-based entropies in Sec. 5.4.1, where the definitions that focus on the conflict are indeed the most successful ones. This fact shows that the evidential framework is more able to model the conflict among the various committee members, through the mass on $\emptyset$, with respect to the probabilistic framework that models it in terms of divergence from the consensus probability.

Finally it is worth remarking the fact that the proposed evidential ME strategy is able to reach the best performance obtained by the Random strategy using only half (or less) data (cf. "ME - Lamata and Moral" and "Semi-evidential Random" curves).

5.4.3 Correlation analysis

The aim of correlation analysis between the various disagreement measures is to understand better how they may differ from one another and the similarity between them. To this extent, we apply the proposed MC, MI and ME criteria with all the investigated entropy measures on the basis of the BetP(H) map obtained after BBA allocation and combination. Traditional SVE, KL and the random sampling are also performed on the basis of the BetP(H) map obtained after BBA allocation, interpreting it as a probability map, to focus only on the new sample selection step (following the semi-evidential approach).

Figure 5.6 shows the correlation matrix in terms of percentage of common samples between the different points selected at every iteration on the basis of the investigated criteria, excluding the initial common 500 samples, so that only the ones selected with respect to the various strategies are taken into account in the computation. We do not plot all the iterations but we focus on the first iterations, where variations are more visible, and on the last one in order to give a sight of the general behavior. We note that in general, going on with the iterations, the different training sets tend to diverge, sign that the size of the considered pool of unlabeled samples $\mathcal{U}$ is indeed appropriate in the sense that the various methods have enough freedom being not constrained by the data. Correlation is especially evident considering the various evidential disagreement measures. Many definitions are correlated to ignorance, and as expected, Dubois and Prade's entropy is very close to it. Yager's entropy and Lamata and Moral's one, on the contrary, are very correlated one to each other and have a consistent overlap with the conflict measure. Nguyen and Pal correlation is also highlighted, and it is easily explainable by the fact that Pal's formulation extends Nguyen's one, taking into account also the cardinality of the focal elements (in our case, in presence of two singleton hypotheses, only the term that refers to the compound set slightly changes). Again, Pal's training set seems very correlated to Jiroušek-Shenoy's and Deng's ones, which are two composite formulations aiming to take into account both conflict and non-specificity. KL divergence seems totally uncorrelated to any other measure, except for the conflict with a marginal degree.

To better understand the degree of correlation between the different measures, Fig. 5.7 shows a visual comparison of the maps obtained with the various entropy definitions for the first iteration of the AL process, so that we can compare them on the basis of the same training set. Figure 5.7a represents a selected part of the unlabeled samples pool $\mathcal{U}$. After the evidential combination of the classifiers, the result is the image map $\mathcal{M}$ of BBAs m_x associated to every pixel x, shown in terms of Belief in Fig. 5.7b, conflict in Fig. 5.7c, and ignorance in Fig. 5.7d. Soft Vote Entropy 5.7e and KL divergence 5.7f maps are shown as well for comparison with all the investigated evidential entropy measures. Once again, we notice the correlation between ignorance and Dubois-Prade entropy in Fig. 5.7k, while the other entropy measures seem more correlated to the conflict, although to different extents. Generally, evidential entropy maps are able to model in a finer way the actual information contained at every pixel locations, so that the regions of interest for the AL process are better enhanced with respect to SVE and KL.

The figure visually shows where and how the entropy measures correlate. While previous Fig. 5.6 provides only a global estimation of the correlation (scalar value), Fig. 5.7 allows for a qualitative visualization of the spatial variation of the correlation. Entropy is higher where the individual detectors are discordant, and the images show that this happens frequently on the border of the heads, because the various classifiers provide different detection sizes (e.g. HOG and GABOR provide more localized detections while LBP and DAISY provide coarser blobs). There are some areas that correspond to a head where entropy is high, and it means that just a part of the classifier committee succeeds in detecting it. We also note that some shoulders of the people may present high entropy values. Specifically, this happens when one or some classifiers miss-classify shoulders as heads due to their similar rounded visual appearance. Finally, it is interesting to visualize that the maps usually agree on the *location* of maximum entropy (borders of the heads, heads detected only by some classifiers, shoulders areas which confuse some classifiers), while at the same time they provide different *amounts* of entropy for the same location, and this is what allows the AL to choose different samples and thus to obtain such diverse training set at the end of the process.

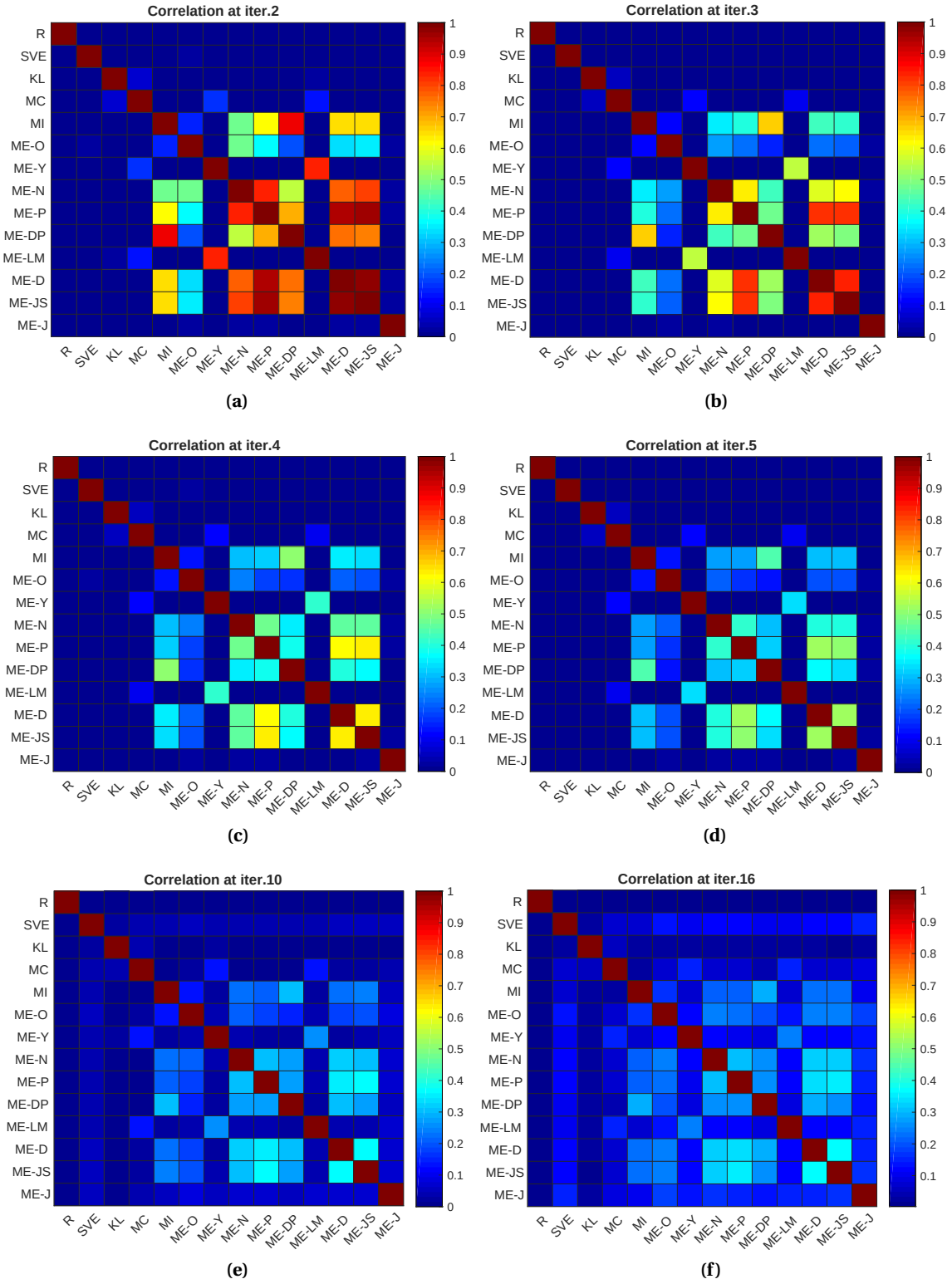
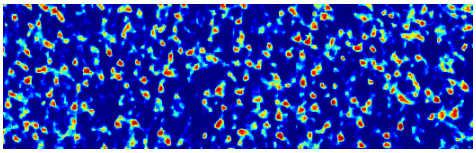
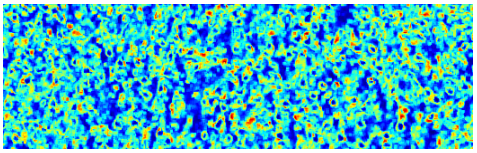


Figure 5.6: Correlation between samples added during successive AL iterations with different strategies, for the initial iterations and the last ones. R = Random, SVE = Soft Vote Entropy, KL = Kullback-Leibler divergence, MC = Maximum Conflict, MI = Maximum Ignorance, ME = Maximum Entropy: O = Höhle, Y = Yager, N = Nguyen, P = Pal et al., DP = Dubois and Prade, LM = Lamata and Moral, D = Deng, JS = Jiroušek and Shenoy, J = Jousselme.

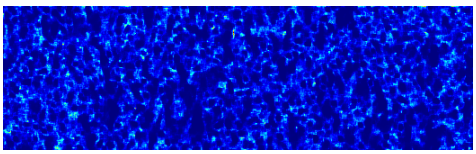
(a) Patch image from $\mathscr{U}$



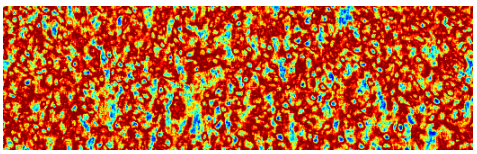
(b) Belief $\mathscr{M}(\mathrm{H})$



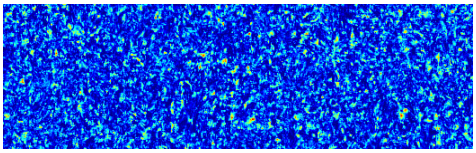
(c) Conflict $\mathscr{M}(\emptyset)$



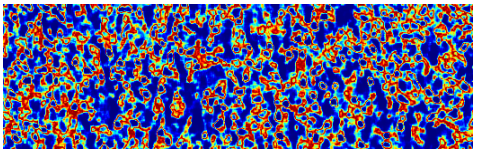
(d) Ignorance $\mathscr{M}(\Theta)$



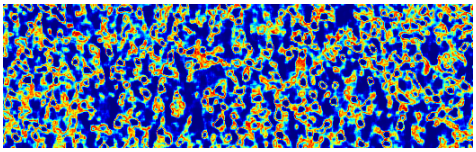
(e) Soft Vote Entropy



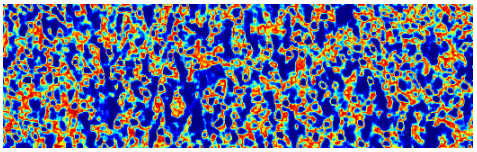
(f) KL divergence



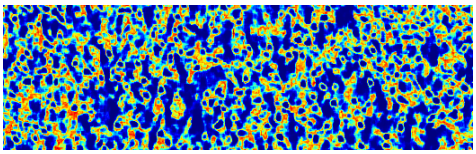
(g) Höhle entropy



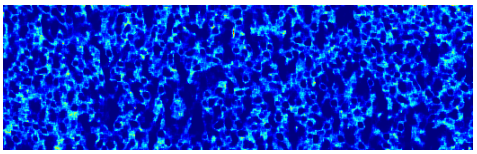
(h) Yager entropy



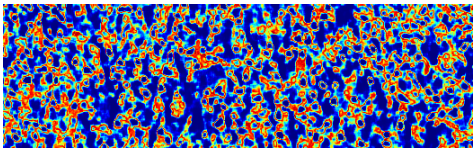
(i) Nguyen entropy



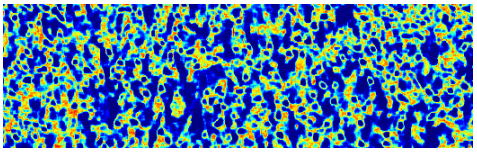
(j) Pal entropy



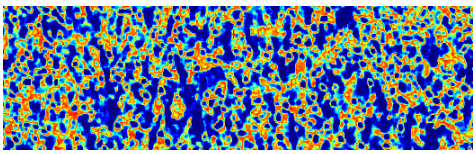
(k) Dubois and Prade entropy



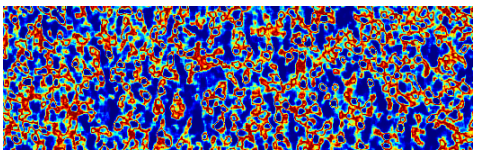
(l) Lamata and Moral entropy



(m) Deng entropy



(n) Jiroušek and Shenoy entropy



(o) Jousselme entropy

Figure 5.7: Different maps obtained using the investigated evidential disagreement measures for a selected patch of the unlabeled samples pool $\mathscr{U}$ (in Fig. (a)) with corresponding BBA allocation $\mathscr{M}$. SVE and KL maps are shown as well for comparison.

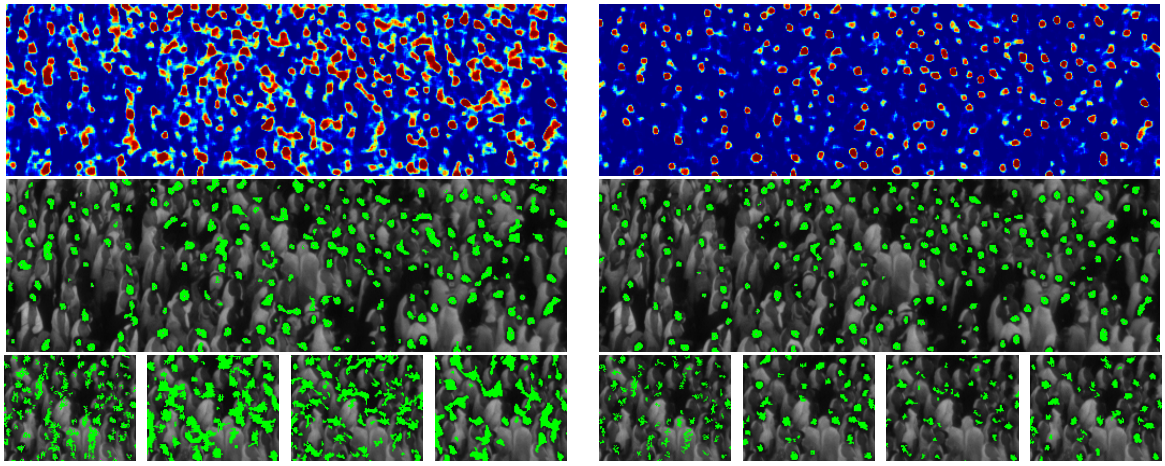


Figure 5.8: Visual comparison of the detections obtained at the first iteration of the process (500 training samples), on the left, and the last iteration (2000 training samples selected using Lamata and Moral Maximum Entropy criterion), on the right. Results are shown in terms of colormap of the BetP(H) map in the first row and detections at PRBEP in the second row. Small patches with the different sources involved in the combination are shown for reference in the third row (namely SVM classifier with HOG, LBP, GABOR, DAISY descriptors).

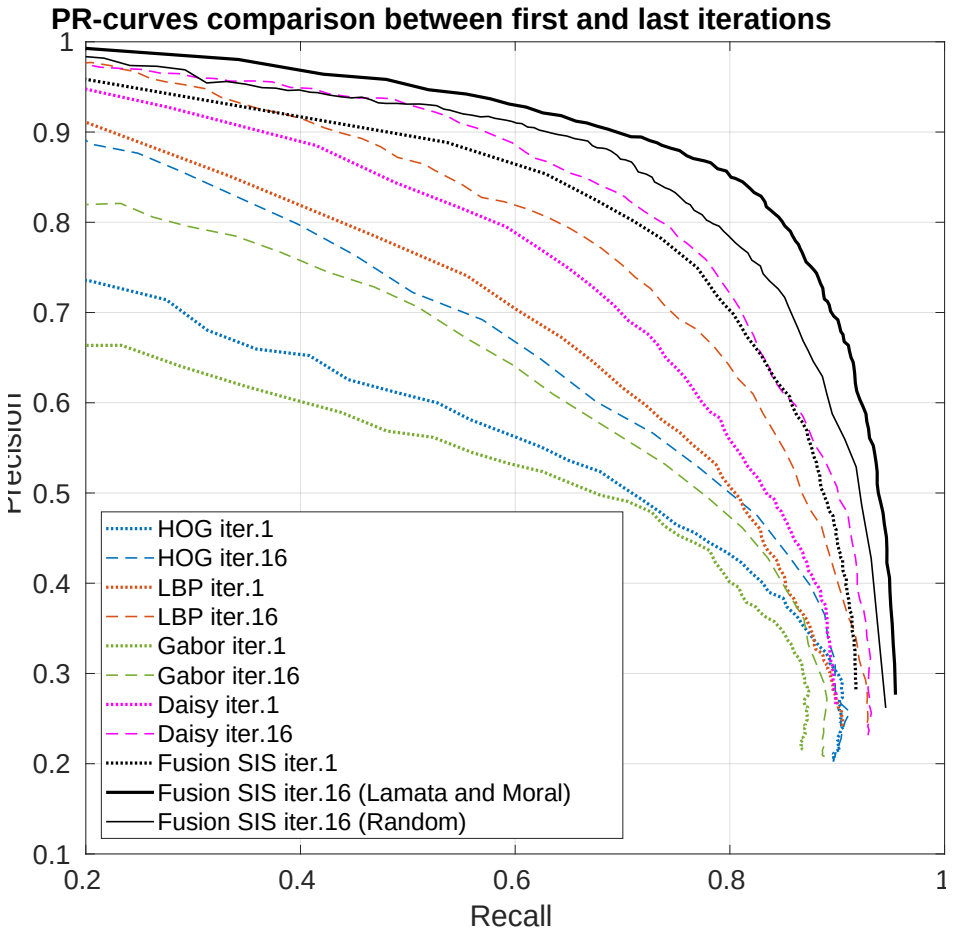


Figure 5.9: PR-curves for the individual classifiers, as well as the fusion between them, for the first and the last iterations of the AL process. PR-curves for the single descriptors are reported as well to see their evolution thanks to the active learning procedure. For the sample selection, we compare Lamata and Moral's strategy with the random selector (which benefits only from a larger training set).

5.4.4 Global benefit of evidential QBC active learning

Figure 5.8 provides a visual comparison between the first and the last iterations of the process, during which the training set increased from 500 samples (on the left) to 2000 training samples (on the right), selected with the Maximum Entropy criterion using Lamata and Moral's definition. The classification results are shown both in terms of colormap of the BetP(H) (soft detections) in the first row, and detections at the PRBEP threshold in the second row. Moreover, detections using the individual sources that compose the committee of classifiers are shown in the last row (HOG, LBP, GABOR, DAISY respectively), in order to highlight their complementarity and the necessity of an adapted fusion between them. While the colormap is useful to identify regions with higher values, and to immediately see that at the end of the process we obtain a less noisy and sharper map, the detections superimposed on the input image are indeed useful to evaluate the actual location of the detections and the presence of false positives (areas with high values which do not correspond to an actual head) or false negatives (heads which are not detected).

The detections are provided here for the value of threshold at which precision is equal to recall (i.e. the PRBEP), which is a reasonable compromise since it allows us to have the same number of false positives and false negatives. PRBEP is equal to 0.74 for the first iteration, meaning that at the beginning of the process for this particular threshold 26% of the heads are lost while at the same time 26% of the detections are not actual heads. At the end of the process, PRBEP becomes 0.835, meaning that we obtain an improvement of almost 10% with the proposed approach, both in terms of precision and recall.

PRBEP threshold is a traditional operative point for many applications and we find it reasonable to adopt it for visualization purposes. The exact values of the thresholds are $th = 0.8$ for the first iteration (on the left) and $th = 0.55$ for the last iteration (on the right). Although the exact values are not really meaningful in themselves, it is interesting to notice the initial bias toward a high threshold, that can be explained by the fact that at the beginning of the process the training set is balanced (i.e. it has the same number of positive and negative samples), while at the end of the process it tends to have more negative samples, reflecting the actual data distribution.

Overall, the proposed evidential fusion, which is able to take into account imprecision both during calibration and in the image space, results to be suited for this application. Besides, the AL algorithm is able to select samples which are indeed useful to improve the performance of all the classifiers, a fact which results in a significant and visible improvement of the final BetP(H) map. At the end of the AL process, the detections are more localized, sharper and a lot of false positives which were present at the beginning have been successfully removed.

To highlight the importance of coupling the fusion strategy with the AL process, Fig. 5.9 shows different Precision-Recall (PR) curves, for the various single classifiers as well as for their fusion. The curves are shown for the first and last iterations, in order to illustrate the relative improvement in terms of performance for all the classifiers.

Besides showing that fusion results are better than individual detectors (which is not mandatory true), the figure has two main purposes. Firstly, it shows the improvement due to AL, comparing the first iteration with the last one for every classifier and their fusion, so that we can see that AL is effective since performance has increased for every classifier at the end of the process. Secondly, considering the fusion result, it shows that the improvement is not only due to the increased size of training set but also to the chosen sample selection strategy. The image underlines indeed the consistent gap between the two fusion results at the last iteration, which corresponds to random sample selector and maximum entropy sample selector (considering Lamata and Moral's entropy definition). This fact underlines the importance of having defined an adapted fusion strategy which is able to take into account imprecision while at the same time providing clues for the AL process.

Chapter 6

CNNs for pedestrian detection in high-density crowds

Contents

6.1 Motivation and related works

In the object detection community, deep learning emerges in the last years as an alternative approach to established methods working with hand-crafted features. Nevertheless, it frees from the necessity of feature engineering, even though this comes at the expense of larger datasets usually needed for training, and the impossibility to easily interpret the results.

Despite the success of deep learning however, as stated in [294], the particular task of pedestrian detection is still a difficult problem and hand-crafted features still appear to be of critical importance. They are indeed intrinsically designed to obtain finer resolution with respect to commonly employed networks which fails at detecting small details mostly due to the presence of pooling layers. Neural networks are therefore usually used in conjunction with traditional methods like ICF detector [110, 257], and even when used alone [158, 295] the variability of scale and appearance of the various individuals is so high that the task is far from being solved. A very recent work [286] exploits the concept of Omega-shape which is learned in a deep framework to handle partial occlusion. However, this work and in general all state-of art methods in pedestrian detection rely on a region proposal step to isolate the targets, but in presence of dense crowds it becomes inapplicable due to the large number of people.

For the similar context of face detection, the recent work presented in [112] proposes a CNN able to detect faces at extremely small scales, by rescaling the input with respect to different scaling factors and merging the different response maps to obtain the final detections. The method has shown to be very robust to small scale objects, blur, and partial occlusions, but it is still targeted to a different problem with respect to our application. Indeed, in face detection the facial features play an important role to increase the discriminative power of the detector.

Recently, for the task of spatially dense classification, Fully Convolutional Networks [169] (FCNs) emerges. They are a particular type of CNN especially used for semantic segmentation applications, where each pixel of the image has to be labeled with respect to the object it belongs, instead of just obtaining a unique label for all the image as it is done in classification. For this reason, we cast our problem of head detection as a segmentation task in presence of *soft labels*, since we do not have information about the precise contour of every head.

In the following, after a review about CNNs and architectures for semantic segmentation using FCNs, we will formulate our problem in terms of segmentation using soft labels and show its relevance in terms of results with respect to two different architectures, a state-of-art UNet [229] and a proposed architecture inspired by [97].

6.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a particular type of neural networks (cf. Sec. 2.2.5) which make explicit assumption about the fact that input data are images, so that they can better encode spatial constraints and reduce the total amount of parameters in the network.

CNNs are still composed by several layers containing neurons with weights and biases which are learned by minimizing a loss function through backpropagation. They also still present a fully connected layer as last layer where a classifier (SVM or Softmax) is used to provide the object label for the task of classification.

Regular NNs involves the use of fully connected hidden layers, where each neuron is fully connected to all the neurons of the successive layer. However, this approach do not scale well in presence of inputs in the form of images, as the number of parameters of the network would soon explode and become intractable. Moreover, full connectivity possibly leads to overfit.

On the contrary, CNNs constrain the architecture in such a way that their layers have neurons arranged in a 3D volume of given *width*, *height*, and *depth*. Moreover, neurons in a layer are only connected to a small region of the layer before, so that local consistency of image data is exploited. Every layer of a CNN transforms thus by means of a differentiable function an input 3D volume into an output 3D volume which is usually deeper, until the last fully connected layer which is employed for classification.

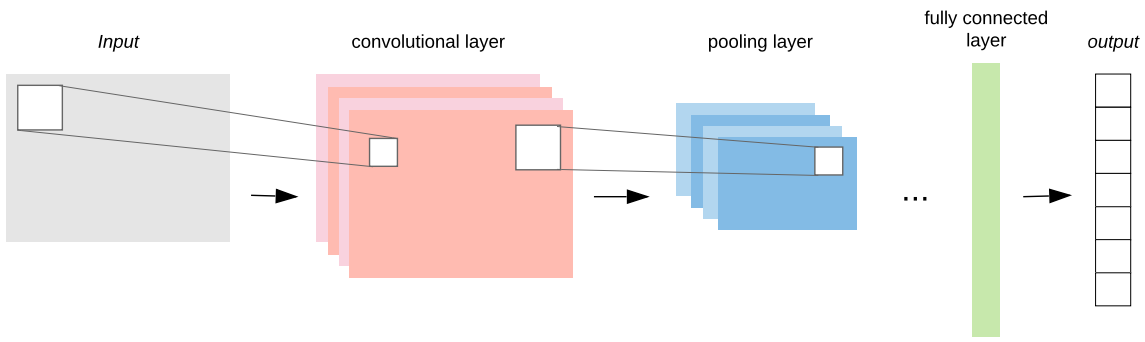


Figure 6.1: Visual example of a CNN. The input volume (bidimensional in the image but usually 3D) is convolved with a set of kernels (i.e. learnable weights, 4 in the example, applied at every pixel location assuming stride=1). The spatial dimension of the resulting 3D volume is then reduced through pooling, while fully connected layer is used for classification (in the example the output is assigned to 7 different classes).

6.2.1 CNN layers

Specifically, there are mainly three different types of layers which can be stacked to compose a CNN architecture: *convolutional layers*, *pooling layers* and *fully connected layers*, as depicted in Fig. 6.1. Let us now examine the different layers along with their different purposes.

6.2.1.1 Convolutional layers

Convolutional layers are the main building blocks of a CNN. The parameters of a convolutional layer consist of a set of learnable weights, or filters, with small spatial extent along width and height dimensions but extending through the full depth of the 3D input volume. During the forward pass, each filter slides through the input over the width and height dimensions and is convolved with the input volume itself. This convolution operation produces a 2-dimensional *activation map* that gives the responses of that filter at every spatial position. To produce the 3D output volume, the activation maps given by every filter are stacked along the depth dimension. The number of filters used in one layer is a hyperparameter that corresponds thus to the depth of its 3D output volume.

Note that each neuron is connected to only a local region of the input volume. The spatial extent (along width and height) of this connectivity is the *receptive field*, or *filter size*, that is a hyperparameter which is usually set to quite small values (i.e. 3×3 or 5×5). It has been nevertheless shown that it is better to stack more layers with small filters than using less layers with larger filters, thanks to the activation functions at every layer which increase the non-linear response ability of the model. Still, the extent of connectivity along the depth axis is equal to the depth of the input volume: connections are local in space, but always full along the entire depth of the input volume. In this way, increasing the number of filters used throughout the layers, we increase the output depth.

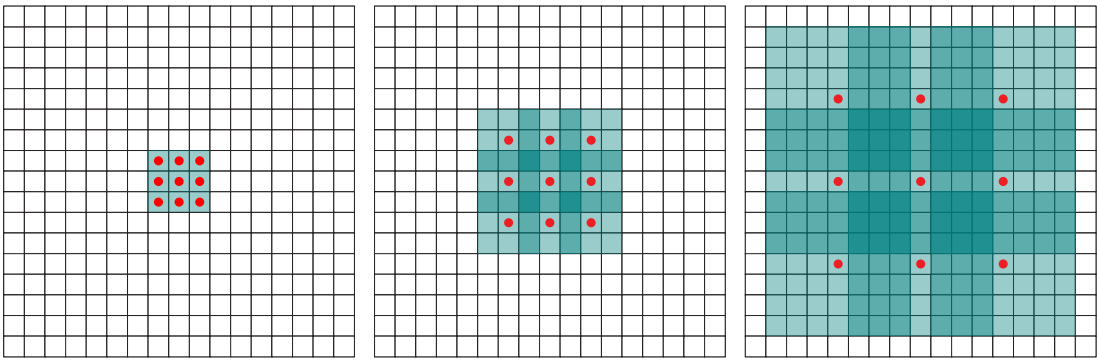


Figure 6.2: Example of dilated convolutions. Red dots specify the cells where the filter is applied, while green cells highlight the receptive field. (a) 1-dilated convolution (3×3 receptive field); (b) 2-dilated convolution (7×7 receptive field); (c) 4-dilated convolution (15×15 receptive field). Image taken from [290].

Other hyperparameters to set are the *stride* at which the filters are slided through the input (larger strides will produce spatially smaller output volumes), and the *padding* which allows us to pad the input volume around the borders so that the spatial size of the output volume can be controlled and preserved with respect to the input (usually, zero-padding is used). Then, in [290], *dilated convolutions* have been introduced specifically for dense prediction. They systematically aggregate multi-scale contextual information without losing resolution, by supporting an exponential expansion of the receptive field through the layers without loss of resolution or coverage. Figure 6.2 shows an example of dilated convolutions, highlighting different patterns of filter application with respect to the dilation size. Note that the number of parameters associated with each layer does not change with respect to the dilation size, while at the same time the receptive field grows exponentially.

The second important feature of CNN with respect to regular networks is *parameter sharing*. Indeed, by the assumption that for image inputs the same feature is useful to be computed at every different spatial positions, each filter has the same parameters throughout the spatial volume. In other words, as the filter moves around the input volume the same weights and biases are being applied in the forward pass and learned during backpropagation. Each filter therefore is useful to perform a certain transformation across the whole image (this is in contrast with fully connected neural networks, which can have different weight values for every connection). Denoting a single 2D slice of depth as a *depth slice*, all neurons in a single depth slice are using the same weights and the forward pass is computed for every depth slice as a convolution of the neuron's weights with the input volume (this is why the layer is called "convolutional", and why the set of learnable weights are referred to as "filter" or "kernel").

Parameter sharing along with the use of a receptive field makes the total number of parameters of a CNN to be far less than the ones of fully connected networks.

6.2.1.2 Pooling layer

The function of the pooling layers is to progressively reduce the spatial size of the representation, decreasing thus the amount of parameters and computation required by the network. They are usually inserted after the convolutional layers and operates independently on every depth slice of the input. Pooling layers perform a spatial downsampling of the volume, preserving the volume depth.

The most common pooling function is a MAX operation, which is performed in a sliding-window fashion throughout the input volume, and gets the biggest value on the window as output, although also average or sum pooling exists.

6.2.1.3 Fully connected layers

Fully connected layers are called like this because their neurons have full connections to all activations in the previous layer. They are usually placed at the end of the network, to perform classification, so that the general layer patterns of a CNN becomes:

$$Input \rightarrow \Big[[conv \rightarrow activ] \cdot \mathrm{N} \rightarrow pool \Big] \cdot \mathrm{M} \rightarrow [\mathrm{FC} \rightarrow activ] \cdot \mathrm{K} \rightarrow \mathrm{FC}$$

where $conv$ represents a convolutional layer, $activ$ the activation function (cf. Sec. 2.2.5.3) - usually a ReLU, $pool$ the (optional) pooling layer, and FC the fully connected layer, while N, M and K are the numbers of repeated layer patterns within brackets.

While the series of convolutional layers along with their activation functions and possibly pooling operations allows us to learn a set of meaningful features, fully connected layers placed at the end learn non-linear combinations of these features.

6.2.2 Fully Convolutional Networks for semantic segmentation

Fully Convolutional Networks (FCNs) have been proposed in [169] as particular types of CNNs for the task of segmentation, as they allow us to perform dense per-pixel predictions by taking input of arbitrary size and producing correspondingly-sized output. Typical classification networks take fixed-sized inputs and produce non-spatial outputs, by assigning a single class (label) to the whole image, whereas FCNs allows us to obtain a dense output map where a different label is assigned for every pixel of the input image.

The name comes from the fact that they contain only convolutional layers (along with their non-linear activations and possibly pooling), but no fully connected layers at the end of the layers' chain. In this way, there is no necessity of performing any patchwise training or region proposal. On the contrary, each pixel of the image is used for training, and in the same way at inference time a prediction score is obtained for every pixel of the testing image. This allows us to take into

account the full context of the image, which is an advantage with respect to patch-based CNNs both in terms of efficacy and efficiency.

The structure of a FCN is that of an *encoder-decoder*: initial encoder layers produce low resolution representations of the input data in its context, while the decoder part recover spatial information refining the spatial precision and localization of the output. While the encoder part is rather similar with respect to the different proposed fully convolutional networks (e.g. a VGG-like architecture [244] deprived of its final fully connected layers), the decoder part varies more.

In the original FCN [169] the decoder is composed by a stack of deconvolution layers (along with their activation functions), with learnable parameters. This allows for a non-linear upsampling learning, but increases the number of total parameters. SegNet [10] proposes a decoder that uses pooling indices computed in the max-pooling step of the corresponding encoder's layer to perform non-linear upsampling, eliminating the need of learnable parameters. Recently, DeepLab [38] achieves very good results by combining FCN with conditional random fields (CRFs).

In the context of medical image analysis, U-Net [229] has been proposed and shown to be very effective also in presence of small training datasets. In order to perform precise localization, high resolution features from the contracting (encoder) path are concatenated with the corresponding opposing learnable deconvolution layers of the expansive (decoder) path, through a mechanism called *skip connections*.

6.3 Head detection in high-density crowds

We choose to cast our specific problem of head detection in high-density crowds as a segmentation task, in the sense that we want to assign a different label (foreground or background since it is a binary problem) to each different pixel of the image, depending whether or not it belongs to a head. Given an input image, we aim thus at performing *dense* prediction by estimating an output map of the same size of the input where pixels that belong to a head are labeled as foreground.

This problem can be indeed related to two different applications of image segmentation, i.e. natural image segmentation for scene understanding and medical image segmentation. Our images represent real-world rich environments, like in natural images, while at the same time presenting small and cluttered objects to detect, like cell nuclei.

However, attention must be payed with respect to two different concerns related to our application, namely the impossibility to obtain a precise ground-truth map and the impossibility to have huge labeled datasets at our disposal. These two aspects will be investigated in the following.

6.3.1 Soft labels definition

Semantic segmentation is different from the usual classification task in one important way. Given an input image, classification techniques assign a unique label to the entire image, whereas segmentation assigns a different label to each pixel of the image. To do so, usually ground-truth maps are available at pixel resolution, with sharp boundaries between one class and another based on the contour of the various objects present in the scenes.

In our specific environment, precise labeling to perform head detection is usually impossible to achieve, due to the presence of clutter and occlusion problems that make the contour of the heads barely distinguishable from the background, in addition to the very small size of the targets. A precise definition of head borders is thus difficult even for a human operator, beside being a tiresome and time consuming task (in a single image almost thousand heads should be precisely pixel-wise labeled for the considered dataset). Moreover, commonly used datasets do not come with precise segmentation ground-truth but rather with just a list of coordinates that indicate the center of the heads.

For these reasons, we investigate the problem of head detection from partially labeled data, namely where only the center of each head is dot-annotated as explained in Appendix A, with only

a prior knowledge about the average radius of a head in pixels (possibly with respect to its location in the image in case of strong perspective variation due to camera tilt).

Starting from the dotted annotation, we could derive a segmentation ground-truth by simply performing a dilation with a circular structuring element centered in each annotation location in the image space. This would result in a binary map with circular blobs whose pixels would take a value of 1. However, this approach presents many drawbacks. Firstly, heads are not precisely circular and regular, so that the circular approximation of the contours would introduce errors at the boundaries. Secondly, in presence of strong occlusions, head contours would not have a circular shape as part of the head would be covered by another one. Lastly, in presence of many close heads we would obtain a single blob in the ground-truth map, loosing the information about the number of heads gathered together along with their center locations.

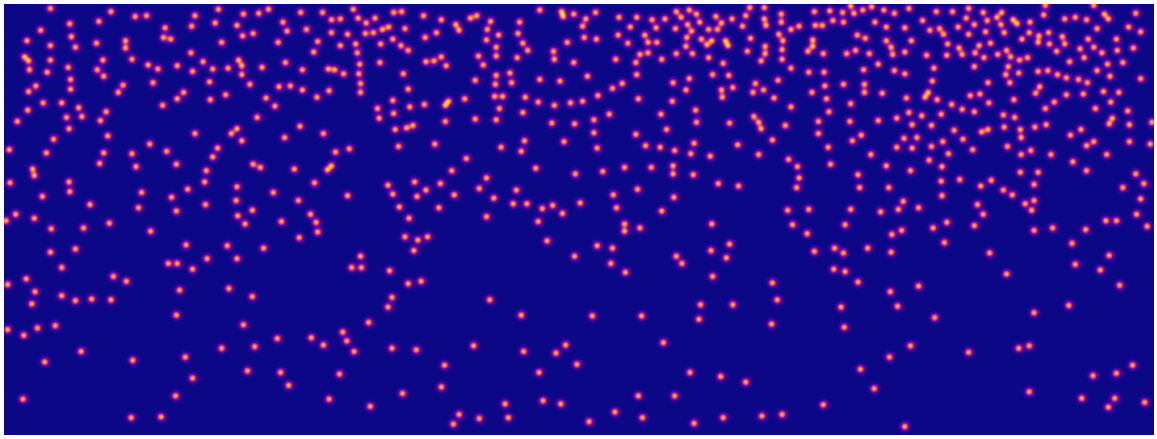


Figure 6.3: Ground-truth map as cumulative Gaussian distributions, one per head. The score associated to each pixel of the ground-truth map is the sum of the contributions of each Gaussian at the given location. In the image, scores span from blue (low) to yellow (high).

To face these issues, we propose a *soft label* definition of the ground-truth map through the use of cumulative Gaussian distributions. Starting from the binary ground-truth map with 1-valued label for each head center location (x_c, y_c), we apply a cumulative Gaussian smoothing such that the ground-truth map for each head is expressed in terms of a Gaussian distribution as:

$$(x, y) \sim \eta \cdot \mathcal{N}\left((x_c, y_c), \sigma_h\right), \tag{6.1}$$

where η is a scaling factor to face the class imbalance problem, while $2\sigma_h$ is the expected head radius.

We consider Gaussian distributions as they are infinitely differentiable functions presenting tails which vanish at infinity, being able to model well the uncertainty about the precise head contour locations. We apply a *cumulative* Gaussian smoothing in the sense that the final ground-truth map is the sum of Gaussian distributions derived from each head center locations. The resulting map is not a probability distribution by itself, but rather the score associated to each pixel represents the sum of probabilities that any head, occluded or not, is located at that position, directly facing in this way also the problems of close and occluded heads. In presence of close heads indeed, maxima would still indicate the head center locations, while in presence of occluded heads the evidence of the partially visible head will be reinforced through the cumulative sum. Figure 6.3 shows an example of ground-truth map obtained with the proposed soft labels.

While performing dotted annotation is much more efficient than performing precise segmentation labeling, it can still be imprecise. It is nevertheless difficult even for a trained operator to precisely indicate the center of each head present in the scene. Ground-truth Gaussian smoothing is thus also able to mitigate location errors in the annotated ground-truth, that could have a higher impact considering sharp-defined objects in presence of small targets (for instance, consider the impact of a mislocation error in the ground-truth of just two pixels in presence of a five

pixels radius head).

To summarize, we find that the introduction of soft labels through cumulative Gaussian distributions is beneficial for several reasons:

- It avoids a clear definition of the borders which is practically impossible for such difficult applications, while at the same time allowing us to obtain a spatial extent of the heads;
- It helps in presence of occlusions by reinforcing the evidence of the presence of an occluded yet partially visible head, since the Gaussian distributions are summed up in each pixel;
- It allows us to easily retrieve the various head center locations in presence of close heads that would result in a single blob, thanks to the Gaussian maxima, i.e. it provides an efficient way to perform *instance segmentation* (explained later);
- It mitigates the impact in the loss function of mislocation errors in the ground-truth head locations;
- It allows us not to loose the information about the number of people present in the image thanks to the cumulative sum of contributions.

Note that the last point is particularly important because it allows us to perform people counting directly from the obtained output map, by simply performing an integration over the entire image or region of interest. This will be explained further in Chapter 8.

6.3.2 Data augmentation

Another important aspect that must be inspected while applying deep learning methods is the availability of large training datasets. Like with all the other supervised learning techniques indeed, a huge number of training data reflects in a better learning capability. This fact is particularly noticeable in presence of very *deep* neural networks, where the high discriminative power comes at the expense of a high risk of overfitting especially in presence of small training sets.

As already pointed out, in the context of high-density crowd analysis the datasets are usually small since data may be difficult to acquire and are nevertheless hard to label for a human annotator. In order to be able to apply deep learning techniques in presence of such small labeled datasets, special care must be taken not to run into overfit while at the same time performing a robust training.

Besides the various traditional techniques such as parameter regularization or the more recent dropout technique (cf. Sec. 2.2.5.2), a powerful way to prevent network overfitting is called *data augmentation*. It allows us to exploit the available images in the most efficient way. To cope with small training datasets indeed, synthetic new data are created starting from the available ones, applying several different modifications to the original images.

The motivation behind it resides in noticing that the available data have been necessarily acquired under a limited variety of conditions (e.g. illumination, orientation, scale, ...). Data augmentation is thus capable of improving the invariance and robustness of the network to various conditions. The dataset is artificially augmented with modified versions of the original training data, making the network more invariant to objects slightly different than the ones encountered during training.

The specific types of augmentation to be applied differ with respect to the considered application. For example, landmark perturbation is often applied in face recognition problems, but is not applicable in the context of pedestrian detection where facial features of the people are not visible. Again, vertical flip (or similar 180° rotation) is usually applied in traditional object classification problems but it is not meaningful in our context since pedestrians in (moving) crowds are standing up on their feet.

We therefore apply horizontal flip, Gaussian noise, salt and pepper noise, brightness and contrast changes. The type of augmentation is applied randomly at each training epoch with a given

probability, so that only a subset of them are performed together. Finally, note that since we are dealing with a segmentation problem, any data augmentation procedure that performs an affine transform or flip must be applied to the ground-truth map as well.

6.3.3 Loss function

As explained in Sec. 6.2.2 there exist several FCNs suited for the task of semantic segmentation. In order to be able to apply them to our specific problem however, some modifications are necessary.

FCNs usually cast segmentation as a dense *classification* problem, in the sense that each pixel is assigned to a given class (where the number of classes is discrete). For this reason, they commonly employ loss functions suited for this task, e.g. cross-entropy (weighted, in case of class imbalance). We are rather interested in performing *regression*, since after the cumulative Gaussian soft labeling seen in Sec. 6.3.1 the pixels of the ground-truth map (and thus our desired output) are not labeled with their class but rather with a real value resulting from the accumulation of head distributions. Since output values are not discrete (disregarding the unavoidable discretization of the Gaussian function over the pixel domain) and possibly not bounded, we choose to use a L2 loss, as a straightforward yet efficient pixelwise estimate of the distance between two 2D maps.

Note that the parameter η of Eq. (6.1) is equivalent to weight the loss for the positive class for classification problem employing weighted cross-entropy loss. The parameter η is particularly important since higher its value, higher the impact of each single pixel belonging to a head in the loss function, and must be set taking into account the expected crowd density (lower the density higher its value, as per-pixel class imbalance would be more relevant).

6.3.4 Network architectures

Among the various architecture for semantic segmentation, we tested the U-Net [229], that has been proposed in the biomedical field and is now a state-of-art network, and we propose a network inspired by [97] that makes use of dilated convolution to be able to recover small objects, proposed in the field of remote sensing imagery. Note that we tried to use also UResNet [95], an encoder-decoder network inspired by both U-Net and ResNet using residual blocks, but the training data at our disposal resulted to be not enough to train this type of network with a too high number of parameters.

6.3.4.1 U-Net

The U-Net architecture [229] is a state-of-the-art segmentation network which has been originally introduced for medical image segmentation. We have chosen to use this network because it has been shown to be very effective even relying on small training datasets, used in conjunction with a massive data augmentation that makes an efficient use of the available training data, while at the same time being efficient at inference time.

The U-Net, whose structure is presented in Fig. 6.4, is built upon the original FCN [169] that introduced the idea of transposed convolutional layers (also called up-convolutions or deconvolutions) in order to reconstruct the output map. However, it improves the latter in some aspects.

Firstly, it presents a symmetric structure between the encoder part, which performs feature extraction by taking into account context information, and the decoder part, which reconstruct the output map. The use of pooling layers in the contracting path reduces the spatial extent of the 3D volume, which is then increased in the expansive path in a mirrored way by means of learnable deconvolutions, up to the original size of the input image. This gives to the architecture the peculiar "U-shape", hence the name U-Net.

The use of pooling layer allows for an important reduction of the number of parameters of the network, contributing to the efficacy of the network even in presence of small training datasets being robust to overfitting. Note that the original paper proposes to use learnable deconvolutions in the expansive path, but other implementations of the U-Net propose to use simple upsampling,

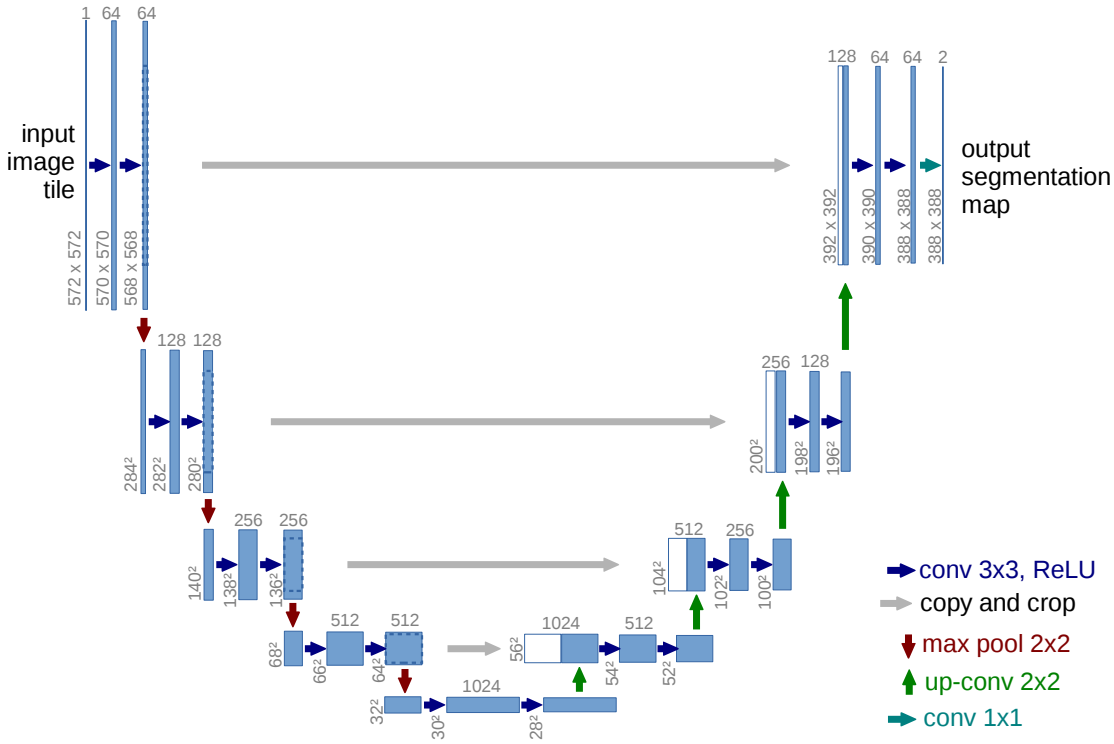


Figure 6.4: U-Net architecture. The "U-shape" (hence the name) is given by the encoder-decoder structure containing a descending path for context extraction (encoding), and an ascending path for output map reconstruction (decoding). The grey arrows represent the skip connections which allows for the combination of upsampled reconstructions and encoded feature maps.

which has no parameters and allows for an even more extreme reduction of the number of total parameters. The downside however is that this latter do not exploit weights to combine the spatial information in a smart way, so transposed convolutions can potentially handle more fine-grained detail. This is why we rely on the original architecture, since in our application we have to detect extremely small objects.

Another novelty introduced by the U-Net is the use of skip-connections. The result of each up-convolution in the expansive path indeed is concatenated with the corresponding opposing features obtained in the contracting path. This operation is essential to avoid producing too coarse output maps and to perform precise localization of the target objects.

6.3.4.2 FE+LFE

In the context of remote sensing image analysis, the authors of [97] highlighted a major problem of segmentation in presence of small and densely aggregated objects. The use of pooling layers indeed tends to degrade the output resolution so that details of the very small objects are lost. In these situations, even the use of shortcuts like skip-connections in the U-Net could not be enough to recover small targets.

Pooling layers are however important, for two different reasons. Reducing the spatial dimension of the 3D volume, they allow for a larger context consideration without increasing the receptive field of the filters, and to reduce the number of total parameters to be learned by the network.

In order to enlarge the receptive field of the filters going deeper with the layers, without degrading the output resolution nor increasing the filters' size (thus increasing the number of parameters), dilated convolutions can be exploited. Linearly increasing the dilation factor through the layers' chain will result in an exponential enlargement of the receptive field that is therefore able to capture larger context and recover smaller objects. Context information is indeed crucial in recovering small objects, as pointed out in [112].

The authors of [97] however noticed that aggressively increasing dilation factors through the

	Layers
FE	Conv 3 × 3, F = 16, D = 1
	Conv 3 × 3, F = 32, D = 1
	Conv 3 × 3, F = 32, D = 2
	Conv 3 × 3, F = 64, D = 2
	Conv 3 × 3, F = 64, D = 3
LFE	Conv 3 × 3, F = 64, D = 2
	Conv 3 × 3, F = 64, D = 2
	Conv 3 × 3, F = 64, D = 1
	Conv 3 × 3, F = 64, D = 1
	Conv 1 × 1, F = 1, D = 1

Table 6.1: Detailed architecture of the proposed network inspired by [97], where F is the number of filters and D is the dilation factor to perform dilated convolutions. It is possible to notice the symmetric structure of the dilations whose factor increases in the Front End (FE) module, allowing us to increase the receptive field, and decreases in the Local Feature Extraction (LFE) module, aggregating local features to obtain spatial consistency in the output map. Note that each convolutional layer is followed by batch normalization (except the last layer) and ReLU activation function.

network's layers in a straightforward way is detrimental in aggregating local features. Dilation causes weights to skip information between cells, and this results in a bad modelisation of the structure of small objects, presenting grid patterns in the final output.

To solve this problem, they propose a network without pooling layers that conversely concatenates a Front End (FE) module of increasing dilation factors, with a Local Feature Extraction (LFE) module of decreasing dilation factors, arranged in a symmetrical way. The FE module is thus able to consider larger context for small objects detection, while the LFE module enforces the spatial consistency of the output by gathering spatial information decreasing the dilation size.

The Front End architecture employed in [97] is a VGG network [244] deprived of the final classifier layer (to obtain a fully convolutional network) and deprived of the max pooling layers. Instead of the latter, dilation factors are increased at the corresponding network depth. Then, the LFE module keeps invariant the number of filters and the kernel size, while decreasing the dilation factors up to one, in a specular way with respect to the FE.

The only drawback of such an architecture is that with respect to the U-Net it needs more memory to perform backward and forward passes, because feature maps at each layer have always the same size as the original input since there is no pooling operation. Thus, we propose a modified architecture (shown in Table 6.1) that keeps the memory use manageable by reducing the number of filters at each convolutional layer.

The choice of the reduction of the number of filters per layer has nevertheless two purposes. On the one hand, it allows us to fulfill hardware constraints (NVIDIA Geforce GTX1080 graphic card with 8GB of video memory), while on the other hand it helps in preventing the network to overfit, as we know we are going to train it with small datasets. To this extent, also batch normalization has been added on top of each convolutional layer, for faster convergence.

Note that to obtain a "simpler" network to prevent overfitting we could have conversely decreased the number of total layers. However, we preferred to reduce the number of filters per layer for two reasons. Firstly, decreasing the number of layers would have prevented the network to learn more complex features. Secondly, we would have not entirely exploited the benefit of increasing/decreasing dilation factors. Moreover, we found that adding many layers with a dilation

factor of 3 was too heavy, and for this reason we preferred to add a single central layer with such a big dilation (however, this depends on the expected size of the targets).

6.3.5 A remark on the last layer

Whatever the fully convolutional network used, another modification to the usual structure of fully convolutional networks regards the last layer. Usually, the layers' chain terminates with a convolutional layer, without any activation function that would apply a non-linear transformation which is usually not needed. In our specific application however, the desired output values are real values which are possibly positively unbounded (a pixel's score is indeed the sum of contributions given by all the surrounding Gaussian distributions representing surrounding heads), while at the same time loosing their meaning as long as they take values under zero.

By definition, being a cumulative sum of Gaussian distributions, the ground-truth maps obtained through the soft labeling procedure contain values which are always greater than or equal to zero. In this way, by simply integrating the map over a region of interest we can obtain the number of people present in that area. Conversely, if the integration over a given region of the ground-truth map is exactly zero, it means that there are no people in that particular area. It is therefore impossible to have in the ground-truth maps negative pixel values, as it is not possible to have a negative number of people present in a given area. By setting the last layer as a convolutional one however, we would allow the network to possibly produce negative output values which would be the origin of noise in the estimation of the number of people.

For this reason, we propose to use an activation function in the last layer which bounds the values at zero, like the sigmoid or the ReLU. For example a sigmoid activation as last layer is used in [120], to constrain the output values between 0 and 1. For our particular application however, the sigmoid is not suited for two main different reasons. Firstly, since the sigmoid function tends toward zero without really reaching it, punctual noise would become more evident. Secondly, since it saturates at 1, we loose the meaning of "cumulative" output: when two heads are one next to the other, they would result in a single large blob and the score of each pixel would no more represent the cumulative sum of surrounding heads contributions.

On the contrary, we propose to use a ReLU activation function in the last layer. It has the effect of a threshold, setting all the negative values to zero. Nevertheless, since it is integrated inside the network, it has beneficial effects on backpropagation with respect to a simple post-processing thresholding. In this way, the network learns easily to return zero for background pixels, being able at the same time to suppress a part of the background noise. The local density estimation is therefore also enhanced, since the network looses its tendency to compensate between low and high values adding noise.

6.4 CNN Results

6.4.1 Network training

To illustrate the benefit of the proposed deep learning approach inspired by semantic segmentation for pedestrian detection with soft labels, the two considered networks are trained on two different datasets: the Makkah dataset exploited so far and a dataset containing images of pedestrians captured in Regent's Park (from now on, the dataset will be referred to as *Regent's Park*).

In order to have more freedom on the batch size choice, while allowing at the same time for a more diverse data augmentation within each batch, Makkah images have been split in three, obtaining 35 training images of size 475×534. Although at first sight this can be seen as a small training set, consider that in each image there are approximately 300 heads, very densely spatially arranged, so that each image is able to convey a lot of information. Moreover, in fully convolutional networks each pixel contributes to the final output and can be seen as a different "training point", and in our case we have more than 250K pixels per image. Considering that a head diameter spans

between 8 and 12 pixels, pixel-level class imbalance issue is solved by setting $\eta = 150$ in Eq. (6.1) (value empirically obtained through validation).

Regent's Park dataset, on the contrary, contains 140 training images with about 40 heads per image. The head size is approximately the same as in Makkah dataset, but the number of people per image is far less. Nevertheless, it is worth to consider it for two reasons, namely to test the proposed approach also in presence of lower crowd densities to show the robustness of the method, and to validate the data imbalance solution in presence of an extreme class misproportion ($\eta = 1500$).

The two networks are trained by using an Adam stochastic optimizer [138], with a learning rate of 10^{-2} for U-Net and 7×10^{-3} for the proposed FE+LFE (exact values have been found through validation). In both cases, the weights of the convolutional layers are initialized with the Kaiming He method [101], which has proven to be particularly adapted for deep networks relying on ReLU activation functions.

Early stopping with a patience of 20 epochs is used in order to terminate the learning process when the networks stop improving on the validation set. This contributes also to mitigating the risk of overfitting. A question arises now, namely how to evaluate the performance of the networks (either on the validation set to perform early stopping and obtain the best model, or more generally to perform inference on unseen testing images).

6.4.2 Evaluation method

Due to the soft label definition, the proposed approach is able to provide as output a score map in which each pixel is associated to a real value representing an accumulation of head distributions. In this way, we are able to perform two tasks at a time, namely density estimation by integration, and pedestrian detection. Let us now concentrate on this latter.

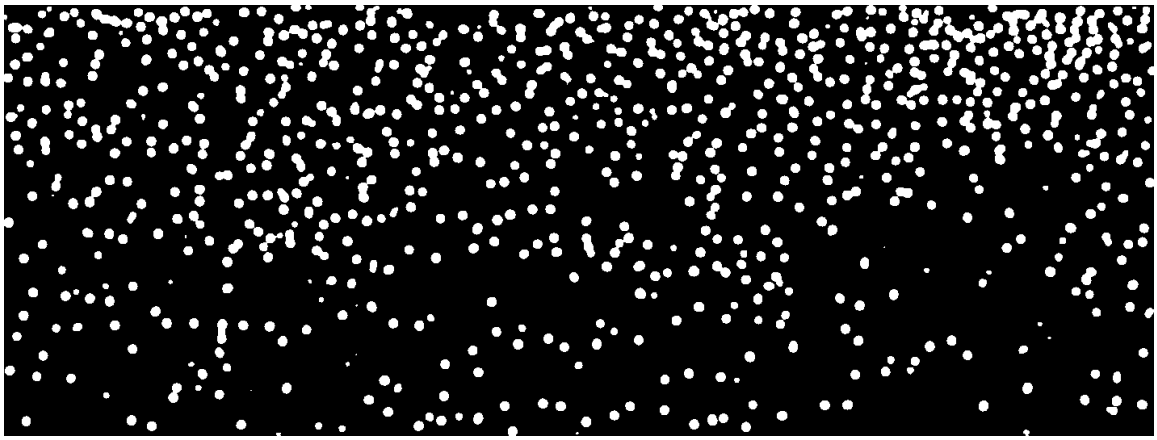
In order to perform pedestrian detection from the heads occupation map, we have to turn again the problem from a semantic segmentation-like task into an *instance segmentation* one. Instance segmentation is considered nowadays one of the most challenging issues, overcoming semantic segmentation because, besides estimating a different label per pixel, it is devoted at grouping pixels belonging to different objects of the same type in different instances. Particularly, in the case of our application, it is aimed at obtaining different blobs of pixels associated to every head and possibly splitting a single blob containing two different instances in two different close heads which partially overlap.

This is not however straightforward because the number of instances is initially unknown. Some methods like Mask R-CNN [100] jointly estimate the semantic segmentation mask along with bounding boxes of each target to perform instance segmentation, at the expense of a more complicated network structure. Alternatively, there exist extensions of U-Net to perform a joint estimation of the segmentation mask and of the boundaries of each objects (by treating them as a different class), but this can work only in presence of well-shaped borders, thus not in our case.

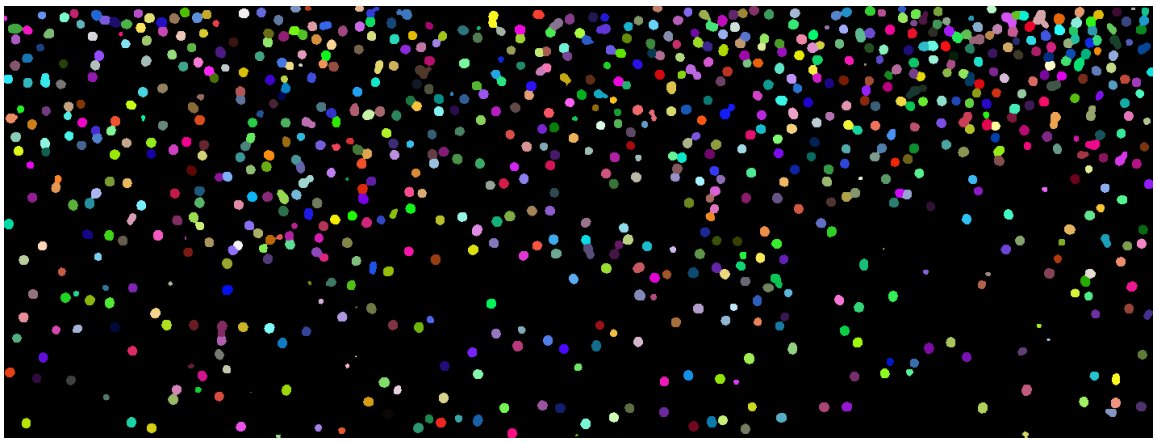
From our side, the peculiarity of the proposed soft labels definition directly gives a solution to the problem, as anticipated in Sec. 6.3.1. Since our output map is a cumulative map of Gaussian distributions, we can simply divide the various heads by getting the location of the maxima of such distributions and applying an inverted watershed algorithm [14] that performs a flooding process starting from these maxima.

However, note that the network estimates output values which are defined on all the positive domain, so that the spatial extent of the detections would be indefinitely large considering pixels of extremely small values. In order to constrain the size of the detections to a reasonable area, once again we refer to the derived ground-truth map with soft labels, where we expect the head radius to be equal to $2\sigma_h$. Thus, we set to zero all the values of the output map which are less than τ, that corresponds to the Gaussian function evaluated at distance $2\sigma_h$ from the labeled center, without forgetting the scaling factor η:

$$\tau = \eta \mathcal{N}_{((x_c,y_c),\sigma_h)}(2\sigma_h). \tag{6.2}$$



(a)



(b)

Figure 6.5: (a) Pedestrian semantic segmentation on a test image from the Makkah dataset, with respect to τ threshold defined in (6.2). Background pixels are black, while white pixels are white. (b) Pedestrian instance segmentation derived from the semantic one, by applying the watershed algorithm over the peaks of the estimated head distributions. The different colors represent independent, possibly partially occluded, heads.

Figure 6.5a shows the semantic segmentation result after τ thresholding, while Fig. 6.5b shows the instance segmentation result with watershed algorithm.

Having turned the problem into an instance segmentation one, the evaluation has to be made at component level, and not pixel-wise such as in semantic segmentation. To do so, we associate to every detection a different global score, which expresses the likelihood of being a real head, given by the maximum of that detection blob (we could think about weighting it with respect to the detection size, but the results would not dramatically change), and we rank the detections in descending order of global scores.

In this way, we can use the *mean Average Precision* (mAP), which is a standard metric for measuring the performance of object detectors. It can nevertheless be seen as an approximation of the area under precision-recall curve, and it is faster to compute than considering each (discretized) threshold, resulting to be particularly adapted to be evaluated at every epoch on the validation images, in order to possibly early stop the training to avoid overfitting.

After having ranked the detections by descending global scores, precision and recall are computed at every rank r, i.e. incorporating the first r detections in the computation. Precision at rank r is defined as the proportion of all the detections ranked r or more which are really TP, whereas recall at rank r is defined as the ratio between the number of TP ranked r or more and the total number of ground-truth detections.

Then, mAP is computed as the average on every image (of the validation or testing set) of the integral of the curve obtained by taking, for each different value of recall, the maximum precision

for that value of recall or higher.

A question arises, namely how to discriminate between TP and FP detections. To this extent, the standard approach is to calculate the *Intersection over Union* (IoU) between ground-truth and detected blobs. It is defined as the ratio between the number of pixels which belong to both the ground-truth and the detection (intersection), and the number of pixels which belong to the ground-truth or the detection (union). Then, a detection is considered TP if its associated IoU is over a predefined value $\mathscr{I}$ (in case a detection does intersect with more than one ground-truth blob, only the one with the highest IoU is labeled as TP, while the others are labeled as FP). We will thus refer to $\text{mAP}_{IoU=\mathscr{I}}$ as the mAP calculated with a IoU threshold of $\mathscr{I}$.

Note that we employ mAP on the validation set to perform early stopping because its maximization is more relevant than just considering the model that gives the lower loss on the validation images, as the loss is computed pixel-wise while the mAP takes into account the notion of "detection" yet being easy to compute. Regarding the final evaluation on the testing set, we will still compute mAP, and we will also compute standard precision-recall curves.

6.4.3 Results

Makkah	$\text{mAP}_{IoU=0.3}$	$\text{mAP}_{IoU=0.5}$
U-Net	0.86	0.74
FE+LFE	0.88	0.74

Table 6.2: Quantitative results of the two considered networks on the Makkah dataset. Results are shown in terms of mAP with 0.3 and 0.5 IoU thresholds.

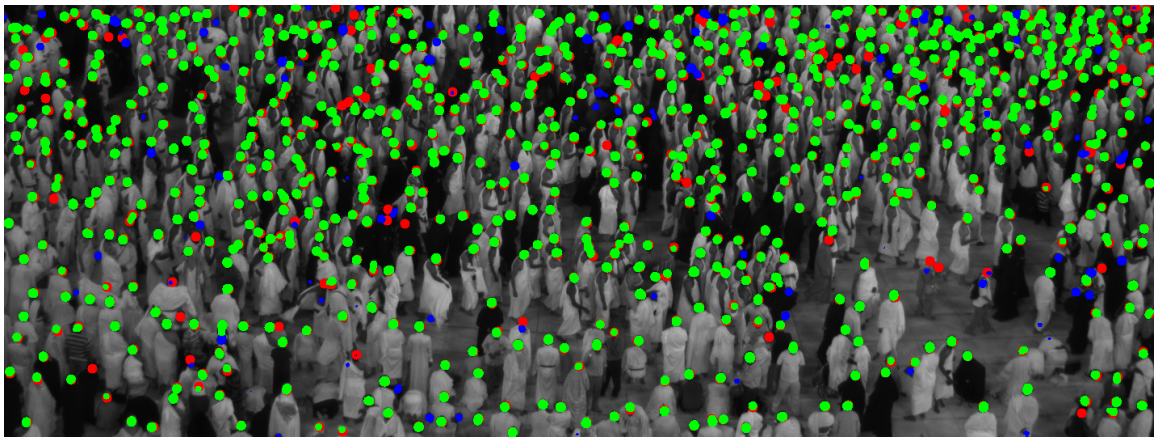
Regent's Park	$\text{mAP}_{IoU=0.3}$	$\text{mAP}_{IoU=0.5}$
U-Net	0.88	0.76
FE+LFE	0.88	0.78

Table 6.3: Quantitative results of the two considered networks on the Regent's Park dataset. Results are shown in terms of mAP with 0.3 and 0.5 IoU thresholds.

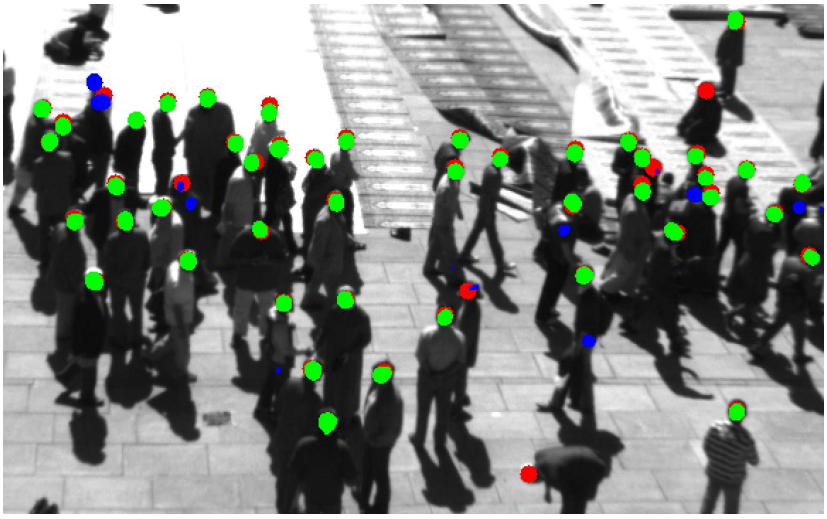
We are firstly interested in a comparison between the state-of-the-art U-Net and the proposed FE+LFE network. To do so, we compute the mAP scores with both $\mathscr{I} = 0.3$ and $\mathscr{I} = 0.5$. Standard deviation σ_h for the creation of soft labels is set to 3. Tables 6.2 and 6.3 show the mAP scores for both architectures for Makkah and Regent's Park datasets respectively. For standard applications it is common to set $\mathscr{I} = 0.5$; however, considering our specific problem, we realize that this could be too strict. In presence of such a difficult application, imprecision in the ground-truth labeling of the head's centers can have a noticeable impact in the resulting IoU. This fact is accentuated by the presence of very small targets. For this reason, we relax the IoU threshold to 0.3, in order to tolerate a small misplacement on the location of the head. We nevertheless provide results for both thresholds.

The two different architectures provide comparable performance in terms of mAP on both datasets. The proposed FE+LFE results to be slightly better than the U-Net. However, consider that the number of parameters of the FE+LFE is considerably less with respect to U-Net, proving the importance of having defined an architecture that, although with less number of filters, is especially crafted for small object detection. Moreover, note that the overall performance of the

method remains similar in presence of two different datasets, showing its robustness with varying number of training images and with varying crowd density. This is indeed a very desirable property.



(a)

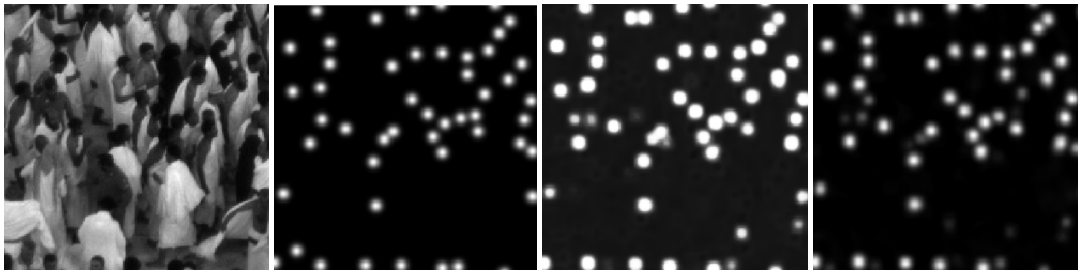


(b)

Figure 6.6: Pedestrian detection results on (a) *Makkah* and on (b) *Regent's Park Dense* datasets, with the proposed FE+LFE network. Red blobs are ground-truth heads, green blobs are TP detections (i.e. detections which are successfully associated to a ground-truth blob, with a IoU threshold equal to 0.3), and blue blobs represent FP detections.

Figure 6.6 shows the results on both datasets of the pedestrian detections obtained considering an IoU threshold $\mathscr{I} = 0.3$ and a standard deviation in the soft label definition $\sigma_h = 3$, with the proposed FE+LFE network. Ground-truth detections are depicted in red, and detections are overlaid on top of them. Green blobs represent thus TP detections, namely detections which are successfully associated to a ground-truth blob, while blue blobs represent FP detections. Again, note the ability of the network to adapt itself to different levels of density.

Now, let us concentrate on the challenging Makkah dataset, and investigate a particular design choice that we made, i.e. the use of a ReLU activation function in the last layer to avoid negative values in the output map. Fig. 6.7 shows the effect of both sigmoid and ReLU functions which indeed eliminate negative values in two different ways, i.e. by non-linearly scaling the values between 0 and 1, and by thresholding at 0 respectively. In particular, the choice of the ReLU appears to be the most appropriate. The sigmoid function adds noise in the background, since it tends toward zero without ever reaching it (this is visible in the background of Fig. 6.7c which is dark-grey rather than black), and it saturates at high values making worthless the notion of "cumulative" head distributions. Conversely, ReLU is good at suppressing background noise while at the same

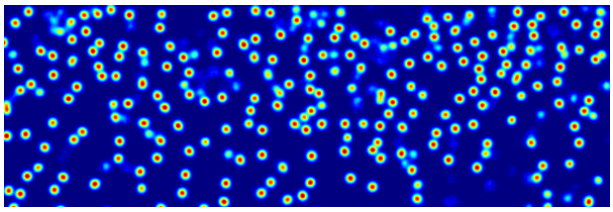


(a) **(b)** **(c)** **(d)**

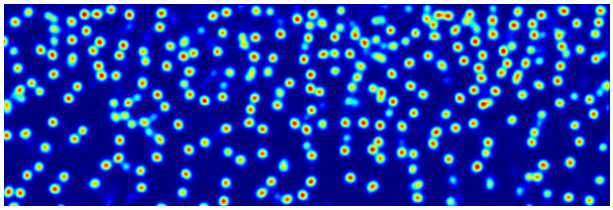
Figure 6.7: Effect on the output map of different activation functions in the last layer. (a) test image patch; (b) corresponding soft-labeled ground-truth; (c) output map with sigmoid activation function in the last layer; (d) output map with ReLU activation function in the last layer.

time not scaling the values in a non-linear way.

(a) Image patch



(b) Output map of FE+LFE



(c) Output map of U-Net

Figure 6.8: Output maps on a testing patch image with the proposed FE+LFE network and with the U-Net network.

Still considering the Makkah dataset, Fig. 6.8 shows a qualitative comparison of the output maps obtained with the FE+LFE and U-Net networks. Visually, both solutions provide well-shaped and localized detections. FE+LFE in particular gives a less noisy output map compared to U-Net.

In order to perform a quantitative evaluation of the FE+LFE and U-Net networks, Fig. 6.9 shows the PR-curves after NMS using the two different architectures, while Table 6.4 shows the corresponding PRBEP and AUPRC values on the Makkah dataset. The proposed FE+LFE network clearly outperforms U-Net, reaching notably higher values of precision, highlighting once again the importance of having defined an architecture capable of precisely detecting small objects.

Note that it is not completely fair to compare PR-curves obtained using deep learning methods with the ones obtained using the supervised learning techniques seen in previous Chapters (e.g. the one in Fig. 5.9 obtained applying the proposed Evidential QBC Active Learning). In fact, the two methods have different purposes. The proposed SVM-based active learning is particu-

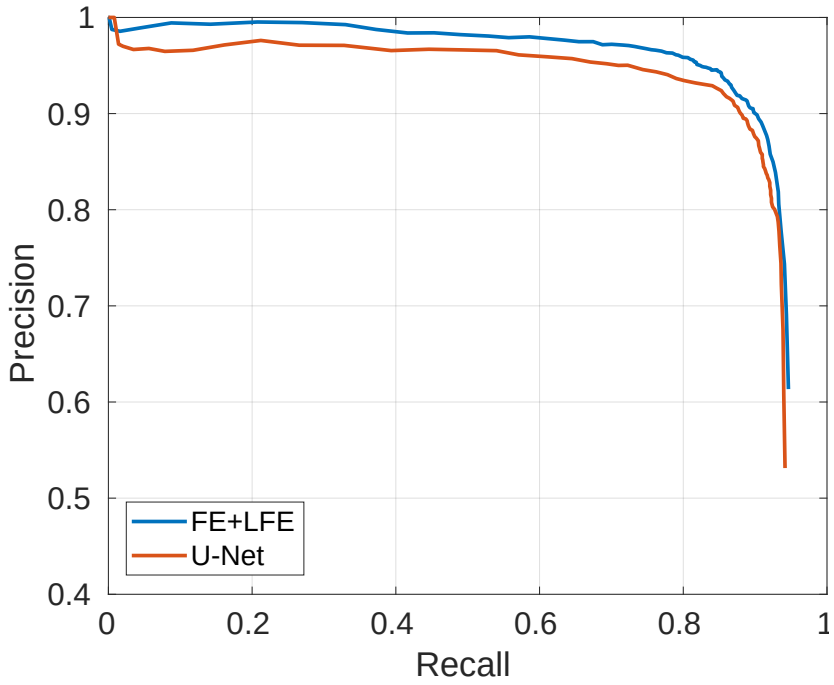


Figure 6.9: PR-curves using FE+LFE and U-Net deep architectures.

Table 6.4: Precision-Recall Break Even Point and Area Under Precision-Recall Curve with the different deep architectures.

	FE+LFE	U-Net
PRBEP	**0.90**	0.89
AUPRC	**0.92**	0.89

larly adapted in situations where we have a specific problem (pedestrian detection in *very dense* crowds) and an extremely small training set (e.g. 2000 samples). Conversely, the proposed deep learning solution is adapted to perform generic pedestrian detection in crowds of varying density (possibly cross-scene, if trained accordingly), with small yet consistent training datasets (even though we rely on a small number of images, consider that a single image contains more than 250K pixels that can be seen as different training samples for the particular fully convolutional architecture type).

Rather than naively compare the two methods, having seen that they are intended to solve different problems, we can ask ourselves how the deep learning solution would work in presence of even less training data. Even further, we can think about a fusion of the results obtained with active learning and the deep learning results trained with the same amount of limited data. These subjects will be better explored in the next Chapter.

Chapter 7

CNN-ensemble and evidential Multiple Classifier System

Contents

7.1 Motivation

So far, two different methods to perform pedestrian detection have been proposed. The first one (Chapter 3, Chapter 4, Chapter 5) is based on SVM and it is particularly adapted in situations where we want to solve a specific problem (i.e. high-density crowd pedestrian detection) using very few labeled data (around 2000 training samples). Working in the BF framework with multiple SVM descriptors, we have designed a BBA allocation that takes into account possible errors in the score calibration and in the pixel-wise detection in the image space, and an active learning procedure being able to directly exploit evidential functions in order to select the most informative training samples.

Conversely, in Chapter 6 we inspected deep learning solutions to perform pedestrian detection in a more general way. We proposed a fully convolutional network especially designed to recover small objects by the use of dilated convolutions, showing the robustness of the method to variations in the crowd density (i.e. from sparse to very dense crowds). In this context however, the training of the considered networks has been done relying on a larger labeled dataset.

Even though deep learning solutions tend to outperform the other supervised learning techniques when trained on large amounts of data, applying them effectively in presence of few labeled data is nowadays an open issue. Most of the existing works are devoted to finding the best network for cross-scene pedestrian detection or counting trained with huge datasets, but few attention is

given to specific real-setting problems where training data are hard to obtain and therefore out-of-the-box networks cannot be used, as they consist in models with billions of parameters too complex to be learned with respect to the available data.

Nonetheless, in recent years many regularization techniques have been proposed to tackle the problem of overfitting, from data augmentation to early stopping and dropout, besides the traditional weight decay. These techniques used together could help in applying deep learning techniques also in presence of small datasets. Indirectly, also batch normalization has shown to be important contributing to a faster convergence.

Simultaneously, a criticism that is often made of deep learning methods is the fact that they act like "black-boxes", making it hard for their users to interpret the obtained results. This limitation is highly relevant when learning from small amounts of data, where a measure of model uncertainty would be particularly important. To this extent Bayesian Neural Networks (BNNs, Bayesian NNs) offer a probabilistic interpretation of deep learning models by inferring distributions over the models' weights, allowing to measure model uncertainty, but they are usually practically limited. Recently, an ensemble-based method relying on the use of dropout at inference time has been proposed in [86], allowing to obtain several realization sampled from the same network with randomly dropped-out units at test time, from which average and standard deviation are computed to obtain a more robust output along with a confidence measure on the prediction.

Following this line of work, we intend to investigate the use of deep learning techniques in presence of small training datasets for specific applications (i.e. in our case high-density crowd pedestrian detection). Again, the use of ensemble methods is important for two different reasons. Firstly, it acts as another regularization technique to mitigate the risk of overfitting (cf. *statistical reasons* in Sec. 2.3.1); secondly, it allows us to measure the model confidence about each prediction.

Even further, we intend to perform fusion of the results previously obtained through SVM-based evidential QBC active learning (from now on referred to as SVM-ensemble) and the deep learning ensemble solution here proposed (CNN-ensemble) trained on the same very small amount of data. The final result will therefore be a Multiple Classifier System composed by two different ensembles, one based on SVM and the other one based on CNN, proving that deep learning techniques can be applied also in presence of extremely small datasets for solving targeted problems, and can benefit from the fusion with another strong classifier.

7.2 Representing model uncertainty in deep learning

Obtaining a measure of uncertainty of a model trained with deep learning techniques is not trivial. The general training of deep learning models allows us to obtain the best model parameters through backpropagation, but they are usually only point estimates. These parameters are then kept fixed at inference time in the forward pass to perform prediction. However we cannot easily know whether a trained model is certain about its output. Binary classifiers such as SVM on the contrary provide a score (in the case of SVM the sample distance from the hyperplane margin) that can be interpreted to understand if the model is making sensible predictions or just almost randomly guessing. However, none of these classifiers provide credible or confidence intervals about their predictions.

Let us consider a toy example of a network trained to recognize multiple classes of dog breeds. If an image of a cat is given as input at inference time, it would anyway return a probability for this cat to belong to each class of dog breeds. These probabilities (generally obtained through Softmax) are often erroneously interpreted as model confidence scores, but they are not. A model in fact can be uncertain about its predictions even in presence of a possibly high Softmax output. In the considered example, the model would be uncertain about how to classify the cat, since it has not seen any cat during its training being able to discriminate only between dog breeds, but the cat image would anyhow be assigned to dog breeds with different probabilities. Another example is given by adversarial inputs, which can be incorrectly labeled even with very high probabilities.

We want the network to be able to measure *predictive uncertainty*, that is the confidence it has with respect to the prediction it makes (ideally, in the example above, we would like the network to convey high uncertainty when fed with the cat input).

This is particularly important for applications related to real-settings such as autonomous driving or security access to critical systems, where relying on model uncertainty to adapt decision making is crucial, and generally for applications for which only a small amount of data is available for the training. In such situations it becomes important to know if a network is uncertain about a prediction because it has never seen a similar example during the training, or because of its limited representational capacity.

7.2.1 Bayesian Neural Networks

BNNs have been firstly studied extensively in [177, 191] and more recently in [19, 93, 139], sometimes being referred to as *variational techniques*. They are based on the observation that an infinitely wide neural network with distributions placed over its weights converges to a Gaussian process [191], thus, considering finite NNs, they are mathematically equivalent to an approximation of the probabilistic deep Gaussian process [52].

In order to obtain model uncertainty estimates, BNNs place a prior probability distribution over each networks' weight. In this way, they potentially offer robustness to overfitting during training along with uncertainty estimates about the predictions.

However, the applicability of these types of models is quite limited, and they have not been largely followed up by the deep learning community. If on the one hand Bayesian probability theory offers mathematically grounded tools to reason about model uncertainty, on the other hand they come usually with prohibitive computational costs. BNNs have shown indeed to be quite difficult to work with, often requiring the optimization of many more parameters with respect to standard networks.

In order to obtain the posterior distribution over the weights, variational inference has been applied to approximate it [93]. An alternative inference approximation for Bayesian NNs based on Monte Carlo techniques has been proposed in [191] that do not rely on any prior assumptions about the form of the posterior distribution. Both methods however had limited practical success. Sampling-based variational inference and stochastic variational inference have been then introduced thanks to recent advances in the field, but also in this case the models come with a prohibitive computational cost [139].

A mixture of Gaussian priors over each weight has been employed in [19], allowing the authors to improve the model performance compared to [93]. However also this method remains computationally too expensive, since it increases the number of model parameters without considerably increasing the model capacity. This makes the approach difficult to use with large complex models as the increase in number of parameters could prevent its applicability with limited hardware capacity.

To conclude, generally existing approaches to obtain model confidence do not scale to complex models and present limited applicability requiring the development of new models from scratch (common deep learning frameworks do not support these methodologies).

7.2.2 MC-dropout

Recently, the authors of [86] developed a new theoretical framework by casting *dropout* (and its variants) in deep NNs as approximate Bayesian inference in deep Gaussian processes. The foundation of this theory directly provides tools to model the uncertainty without the need to change neither the model architecture nor the objective function.

The authors have shown that a neural network with arbitrary depth and non-linearities, with dropout applied at every layer, is mathematically equivalent to an approximation of the probabilistic deep Gaussian process. This means that the optimal weights found through the optimization of a NN with dropout are the same as the optimal variational parameters in a Bayesian NN with

the same structure. Further, this means that a network already trained with dropout *is* indeed a BNN.

Moreover, this result is valid not only using dropout, but also its variants such as DropConnect [266] or multiplicative Gaussian noise [251], i.e. using any Stochastic Regularization Technique (SRT). SRTs are techniques used to regularize a deep learning model through the injection of stochastic noise directly into it (the most popular technique is dropout which switches off units with respect to a previously set probability). The intuition is that SRTs approximately integrate over the models' weights, so that they can be interpreted as performing approximate inference and, as a result, uncertainty information can be extracted.

Practically, after training the network, Monte Carlo (MC) methods are used at test time to draw samples from a Bernoulli distribution across the network's weights, by performing T stochastic forward passes through the network with dropout. This is why the method is known as *MC-dropout*. Note that this does not require any additional parametrization, and from this it is easy to derive the sample mean by averaging the results and the standard deviation that can be interpreted as predictive uncertainty.

In this way once again we obtained an ensemble method, composed now by T different realizations given by dropping out different units of the network at each forward pass. This method has several advantages, i.e. it is easily adaptable to complex models and does not require any change to the model architecture or optimization procedure, besides being very easy to implement in practice.

We shall finally mention a different but related approach, that cannot be considered as approximate inference in BNNs but nevertheless can be used to estimate model uncertainty relying on ensemble learning. This technique builds an ensemble of deterministic models (each model in the ensemble produces a point estimate rather than a distribution) by independently training the same network on the same dataset many times with different weights initialization. Then, at inference time, an average is made to obtain a prediction and the prediction uncertainty is measured through the variance of the outputs of all the models.

Very recently, [152] proposed *deep ensembles* based on this idea, relying also on adversarial training [92, 254] to smooth predictive distributions, treating the ensemble built in this way as a uniformly-weighted mixture model and approximating the ensemble prediction as a Gaussian whose mean and variance are the ones of the mixture respectively.

However, this approach is not always suitable for a number of reasons. Firstly, training many neural networks can be a long process. Secondly, even if this approach is anyhow computationally more efficient than many Bayesian approaches presented in the previous section, its produced uncertainty estimates lack in many ways [84]. Lastly, the use of mechanisms such as batch normalization allows for faster convergence to a robust solution even with different weights initialization, making this process unprofitable in many cases.

7.3 CNN-ensemble

MC-dropout has been successfully applied in different applications, from segmentation for scene understanding [134] to camera re-localization [135], allowing to model the predictive uncertainty through standard deviation of the stochastic realizations obtained with dropout. Now, we want to formulate the Bayesian counterpart of the proposed FE+LFE network that uses MC-dropout to get samples from the posterior distribution over the network's weights.

7.3.1 Bayesian FE+LFE

For the definition of the Bayesian FE+LFE we follow the formulation of Bayesian SegNet [134], which is built upon SegNet [10] network for pixel-wise segmentation.

Given a list of training inputs $\mathbf{x}$ and corresponding outputs y, we are interested in finding the posterior distribution over the network's convolutional weights, $\mathbf{W}$:

$$p(\mathbf{W}|\mathbf{x}, y), \tag{7.1}$$

which captures the most likely weights to have generated our outputs given the observed data. However, this posterior distribution is generally intractable, and needs to be approximated. To this extent, variational inference can be used, that allows us to approximate the intractable posterior through a function $q(\mathbf{W})$ over the network's weights. This function is learned by minimizing the KL divergence between this approximating distribution and the actual posterior:

$$\mathscr{D}_{\mathrm{KL}}(q(\mathbf{W})||p(\mathbf{W}|\mathbf{x}, y)). \tag{7.2}$$

As proposed in [85], given a CNN with L layers of dimension K×K, we define the approximating variational distribution $q(\mathbf{W}_i)$ for every convolutional layer i with units j as:

$$\begin{aligned} \mathbf{W}_i &= \mathbf{M}_i \cdot \mathrm{diag}([z_{i,j}]_{j=1}^{\mathrm{K}_i}), \\ z_{i,j} &\sim \mathrm{Bernoulli}(p_i^{\mathrm{drop}}) \qquad i = 1,\ldots,\mathrm{L}, \quad j = 1,\ldots,\mathrm{K}_{i-1}, \end{aligned} \tag{7.3}$$

where $z_{i,j}$ are Bernoulli distributed random variables with probabilities p_i^{drop} (i.e. dropout probabilities), and $\mathbf{M}_i$ contains the variational parameters to optimize. Note that dropout probabilities p_i^{drop} could also be optimized, but as in [134] we kept them fixed to an equal constant value found through validation. In this way, as proven in [85], we obtain the approximate model of the Gaussian process.

Now, we train the network and we sample the posterior distribution over the weights using dropout at test time, performing T different forward passes through the network. As a result for a given testing image we obtain T different realization maps $\hat{\mathcal{M}}_1,\ldots,\hat{\mathcal{M}}_{\mathrm{T}}$, output of different dropout-perturbed versions of the original network. Classically, the mean map $\mathcal{M}_\mu$, given by the mean value evaluated independently for each pixel, is interpreted as the final prediction map, while the standard deviation map $\mathcal{M}_\sigma$ is interpreted as an estimate of the predictive uncertainty. However, we propose once again to work in the BF framework, that we consider more suited to model the specific imprecision of each different realization obtained with dropout. In the following, we will explain the proposed BBA allocation for every realization that will allow us to perform a robust fusion among them as well as to obtain evidential measures of predictive uncertainty for every pixel of the final output map.

7.3.2 BBA allocation for CNN-ensemble

While being an easy yet mathematically grounded approach to obtain a measure of uncertainty out of any kind of deep network, the method presented above has some drawbacks. Firstly, for practical reasons, often we can perform only a limited number of forward passes, and the mean value could not be so representative of the actual distribution especially in presence of outliers. The second problem, as reported in [84] (in particular in Sec. 3.3.2) is more theoretical, and comes from the fact that the obtained uncertainty is not calibrated (it can scale differently for different datasets) and usually underestimated (variational inference is known indeed to underestimate predictive variance).

Leaving partially apart the mathematical ground in favor of an analysis more adapted to our specific setting, we shall note that median has been shown to be a more robust estimator than the average in presence of outliers. In the same way, instead of relying on the standard deviation, we can better employ the Median Absolute Deviation (MAD), which is a robust measure of the variability of a univariate sample of quantitative data. The MAD is a robust statistic, being more resilient to outliers than the standard deviation. In this latter indeed the distances from the mean value are squared, so large deviations are more weighted and outliers can heavily influence it. In the MAD conversely, large deviation of a small number of outliers is irrelevant.

In our context, given the T realization maps, the MAD map $\mathcal{M}_{\mathrm{MAD}}$ is defined as:

$$\mathcal{M}_{\text{MAD}} = \text{median}\left(\left| \hat{\mathcal{M}}_i - \text{median}\left(\left\{ \hat{\mathcal{M}} \right\}_1^{\text{T}} \right) \right| \right), \tag{7.4}$$

where $\text{median}\left(\left\{ \hat{\mathcal{M}} \right\}_1^{\text{T}} \right)$ is the median over all the T realizations.

However, even with the use of these statistics the problem of finding a good representation of the model predictive uncertainty is not completely solved. To this extent, we propose to rely once again on the Belief Function framework to obtain a better estimation of the model imprecision given the T realizations that can be interpreted as different sources for an evidential combination. In order to do so, we need to perform BBA allocation.

We have T maps $\hat{\mathcal{M}}_1, \ldots, \hat{\mathcal{M}}_{\text{T}}$ which corresponds to the T output realizations obtained with forward passes through the network with dropout, and we are interested in finding a BBA allocation to perform the combination among them and derive evidential measures of imprecision.

Firstly, we can derive Bayesian BBA maps $\mathcal{M}_1^{\mathcal{B}}, \ldots, \mathcal{M}_{\text{T}}^{\mathcal{B}}$, four-layer images where a BBA is associated to each pixel $\mathbf{x}$ of each realization, so that we obtain T maps of BBAs $\left\{ m_{\mathbf{x},t}^{\mathcal{B}}, \mathbf{x} \in \mathscr{P} \right\}$, where $\mathscr{P}$ is the pixel domain and $t = 1, \ldots, \text{T}$. These Bayesian BBAs maps are 4-layers images where each layer corresponds to the mass values of any hypothesis in $\left\{ \emptyset, \text{H}, \overline{\text{H}}, \Theta \right\}$. $\mathcal{M}_t^{\mathcal{B}}(\text{A})$ corresponds to the layer image associated to hypothesis A for the realization (source) t. Note that in this preliminary Bayesian BBA allocation, layer images corresponding to non-singleton hypotheses are null, by definition. So, for each source t, with $t = 1, \ldots, \text{T}$:

$$\begin{cases} \mathcal{M}_t^{\mathcal{B}}(\emptyset) &= \{0\}_{\mathbf{x} \in \mathscr{P}}, \\ \mathcal{M}_t^{\mathcal{B}}(\text{H}) &= \hat{\mathcal{M}}_t, \\ \mathcal{M}_t^{\mathcal{B}}(\overline{\text{H}}) &= 1 - \hat{\mathcal{M}}_t, \\ \mathcal{M}_t^{\mathcal{B}}(\Theta) &= \{0\}_{\mathbf{x} \in \mathscr{P}}. \end{cases} \tag{7.5}$$

Now, we want to take into account the reliability of the pixel-wise prediction given by every source in order to perform a pixel-wise tailored discounting. Note that this would be impossible in the probabilistic framework; moreover, we are not just computing an overall source discounting, but rather each pixel of each source will be discounted differently on the basis of its reliability.

To measure this latter, we take inspiration from the MAD. For each source t, we compute a discounting coefficient map $\Gamma_t : \left\{ \gamma_{\mathbf{x},t} \right\}_{\mathbf{x} \in \mathscr{P}}$ such that a different coefficient $\gamma_{\mathbf{x},t}$ is associated to every pixel of each source,

$$\Gamma_t = \alpha \left(1 - \left(\left| \hat{\mathcal{M}}_t - \text{median}\left(\left\{ \hat{\mathcal{M}} \right\}_1^{\text{T}} \right) \right| \right) \right). \tag{7.6}$$

In this way, we discount more pixels whose value is more distant to the median value among the T realizations, since they are supposed to be less representative (even possibly outliers). The α parameter is a scaling factor which allows us to control the amount of discounting.

Applying the proposed discounting, we derive the following BBAs map for every source t: $\forall \text{A} \in \left\{ \text{H}, \overline{\text{H}} \right\}$,

$$\begin{cases} \mathcal{M}_t(\emptyset) &= \{0\}_{\mathbf{x} \in \mathscr{P}}, \\ \mathcal{M}_t(\text{A}) &= \Gamma_t \star \mathcal{M}_t^{\mathcal{B}}(\text{A}), \\ \mathcal{M}_t(\Theta) &= \{1\}_{\mathbf{x} \in \mathscr{P}} - \mathcal{M}_t(\text{H}) - \mathcal{M}_t(\overline{\text{H}}), \end{cases} \tag{7.7}$$

where $\text{M}_1 \star \text{M}_2$ represents the Hadamard product between matrices M_1 and M_2.

To combine the T different maps to obtain a single output map $\mathcal{M}$ with BBAs associated to each pixel $\mathbf{x}$, i.e. $\{m_{\mathbf{x}}\}_{\mathbf{x} \in \mathscr{P}}$, we use the conjunctive combination rule (cf. Eq. (4.8)). Note that although the different maps are not independent being sampled from the same distribution, we

choose to employ the conjunctive rule rather than the cautious rule [58], which is usually more adapted in case of non-distinct items of evidence since no elementary item of evidence would be counted twice. Indeed, we want to exploit redundant information to enforce our belief in an evidence if by sampling on the posterior distribution we get similar pieces of evidence. In our case where $|\Theta| = 2$, the analytic result using the conjunctive combination rule may be easily derived: $\forall \mathrm{A} \in \left\{\mathrm{H}, \overline{\mathrm{H}}\right\}$,

$$\begin{cases} m_{\mathbf{x}}(\mathrm{A}) &= \sum\limits_{\substack{(\mathrm{B}_1,\ldots,\mathrm{B}_\mathrm{T}) \in \{\mathrm{A},\Theta\}^\mathrm{T}, \\ \exists t \in [1,\mathrm{T}] s.t. \mathrm{B}_t = \mathrm{A}}} \prod_{t=1}^{\mathrm{T}} m_{\mathbf{x},t}(\mathrm{B}_t), \\ m_{\mathbf{x}}(\Theta) &= \prod_{t=1}^{\mathrm{T}} m_{\mathbf{x},t}(\Theta), \\ m_{\mathbf{x}}(\emptyset) &= 1 - m_{\mathbf{x}}(\mathrm{H}) - m_{\mathbf{x}}\left(\overline{\mathrm{H}}\right) - m_{\mathbf{x}}(\Theta). \end{cases} \tag{7.8}$$

The result is thus a four-layer map $\mathscr{M}$ of BBAs $m_{\mathbf{x}}$, that can be used to derive evidential measures of uncertainty about the network prediction. To this extent, we can obtain the ignorance map as $\mathscr{M}(\Theta)$, that represents the remaining ignorance which has been decreased by the combination but not completely solved, indicating a lack of sufficient information during training to perform a reliable prediction. Likewise, $\mathscr{M}(\emptyset)$ is often interpreted as a conflict map [151], and presents higher values for pixels whose prediction completely disagrees through the various realizations.

Finally, in every pixel $\mathbf{x}$ the decision is taken from $m_{\mathbf{x}}$. As already seen, pignistic probability may be used to give a probabilistic interpretation to the BBAs. The BetP(H) map can be computed with Eq. (4.23). This allows us to assign a $\mathrm{BetP}_{\mathbf{x}}(\mathrm{H})$ value to the resulting BBA associated to each pixel $\mathbf{x}$ that will be differently normalized on the basis of its conflict value, $m_{\mathbf{x}}(\emptyset)$.

Table 7.1: Example of different values obtained sampling the posterior distribution with MC-dropout technique with T=4, for two different pixels $\mathbf{x}_1$ and $\mathbf{x}_2$, along with the corresponding discounting coefficient $\gamma_{\mathbf{x},t}$ obtained with Eq. (7.6) setting $\alpha = 0.5$. After having performed the conjunctive combination among the discounted BBAs, $\mathrm{BetP}_{\mathbf{x}}(\mathrm{H})$ and $m_{\mathbf{x}}(\Theta)$ results are shown for the two pixels $\mathbf{x}_1$ and $\mathbf{x}_2$.

	$t=1$	$t=2$	$t=3$	$t=4$	median	$\gamma_{\mathbf{x},1}$	$\gamma_{\mathbf{x},2}$	$\gamma_{\mathbf{x},3}$	$\gamma_{\mathbf{x},4}$	$\mathrm{BetP}_{\mathbf{x}}(\mathrm{H})$	$m_{\mathbf{x}}(\Theta)$
$\mathbf{x}_1$	0.8	0.8	0.82	0.82	0.81	0.99	0.99	0.99	0.99	0.87	0.06
$\mathbf{x}_2$	0.01	0.99	0.27	0.73	0.5	0.51	0.51	0.77	0.77	0.5	0.2

To illustrate the benefit of the explained BBA allocation for the CNN-ensemble, Table 7.1 proposes a toy example where MC-dropout is applied to sample the posterior distribution obtaining T=4 realizations, for two different pixels $\mathbf{x}_1$ and $\mathbf{x}_2$. Then, discounting coefficients $\gamma_{\mathbf{x},t}$ are derived using Eq. (7.6), setting $\alpha = 0.5$. After having performed the conjunctive combination among the discounted BBAs, $\mathrm{BetP}_{\mathbf{x}}(\mathrm{H})$ and $m_{\mathbf{x}}(\Theta)$ are shown for the two pixels $\mathbf{x}_1$ and $\mathbf{x}_2$. The posterior distribution sampled for pixel $\mathbf{x}_1$ presents similar values with respect to the one sampled for pixel $\mathbf{x}_2$, so that all the realizations are close to the median value and thus we obtain high discounting coefficients that reflect in reliable BBAs that do not need to be much discounted. Conversely, $\mathbf{x}_2$ presents a sampled distribution which is more spread out, so that more discounting (i.e. lower discounting coefficients) is applied. This fact reflects in higher value of ignorance for $\mathbf{x}_2$, that may be interpreted as higher predictive uncertainty.

7.3.3 Global overview of CNN-ensemble

In order to provide a global overview of the CNN-ensemble method, Fig, 7.1 and Fig. 7.2 shows two different flowcharts, namely for the proposed evidential approach and for the traditional one respectively. After having trained the CNN and obtained a model, this latter is used at inference time to set the optimal weights of the network to perform inference on unseen images. We then

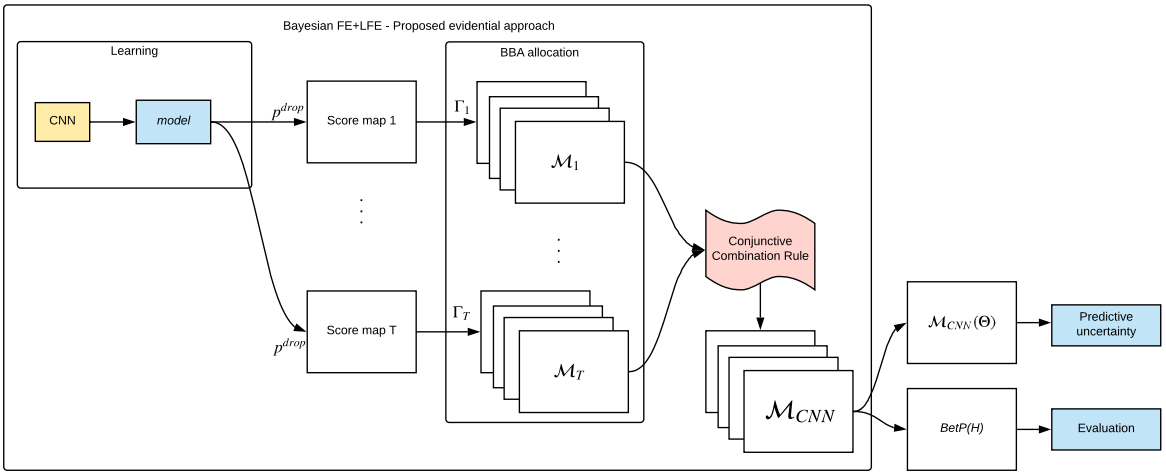


Figure 7.1: Proposed CNN-ensemble that performs evidential fusion of the T realizations obtained through MC-dropout.

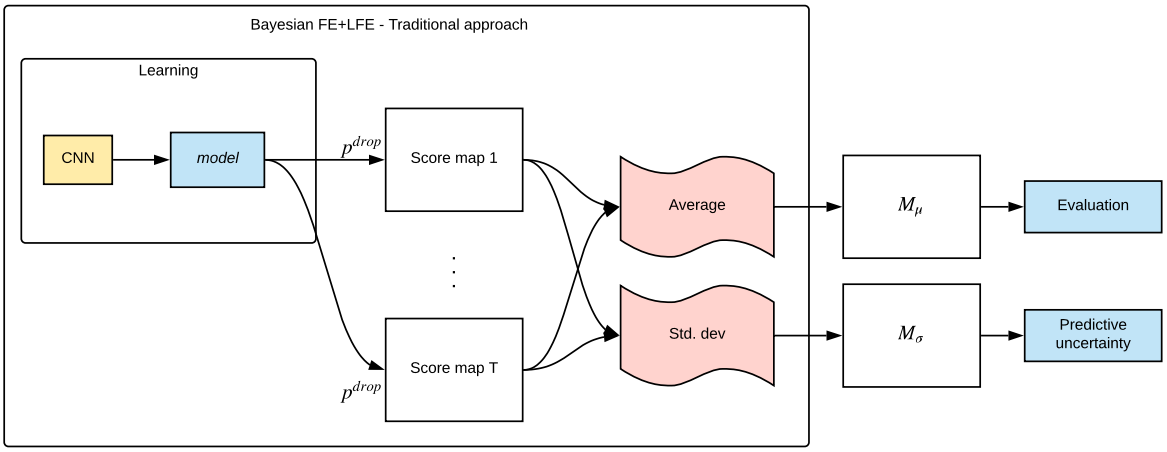


Figure 7.2: CNN-ensemble that traditionally compute mean and standard deviation from the T realizations obtained through MC-dropout.

appy the MC-dropout strategy that by randomly dropping out units with probability p^{drop} allows us to obtain T different realizations of the same posterior distribution.

Now, the traditional strategy computes the mean and the standard deviation out of the T realizations, obtaining $\mathcal{M}_\mu$ and $\mathcal{M}_\sigma$ maps which can be interpreted as a more robust output map and as a measure of model's predictive uncertainty respectively. With the proposed method instead, after having obtained the T realizations we perform a pixelwise BBA allocation based on the deviation from the median value, which allows us to obtain four-layers BBA maps that can be combined together with the conjunctive combination rule. After the combination, the resulting BetP(H) map is interpreted as a robust output, while the resulting mass on the compound set, i.e. $\mathscr{M}(\Theta)$ is interpreted as a measure of model's predictive uncertainty, representing the total ignorance which has not been completely solved by the combination.

7.3.4 Results of CNN-ensemble

The use of an ensemble, along with the already discussed regularization techniques, allows us to apply a deep learning-based solution even in presence of a very small dataset. In particular, as training set we use the pool of data available for the active learning solution for choosing the new samples to add to the training set, noted $\mathscr{U}$, i.e. the pool of unlabeled samples for the active

Table 7.2: Precision-Recall Break Even Point and Area Under Precision-Recall Curve with the different architectures trained on the same limited amount of data.

	SVM-ensemble	CNN	CNN-ensemble Mean	CNN-ensemble Fusion
PRBEP	0.81	0.85	0.85	**0.86**
AUPRC	0.86	0.89	**0.90**	**0.90**

learning. Note that in the traditional active learning an oracle is supposed to answer about the true label of a sample only when it has been chosen by the algorithm to be added to the set, so that $\mathscr{U}$ is indeed a pool of yet unlabeled samples. In our case, the pool is not unlabeled as we dispose of ground-truth maps, nevertheless for consistency we keep the notation "$\mathscr{U}$". Note also that the active learning solution do not use all the available data in $\mathscr{U}$, but selects only 2000 samples out of it; nonetheless, we consider this a rather fair comparison between the two classifiers since the data available to the two methods is a-priori the same.

The training of the deep learning solutions on the small training set $\mathscr{U}$ has been possible only with the FE+LFE network (and not with the U-Net which to start to converge needs at least four times the data in $\mathscr{U}$), thanks to its global relatively small number of parameters kept low by the use of few filters per layer. In order to obtain the CNN-ensemble, we applied MC-dropout method. Dropout is added in the central layers as in [134], i.e. before and after the bottleneck layer with dilation factor equal to 3 (cf. Table 6.1). The probability of dropout p^{drop} is set to 0.2 since the default value of 0.5 resulted to be detrimental for the final result. The number of realizations T is fixed to 10.

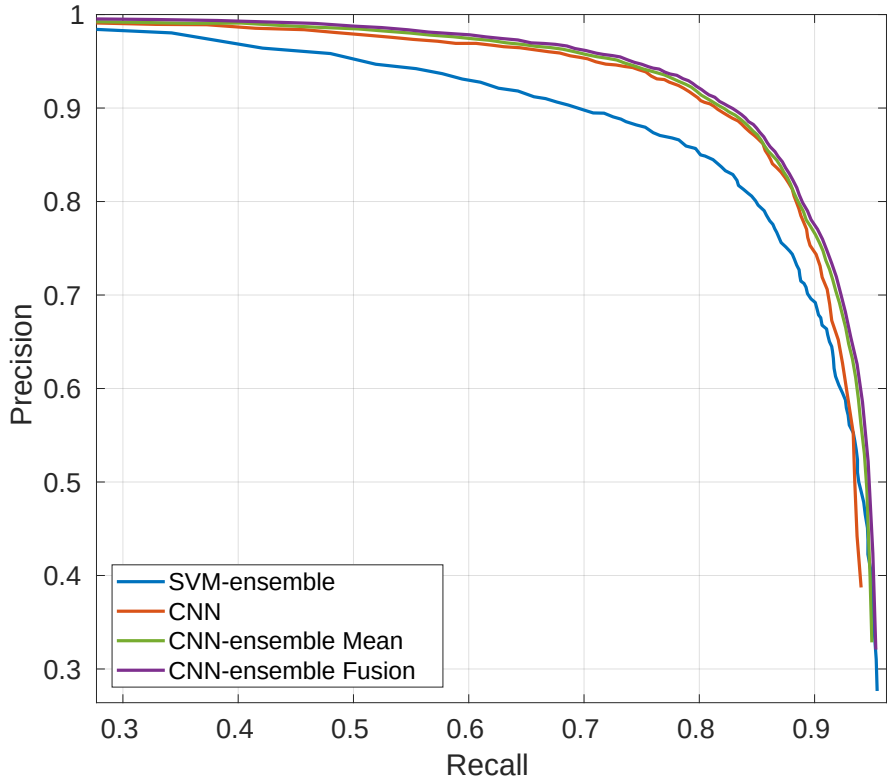


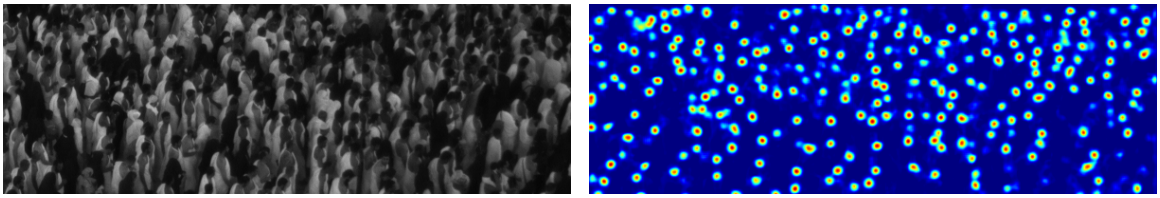
Figure 7.3: PR-curves of SVM-ensemble and deep learning solutions. All the classifiers disposed of the same amount of (limited) data for the training.

Figure 7.3 shows the PR-curves obtained in Chapter 5 with the active learning solution based on the use of a committee of SVMs (denoted as "SVM-ensemble") with the deep learning-based solutions obtained training the network on the same limited amount of data. Specifically, after

training the FE+LFE network, "CNN" refers to the output map obtained with the traditional forward pass to perform inference. "CNN-ensemble Mean" and "CNN-ensemble Fusion" refer instead to the use of MC-dropout to obtain the ensemble, combining the members through the traditional average operator and with the proposed evidential approach respectively. Table 7.2 provides quantitative values for PRBEP and AUPRC with the same names notation.

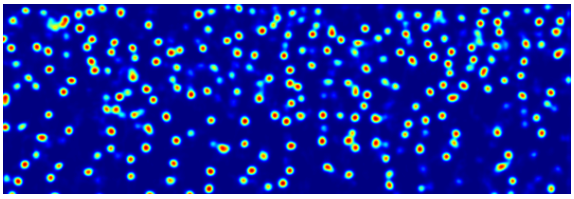
As it is possible to see both from Fig. 7.3 and from Table 7.2, deep learning-based solutions tends to outperform SVM-based one, most noticeably regarding the precision values. Nevertheless, SVM has been trained with a chosen fraction of the available samples pool $\mathscr{U}$, with respect to deep learning-based methods that are able to exploit all the available data.

The use of the CNN-ensemble to perform inference rather than the usual forward pass is beneficial especially in increasing the recall values, meaning that the ensemble is able to retrieve more heads. Note that there is not a great difference between the mean output map $\mathcal{M}_\mu$ (CNN-ensemble Mean) and the BetP(H) map after having performed the fusion of the T realizations (CNN-ensemble Fusion), although this latter is slightly better. However, having defined a BBA allocation for each realization allows us to have a final BBA map that can be easily combined together with the BBA map given by the SVM-ensemble.

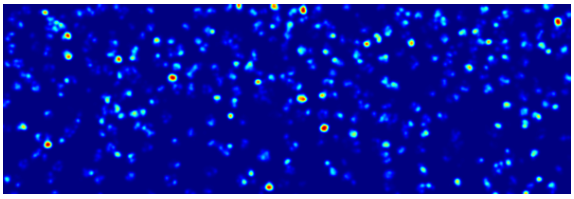


(a) Image patch

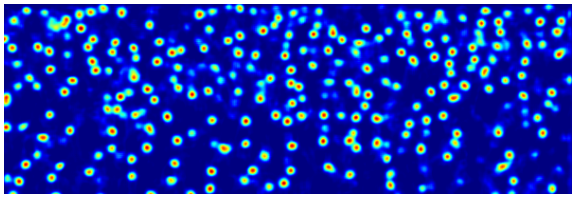
(b) Output map of FE+LFE



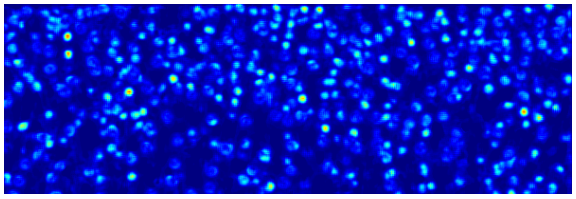
(c) Output map of CNN-ensemble Fusion, i.e. BetP(H)



(d) Ignorance map of CNN-ensemble Fusion, i.e. $\mathscr{M}(\Theta)$



(e) Output map of CNN-ensemble Mean, i.e. $\mathcal{M}_\mu$



(f) Standard deviation map of CNN-ensemble, i.e. $\mathcal{M}_\sigma$

Figure 7.4: Output maps on a testing image patch with the deep learning solutions trained on the same amount of limited data, as well as model's predictive uncertainty outputs through traditional standard deviation and proposed evidential ignorance.

Figure 7.4 shows the final output maps for a given image patch, considering the traditional forward pass for inference in Fig. 7.4b with respect to the CNN-ensemble based output maps in Fig. 7.4c and Fig. 7.4e, respectively obtained through the proposed BBA allocation and evidential fusion, and through the classical average of the T realizations. Figure 7.4d and Fig. 7.4f instead, represent the ignorance map after the evidential conjunctive combination and the classical standard deviation map respectively, which can be interpreted as a measure of predictive uncertainty. The predictive uncertainty map obtained with the proposed evidential method is clearly more precise in the localization of areas where the model is uncertain, while the standard deviation map although being useful is less localized and noisier.

7.4 Final Multiple Classifier System

Until now we have proposed two different ensemble-based methods based on two different classifiers, namely SVM and CNN, and shown that they can achieve remarkable performance even when trained on a small amount of data. In order to obtain the final MCS, we intend to perform a fusion between the two ensembles. Note that this is not straightforward, since in presence of few, strong classifiers the fusion strategy must be particularly well-designed in order to exploit their respective strengths (cf. Sec. 2.3.2.4).

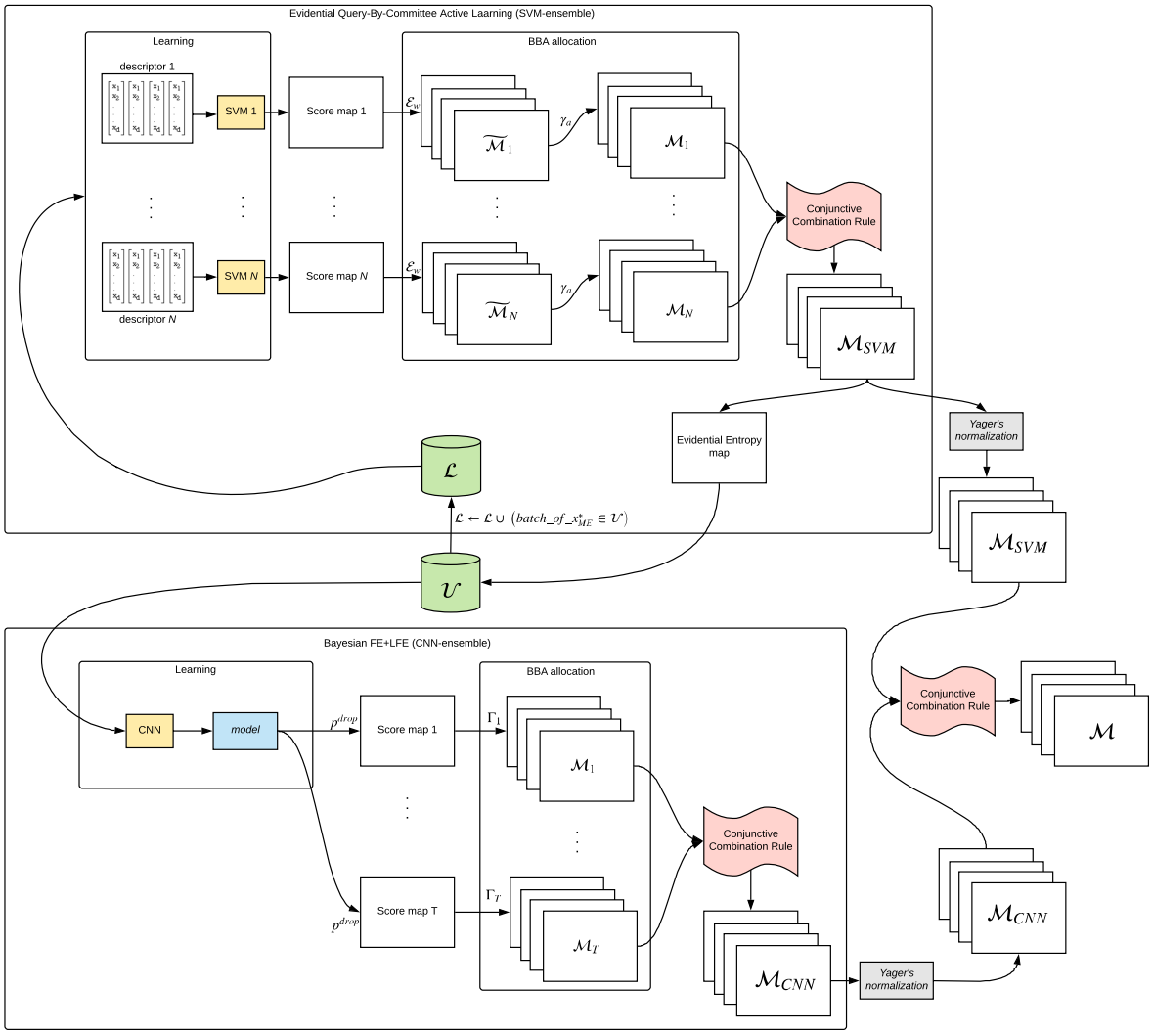


Figure 7.5: Proposed evidential Multiple Classifier System flowchart.

Figure 7.5 shows the overall flowchart of the final evidential MCS. Starting from the initial pool of samples $\mathscr{U}$, we perform in a parallel way the SVM-based active learning procedure to select the most informative samples to be added to $\mathscr{L}$, while at the same time we train the FE+LFE network on $\mathscr{U}$.

To summarize, the evidential QBC active learning procedure consists of the following steps:

- Training the four different SVM classifiers based on different features, i.e. HOG, LBP, GABOR and DAISY (explained in Chapter 3);
- Performing BBA allocation for each pixel of each source, taking into account possible imprecision in the score calibration procedure and in the image space, and combining them through the conjunctive combination rule as explained in Chapter 4;

- Selecting the new samples to be added to the SVM training set $\mathscr{L}$ based on evidential entropy disagreement measures, as explained in Chapter 5.

At the end of the evidential QBC procedure, the result is a single four-layers BBA map ($\mathscr{M}_{\text{SVM}}$) with a BBA associated to each pixel that intrinsically contains evidence of belonging to H and $\overline{\text{H}}$, i.e. $\mathscr{M}_{\text{SVM}}(\text{H})$ and $\mathscr{M}_{\text{SVM}}(\overline{\text{H}})$ respectively, as well as a component of ignorance ($\mathscr{M}_{\text{SVM}}(\Theta)$) which is not solved through the combination and a component of conflict ($\mathscr{M}_{\text{SVM}}(\emptyset)$) that arises through the combination itself.

Regarding the second component of the MCS, namely the deep learning-based one, it also consists of several steps:

- Training the FE+LFE network (whose architecture is presented in Chapter 6) based on the small training dataset $\mathscr{U}$;
- Applying MC-dropout procedure at inference time to obtain the T realizations, as explained in the previous Section 7.2;
- Performing BBA allocation for each realization to model the network's predictive uncertainty about each pixel's prediction based on the deviation from the median value of the sampled posterior distribution, and combining them through the conjunctive combination rule, as explained in the previous Section 7.3.

The output of the proposed evidential CNN-ensemble is thus a single four-layers BBA map ($\mathscr{M}_{\text{CNN}}$), where each pixel contains evidence of belonging to $\emptyset, \text{H}, \overline{\text{H}}, \Theta$ respectively. Here we interpret the ignorance value related to each pixel as the model's predictive uncertainty about it, being able thus to model the imprecision in addition to the uncertainty value provided by the network.

After having combined a relatively high number of sources through the conjunctive rule both to obtain $\mathscr{M}_{\text{SVM}}$ and $\mathscr{M}_{\text{CNN}}$, we note that they both contains not negligible masses on the empty set representing the conflict that arises through the combinations. While the mass on the Θ focal element indeed naturally decreases thanks to the combinations, the more conjunctive combinations we perform, the more the mass on the empty set inevitably increases. This could lead to disproportionate values of conflict with respect to the masses on the other focal elements. To solve this issue, classically Dempster's rule is adopted or a normalization of the BBAs is lately performed, but in this way the conflicting mass would be equally spread over the remaining hypothesis. Instead, as done in [151], we focus on the normalization included in Yager's combination rule [289] that, in the absence of knowledge about the conflict origin, transfers it to the ignorance component.

Finally, the conjunctive combination rule is performed between the normalized $\mathscr{M}_{\text{SVM}}$ and $\mathscr{M}_{\text{CNN}}$, obtaining the final BBA map $\mathscr{M}$ which can be used either for decision, computing the associated BetP(H) map, and to obtain a measure of the imprecision about the final prediction, naturally given by $\mathscr{M}(\Theta)$.

7.4.1 Results of the evidential MCS

To illustrate the benefit of the final evidential MCS, Figure 7.6a shows the PR-curve of the proposed approach described with the flowchart reported in Fig. 7.5, where the SVM-ensemble and CNN-ensemble BBA output maps are combined together after Yager's normalization. PR-curves of SVM-ensemble and CNN-ensemble alone are reported as well, to show the improvement obtained thanks to their fusion.

Figure 7.6b shows the comparison of the proposed approach with respect to two other strategies, namely the fusion between SVM-ensemble and the result of a simple discounting performed on the mean map $\mathcal{M}_{\mu}$ based on the standard deviation values in $\mathcal{M}_{\sigma}$, and the product of BetP(H) maps (interpreted as probability maps) given by the two ensembles. The two initial sources SVM-ensemble and CNN-ensemble are reported as well with dotted lines. Values of PRBEP and AUPRC for the considered approaches are then detailed in Table 7.3.

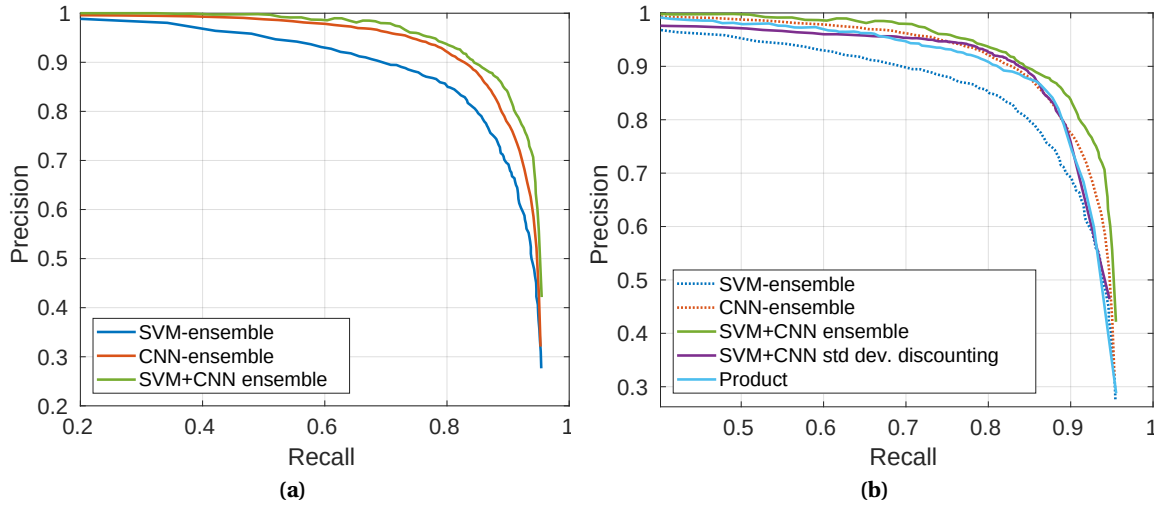


Figure 7.6: (a) PR-curves of SVM-ensemble and CNN-ensemble, along with their combination SVM+CNN ensemble; (b) Comparison in terms of PR-curves of the proposed SVM+CNN ensemble with respect to product of BetP(H) maps given by the two ensembles, and a fusion between the SVM-ensemble BetP(H) map with the result of a simple discounting performed on the mean map $\mathcal{M}_\mu$ based on the standard deviation values in $\mathcal{M}_\sigma$.

Table 7.3: Precision-Recall Break Even Point and Area Under Precision-Recall Curve of the BetP(H) result with the proposed MCS composed by SVM+CNN ensemble, as well as a comparison with respect to product of BetP(H) maps given by the two ensembles, and a fusion between the SVM-ensemble BetP(H) map with the result of a simple discounting performed on the mean map $\mathcal{M}_\mu$ based on the standard deviation values in $\mathcal{M}_\sigma$. SVM-ensemble and CNN-ensemble performances are reported as reference.

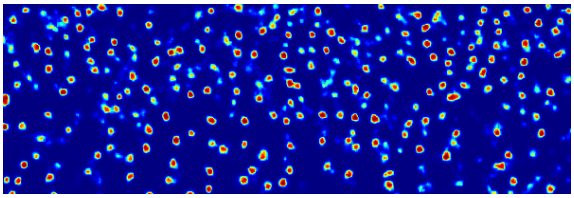
	SVM-ens.	CNN-ens.	SVM+CNN ens.	SVM+CNN std dev. disc.	Product
PRBEP	0.81	0.86	**0.87**	0.86	0.86
AUPRC	0.86	0.90	**0.92**	0.89	0.90

Both from the PR-curves and from the values reported in the table, we can see that the evidential fusion of the two ensembles preceded by Yager's normalization resulted to be the best approach. Conversely, both the product of probabilities and the simpler discounting method fail to exploit all the available information so that the final result do not improve on CNN-ensemble or rather worsen it. This is due to the fact that, being already a map of BBAs obtained after the fusion of the T realizations, CNN-ensemble's BetP(H) map is more informative than the mean map on which we apply a hand-crafted discounting (even though tailored with respect to standard deviation).

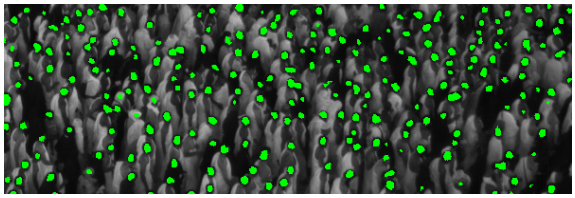
Finally, we can notice that the the proposed MCS system (i.e. SVM+CNN ensemble) is able to reach a value of AUPRC=0.92, which equals the one obtained training the FE+LFE network with all the available data (cf. Table 6.4). This proves that, in presence of few labeled data, the joint use of two classifiers (in our case SVM and CNN) is able to reach competitive performance.

Figure 7.7 provides visual results obtained testing the proposed evidential MCS on a given image patch, in terms of BetP(H) output map, detection map at the threshold corresponding to the PRBEP, and the final ignorance map of the system. The obtained BetP(H) map presents well-localized and well-shaped detections. Regarding the ignorance map, which we interpret as the global system's predictive uncertainty, we notice that it presents higher values in the surrounding of the heads. This is due to the fact that we applied Yager's normalization before the combination of the two ensembles based on the different classifiers, reversing the conflict mass (which is higher at the border of the heads) on the compound set. Thus, a part of the ignorance is not solved with the final combination resulting in the obtained map. Nevertheless, disregarding from the high val-

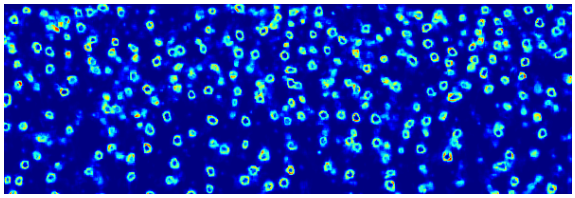
(a) Image patch



(b) Output map of SVM+CNN ensemble, i.e. BetP(H)



(c) Detections at PRBEP threshold



(d) Ignorance map of SVM+CNN ensemble, i.e. $\mathscr{M}(\Theta)$

Figure 7.7: Visual results obtained testing the proposed evidential MCS on an image patch, in terms of BetP(H) output map, detection map at the threshold corresponding to the PRBEP, and the final ignorance map of the system.

ues on the head's borders, the map is interesting in that it highlights the regions where none of the classifiers (nor the SVM nor the deep learning-based one) were able to give a committed answer about the predicted pixel's value.

Chapter 8

Density Estimation

Contents

8.1 Motivation

The objective of this work is to propose methods to perform a complete analysis of specific scenes, that can be useful both for crowd analysis and for synthesis. The modeling community needs indeed detailed information about the studied scene, in order to propose always improved simulations that can be used for two tasks. Firstly, one aim is to help in infrastructure assessment for prevention and security purposes through simulations; secondly, simulations can be used to derive synthesized data that can increase the size of the training set for analysis algorithms, since real-settings data are usually scarce.

To this extent, we have proposed a pedestrian detection method in high-density crowds which is naturally predisposed also for the task of density estimation. Instead of providing as output (whatever the classifier used, or combinations among them) a list of bounding boxes representing the various pedestrians (strategy that do not scale well with the number of people in the crowd), we output real-valued maps which can in turn be analyzed to recover each pedestrian location, as seen until now, or to perform people counting and density estimation.

In this Chapter we explore thus the classifiers and ensembles seen so far, for the specific task of density estimation starting from real-valued outputs which can be interpreted as density maps.

We propose a new evaluation method which can be performed at multiple scales, and by means of BFT we derive upper and lower bounds to the local people count which provides an imprecision interval around the estimation. This overcomes two major limitations that exist in the literature when measuring the performance of a density estimator, namely the fact that a single global number is provided which represents the overall number of people in the image, without any idea about the uncertainty of the estimation and without performing local density estimation at multiple scales.

8.2 State of the art

Crowd counting and density estimation are two related applications which are of paramount importance when performing a macroscopic analysis of the scene. One of the reasons for which the urban infrastructure sector has not fully taken advantage of vast available video data is indeed the difficulty to extract accurately macroscopic observations in high-density conditions.

Although it does not require precise target localization, density estimation inside crowds is still a challenging problem, due to phenomena such as strong occlusion and visual homogeneity. For this reason, the problem of people counting in high-density crowds has attracted significant attention from researchers in the recent past, using a variety of approaches, initially based on hand-crafted features and more recently on learnable ones. A number of surveys are available focusing on traditional methods [171, 231] and deep learning techniques [246] respectively.

Classically, the different methods for people counting are subdivided with respect to the particular approach they are based on:

- *Counting by Detection* – Firstly performs pedestrian detection and then derives the number of people directly from the number of detections;
- *Counting by Regression* – Learns a mapping between features extracted from image patches to their counts;
- *Counting by Density Estimation* – Learns a mapping between patch features and corresponding object density maps, incorporating thus spatial information in the learning process.

8.2.1 Counting by Detection

Most of the initial research was focused on Counting by Detection (CD), which is a straightforward way of approaching counting by delegating the task to a detection algorithm [66, 74]. In this framework, a sliding window detector is used to detect people in the scene and the number of people is simply obtained by counting the overall detections. Many different classifiers based on hand-crafted descriptors have been used for this task. For example, in [160] a part-based detector is used along with a boosting technique for specific body parts such as heads and shoulders.

The main inconvenience of this approach is that relying on the detector to provide crisp detections requires to perform thresholding and non-maximal suppression, which are not adapted in the case of close or partially occluded objects. For this reason, these methods can be successful in low-density crowd scenes but they do not scale well in presence of high-density crowds.

8.2.2 Counting by Regression

Counting by Regression (CR) approaches aim to map image features to the number of objects being present [30, 36]. These methods consist basically in two different steps, namely low-level feature extraction and regression modelling. Once global and local traditional features (such as LBP, HOG, GLCM) have been extracted, different regression techniques such as linear regression [202], piecewise linear regression [32], ridge regression [37], are used to learn a mapping from these features to the actual count of people in the crowd.

Initial works relying on CR and its variations based on region clustering [32] or motion patterns [218] were not aimed at tackling high-density crowds. Count estimation in small crowds is performed in [188] relying on accurate camera calibration and area of projection. However, this strategy is ideally suited for crowds that may be divided into groups of relatively homogeneous densities. In [178] self-organizing neural maps are used to infer the crowd density from image texture, but the task is aimed at identifying the correct density range rather than accurate counting, particularly in a high-density crowd.

In presence of high-density crowds, [119] claims that no single feature is reliable enough to provide sufficient information for an accurate counting, due to severe occlusions, foreshortening and perspective variations. They thus extract different types of features that capture different information, i.e. Fourier analysis along with head detections and SIFT interest point based counting in local neighborhoods. They propose a multi-scale approach where people counts at localized patches are computed independently. These local counts are then globally constrained in a multi-scale Markov Random Field (MRF) framework to get an estimate of count for the entire image.

More recently, [82, 267] were among the first works to rely on deep learning for the task of crowd counting. The former used a modified version of the AlexNet [143], by replacing the last fully connected layer with a single neuron to output the people count. The latter proposed to classify the image into various classes, i.e. very high density, high density, medium density, low density and very low density using a multi-stage CNN.

Even though in occluded scenes CR methods have been shown to be better suited than CD ones, their main limitation is that they do not infer the actual object locations (although their output may be used as a prior for guiding detection and tracking [226]).

8.2.3 Counting by density estimation

Lempitsky et al. [155] were the first researchers to propose a completely different framework, based on performing counting by simultaneously estimating a density map, i.e. Counting by Density Estimation (CDE). In this way, the number of people in a particular area is directly inferred by integrating over that area. They proposed to learn a linear mapping between between local patch features and corresponding object density maps, incorporating thus spatial information in the learning process. Since this seminal work will be fundamental for our objective, we will detail it better in Sec. 8.3.

Motivated by the fact that for complex scenes a linear regression model may be too simple, in [207] a non-linear mapping between local patch features and density maps is learned through random forest regression. Regression trees are also used in [76] to alleviate the computational cost required to solve the optimization program of [155].

Over the last years, deep learning advancements significantly improved the state-of-the-art performance of people counting in high-density crowds. Although there exist some deep learning-based methods which can be identified as CR, most of the recent methods are based on the estimation of a density map, therefore falling under CDE category.

Multi-column Convolutional Neural Network [298] (MCNN) proposes to build a network that is composed by three parallel columns corresponding to filters with receptive fields of different sizes (large, medium, small), to model the density maps corresponding to heads at different scales. Then, the output of the three columns is fused together by learnable 1×1 convolutions (and not by simply averaging the features as classically done in Multi-column Deep Neural Networks (MDNNs)). This method ensures robustness to large variation in object scales.

Again, to take into account possible scale variations, Hydra CNN [199] is trained with pyramid of image patches extracted at multiple scales. These are fed to different Counting CNNs (CCNNs), whose outputs are concatenated and fed to the main body of the network, which consists of two fully-connected layers followed by ReLu, dropout layer and a final fully connected layer to estimate the object density map. While the different CCNNs extract image features at different scales, the body is able to learn a high-dimensional representation that fuses their multi-scale outputs performing a multi-scale non-linear regression.

To maintain a scale-aware approach yet reducing the number of parameters that is usually high when using multi-column networks, a single column fully convolutional network is proposed in [179], by incorporating the scale information into the model with a multi-scale averaging step during prediction. At inference time, the network is fed with multiple scales of the same image. The crowd count is estimated for each scale and the final count is obtained by taking an average of all the estimates.

Cascaded Multi-task CNN is then proposed in [245], to simultaneously learn a density map along with a classification of the crowd into various density levels. Classifying crowd count into various levels is equivalent to coarsely estimating the total count in the image. In this sense, they incorporate a high-level prior into the density estimation network, enabling the layers in the network to learn globally relevant discriminative features.

Very recently, CSRNet [164] has been proposed to overcome some limitations of multi-column approaches, such as the need of a large amount of training time due to the high number of parameters, and a branch structure which may not be always effective for specific scene congestion levels. CSRNet presents a fully convolutional structure composed by two parts, i.e. a front-end module (VGG-16 deprived of the fully connected layers), and a back-end module which makes massive use of 2-dilated convolutions being able to aggregate the multi-scale contextual information as well as to maintain the output resolution.

Whatever the density map learning method used, in the context of CDE framework there is the need to have at training time ground-truth density maps (which possibly take into account perspective correction), rather than just a number specifying the number of people being present in the image as for CR. These ground-truth density maps are usually obtained by placing a Gaussian on each head center, as already explained in Sec. 6.3.1 (as previously anticipated in that section indeed, the proposed network is designed to be able to tackle at the same time pedestrian detection and density estimation tasks).

Perspective correction can be directly applied on ground-truth maps if the geometry of the scene is known. Alternatively, in [298] the authors propose *geometry-adaptive kernels*. By observing that in high-density crowds the head size of a person is directly related to the distance from the head centers of the neighbors, the standard deviation of each Gaussian kernel (which regulates its spatial spread) is adaptively determined based on its average distance to the neighbors. The ground-truth density maps created using this technique incorporate distortion information without the use of perspective maps; however, it is an effective technique only in presence of constant density throughout the image space.

8.3 Learning to count

We now propose a learning-to-count strategy with a generic detection algorithm which benefits from a counting regressor in order to identify crowded subregions with inadequate head detection performance, and to improve their representativeness in the training set.

Our basic assumption is that in some contexts where CDE is known to perform more deficiently, this behavior is not only due to the regression step, but rather to a lack of appropriated data during the learning step. Thus, we propose to mediate through a feedback loop the performance estimated during the regression step. This feedback aims to improve the quality of the input data in areas where the image characteristics are unreliable.

8.3.1 The MESA distance

Under a regularized risk framework, the general objective of CDE methods is to recover a transform defined by a parameter w which maps an estimated density map F to a user-specified ground-truth map G (obtained by placing Gaussian kernels at each head location). This may be formulated as:

$$\hat{w} = \underset{w}{\operatorname{argmin}} \left(w^2 + \lambda \sum_{i=1}^{m} \mathscr{D}(\mathrm{G}(\cdot), \mathrm{F}(\cdot|w)) \right), \tag{8.1}$$

where m is the number of training images, $\mathscr{D}$ is a distance measure and λ a scalar weighting parameter. Note that the factor $\hat{w}$ relating the numerical output to the actual pedestrian count is equal to 1 for deep-learning based methods trained on actual density maps, but in the general case it may be determined as in [155] on a validation set.

In [155], the authors address a major limitation of image-level regressors based on Eq. (8.1) when using as distance measure an absolute or squared difference between the sums over the entire images. Such simple approach requires a large variety of image samples during training. Therefore, they propose a new distance called MESA, which takes into account the mapping penalty for all the possible boxes B within the 2D box space $\mathscr{B}$ of mapping and ground-truth areas:

$$\mathscr{D}_{\mathrm{MESA}}(\mathrm{G}, \mathrm{F}) = \max_{\mathrm{B} \in \mathscr{B}} \left| \sum_{\mathbf{x} \in \mathrm{B}} \mathrm{G}(\mathbf{x}) - \sum_{\mathbf{x} \in \mathrm{B}} \mathrm{F}(\mathbf{x}) \right|, \tag{8.2}$$

where $\mathbf{x} \in \mathrm{B}$ represents each pixel of the box B.

The significant strengths of this distance are an improved robustness to additive local noise, as well as the ability to exploit not only the ground-truth count but also its spatial layout.

With respect to existing image-level regressors, we consider that the MESA approach is better suited for high-density annotated images for the following two reasons. Firstly, as a L_∞ distance between combinatorial sub-area vectors of the ground-truth and of the score map provided by the detector, the MESA distance is ideally suited for a feedback strategy which is aimed at identifying subareas where the input map should be improved. Secondly, many applications such as physical modeling of crowds rely on local density estimations, and through the set of boxes $\mathscr{B}$, the MESA distance considers all image scales in order to achieve better robustness of density estimation across the whole scale space.

8.3.2 Active Learning for count regression

We propose to apply the MESA distance to the probabilistic output of a general detection algorithm, and use the subregion (box) with the most violated constraint provided by the regression in order to select new informative training examples for the detector. In this way, the potential nonlinearity between the feature space and the mapping is dealt with by the learning step, and the regression is used secondarily to pinpoint badly mapped image parts which can provide new valuable training samples. In this sense, the algorithm may be seen as an *objective-driven* active learning with the aim of count regression. Indeed, the objective itself (count regression in our case) is directly involved into the choice of the new training samples that will improve the estimations.

We consider a generic binary classifier which provides for each tested instance (pixel) $\mathbf{x}$ a score $s(\mathbf{x})$ representing the probability of $\mathbf{x}$ belonging to the positive class $\mathrm{P}(y = 1|\mathbf{x})$. Our aim is to recover the scalar factor $\hat{w}$ which maps a density $\mathrm{F}(\mathbf{x}) = \hat{w}\, s(\mathbf{x})$ based on Eqs. (8.1) and (8.2).

Computing the MESA distance may be cast efficiently as a max 2D subarray problem, while determining $\hat{w}$ requires solving a convex QP with a combinatorial number of linear constraints in a tractable manner using cutting-plane optimization [155]. Concurrently with solving for the optimal $\hat{w}$, we identify the box $\tilde{\mathrm{B}}$ corresponding to the maximal mapping error, i.e. the box with maximal MESA distance with respect to the ground-truth map. This allows us to select inside $\tilde{\mathrm{B}}$ the most informative samples that would improve at the next learning iteration the score in the critical area $\tilde{\mathrm{B}}$.

For illustrating our method, we rely on an SVM classifier along with the HOG descriptor. We adopt an uncertainty sampling approach (cf. Sec. 5.2.1), which iteratively requests the labels for the instances whose classes are the most uncertain, i.e. in the context of SVM, the instances which are the closest to the separation hyperplane [234].

Since our potential training set is quite large, we adapt [25] which considers the *diversity* between samples. In particular, the authors propose a selection strategy which aims to reach a trade-off between:

- The minimum distance from the hyperplane margin;
- The maximum angle between the hyperplanes defined by each sample.

Denoting I^* the pool of indexes of available samples with a distance from the hyperplane less than one, the training batch S is built by incrementally adding a new example $\mathbf{x}_t$ such that:

$$t = \underset{i \in I^* \setminus S}{\operatorname{argmin}} \left(\beta \| f(\mathbf{x}_i) \| + (1-\beta) \max_{j \in S} k^*(\mathbf{x}_i, \mathbf{x}_j) \right), \tag{8.3}$$

where $\| f(\mathbf{x}_i) \|$ is the distance of the sample $\mathbf{x}_i$ to the separation hyperplane, and where, given the two sample hyperplanes h_i and h_j and the kernel function k, we have:

$$k^*(\mathbf{x}_i, \mathbf{x}_j) = | \cos(\angle(h_i, h_j)) | = \frac{| k(\mathbf{x}_i, \mathbf{x}_j) |}{\sqrt{k(\mathbf{x}_i, \mathbf{x}_i) k(\mathbf{x}_j, \mathbf{x}_j)}}. \tag{8.4}$$

The β parameter can be tuned to control the trade-off between the classical strategy which takes into account only the distance from the hyperplane and the new approach that combines it with the diversity measure.

Since it is prohibitively costly to compute angles among all the available instances in I^*, we propose a greedy preliminary selection of a potential sample set. Denoting by H the learning batch size, we select the KH examples closest to the hyperplane by using a priority queue over the potential training set with a negligible computational overhead. Then we apply the exhaustive diversity search in terms of cosine similarity among these KH samples, by caching only a K^2H^2 element Gramm matrix. For our needs, we found that K = 10 is adequate, but higher values will promote more diversity with an increased computational cost.

Figure 8.1 shows a visual representation of the proposed AL algorithm for count regression.

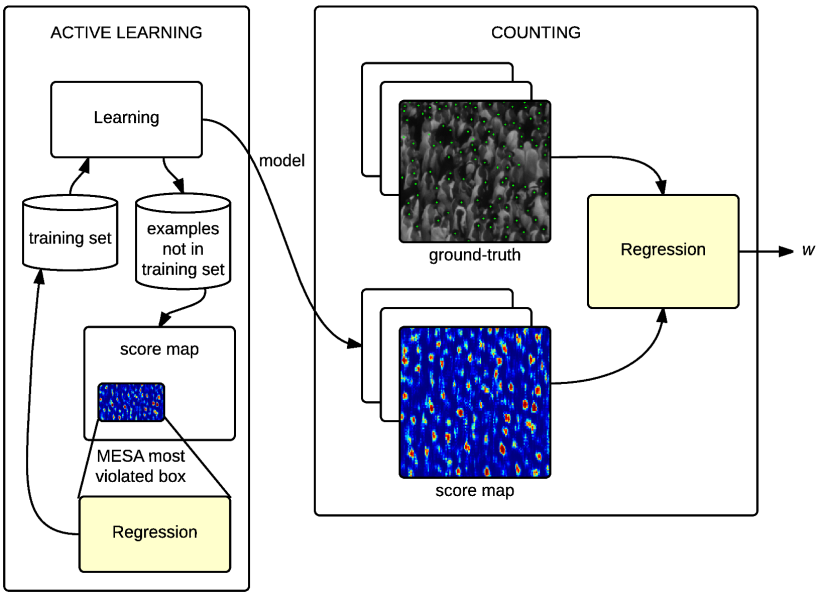


Figure 8.1: A visual representation of the AL for count regression algorithm workflow.

8.3.3 Perspective correction

A detector which has been trained with examples of varying size provides similar pixel-level scores for identical objects which have different sizes in pixels due to the perspective change. This would

affect significantly the MESA hyperparameter $\hat{w}$ which could only settle for an inadequate compromise among the various sizes.

Similarly to [76], we compute a perspective map $\mathscr{M}_{\mathrm{D}}$ based on an accurate camera-to-ground pose estimation [4]. Then we are able to compensate the distortion for pixel $\mathbf{x}$ by multiplying the detector score $s(\mathbf{x})$ with the corresponding factor provided by the distortion map $\mathscr{M}_{\mathrm{D}}$:

$$\hat{s}(\mathbf{x}) = \mathscr{M}_{\mathrm{D}}(\mathbf{x})\, s(\mathbf{x}). \tag{8.5}$$

8.3.4 Results

We compared our new active learning approach with two widely used methodologies: the classical strategy which selects the closest examples to the separation hyperplane, from now on called *distance*, and the *diversity* strategy proposed by [25] explained in Sec. 8.3.2. In order to prove the effectiveness of AL, we compared it also with a *random* strategy, which iteratively selects random examples from the pool.

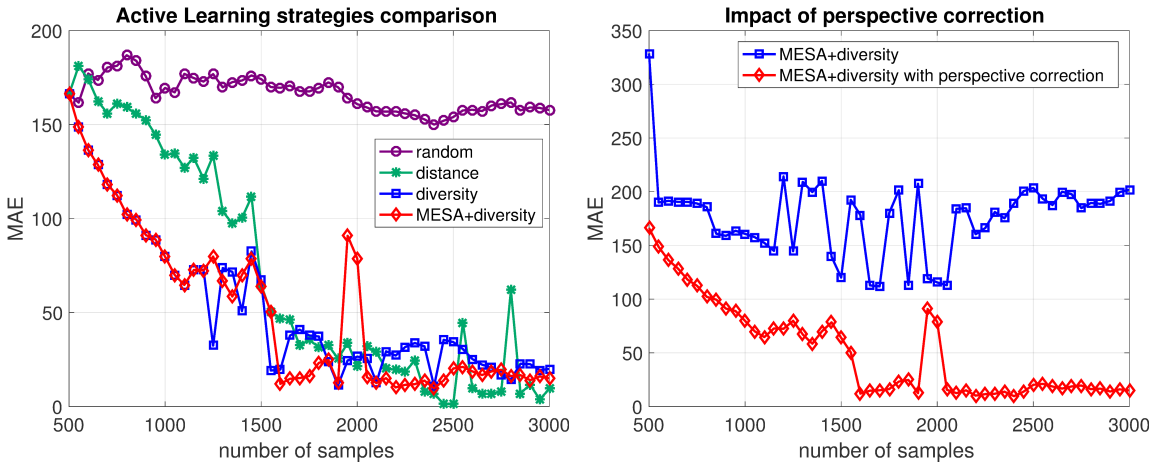


Figure 8.2: (a) Comparison between different active learning strategies. (b) Impact of perspective correction on count estimation.

Figure 8.2a shows the Mean Absolute Error (MAE), namely the absolute error in terms of people count with respect to the ground-truth, averaged over all the testing images. Perspective correction is applied for all the methods. The *random* strategy does not provide meaningful improvements as the training set becomes larger. On the contrary, the errors of all the active learning techniques significantly drop from the beginning. In particular the *distance* approach improves slower, and presents some oscillations even towards the final iterations, while errors for the *diversity* strategy, and for the proposed approach called *MESA+diversity* drop immediately and then remain stable towards the end, highlighting the importance of the variety between the selected samples. It is possible to notice that for the first iterations the samples selected by the two methods based on *diversity* are the same. This happens because the box selected using the MESA distance as the most violated one is very large.

Figure 8.2b shows the importance of the perspective correction for the MESA regression, which compensates the head size variation with respect to the camera. The perspective correction step is crucial in order to obtain a low MAE and a stable behavior.

The proposed approach is applicable to relatively small training sets made up of a few thousands of compact head annotations. Prior information about the geometry of the scene may be easily integrated as well into the algorithm through a perspective correction map. Overall, the proposed strategy is fairly easy to deploy for a given scene.

However, it presents some limitations as well. Firstly, it is based on the use of a single classifier (SVM+HOG), and we observed that it is not easily applicable to other descriptors which benefit less from the addition to the training set of punctual problematic samples. Secondly, although multi-scale information is taken into account to obtain $\hat{w}$, MAE statistic used in the experiments refers only to the global scale and do not take into account local density estimation. Finally, the people count is given as a real-valued number, without highlighting the possible imprecision in the estimation, that can come from the classifier itself and the data it uses to learn from.

For all these reasons, we now propose a new evaluation strategy for ensemble-based methods which can be performed at multiple scales and provides bounds to the estimated count.

8.4 A new evaluation method

Once the output of a density estimator is available, there are multiple avenues for interpreting the result and evaluating the quality of the output. From a traditional point of view, the baseline error is the L_2 distance between the output and the density ground-truth. Two metrics then are widely used as performance indicators for crowd counting, namely the Mean Absolute Error and the (Root) Mean Squared Error ((R)MSE). They are defined as:

$$\mathrm{MAE} = \frac{1}{\mathrm{N}} \sum_{n=1}^{\mathrm{N}} \left| y_n - \hat{y}_n \right|, \tag{8.6}$$

$$\mathrm{MSE} = \frac{1}{\mathrm{N}} \sum_{n=1}^{\mathrm{N}} (y_n - \hat{y}_n)^2, \tag{8.7}$$

where N is the number of images in the testing dataset, y_n is the actual count, and $\hat{y}_n$ is the estimated count for image n.

Beside this measure of quality, other indicators are generally useful as well:

1. The *repeatability* of the algorithm, which aims to evaluate the agreement between the results of successive measurements of the same observation, carried out under the same conditions of measurement [122]. In our particular context, we interpret repeatability as the stability of the output when the configuration of the crowd (in terms of the distribution of the low-density and high-density areas) under a given camera evolves at a large temporal scale (compared with the typical dynamics of pedestrians);

2. The *short-term stability* of the density estimator, which characterizes how the output is affected by small, continuous changes in the pedestrian configuration (caused typically by the ongoing occlusions).

Thus, if repeatability indicates that based on a good performance at a specific moment one may expect a good performance for a different configuration of the crowd at a later time, short-term stability ensures that the output value is not affected locally in a significant manner by a slight change in the input, which denotes a good generalization capability. However, especially for the second indicator which is intrinsically local, global error evaluation metrics (such as the L_2 distance mentioned above) are not adapted since the local perturbations will be summed and the output will benefit from the compensation of the fluctuations, according to the central limit theorem.

All the more so, the density estimation in the case of crowds has a particular interest regarding the locality of the estimation. Some phenomena such as the propagation of the stop-and-go waves, or the perturbation of the flow by a static or dynamic obstacle, can be studied only by relying on the spatio-temporal variation of the density, where the spatial scale is of the order of the direct interaction and observability distance among the pedestrians (0.5-2m). These considerations underline the need to evaluate the density quality using a criterion which is able to characterize the output at multiple scales, going from global to local.

Based on the fundamental idea which is behind the MESA distance, we have adopted for our purpose a multi-scale error function that we denote E_σ as the union $\mathrm{E}_\sigma = \{\mathrm{E}_{\sigma_1}, \cdots, \mathrm{E}_{\sigma_n}\}$, where E_{σ_i} denotes the set of all errors computed for a specific scale:

$$\mathrm{E}_{\sigma_i} = \left\{ \left| \sum_{\mathbf{x}\in \mathrm{S}_j} \mathrm{G}(\mathbf{x}) - \sum_{\mathbf{x}\in \mathrm{S}_j} \mathrm{F}(\mathbf{x}) \right|, \forall \mathrm{S}_j \in \mathscr{S}_i \right\}. \tag{8.8}$$

Here, $\mathscr{S}_i$ denotes the set of all square boxes of a given size l_i that may be contained within the image space. The largest size l_1 is defined by the largest square fitting inside the image space, and for $i > 1$ we have:

$$l_i = \frac{l_{i-1}}{\delta}, \tag{8.9}$$

where the scale factor δ controls the transition between consecutive scales and is set typically in the range $[1.1, 2]$. A minimal value l_{min} is set in order to avoid considering squares below a very small size comparable to the direct interaction distance between two pedestrians such as 0.3-0.4m for example, and thus l_n is the last size for which $l_n > l_{min}$.

One significant drawback of Eq. (8.8) is that the error magnitudes depend on the actual density of the crowd and cannot be used as such in order to evaluate it on images with various degrees of density. Thus, it is highly informative to rely as well on a normalized error measure that we denote as the relative multi-scale error function $\tilde{\mathrm{E}}_\sigma$, constructed similarly except for the fact that now we normalize the local errors by the local density:

$$\tilde{\mathrm{E}}_{\sigma_i} = \left\{ \left| \frac{\sum_{\mathbf{x}\in \mathrm{S}_j} \mathrm{G}(\mathbf{x}) - \sum_{\mathbf{x}\in \mathrm{S}_j} \mathrm{F}(\mathbf{x})}{\sum_{\mathbf{x}\in \mathrm{S}_j} \mathrm{G}(\mathbf{x})} \right|, \forall \mathrm{S}_j \in \mathscr{S}_i \right\}. \tag{8.10}$$

The relative error computation requires however to check whether the denominator $\sum_{\mathbf{x}\in \mathrm{S}_j} \mathrm{G}(\mathbf{x})$ is sufficiently large in order to avoid division by small values. For our experiments, we impose at least half a head to be present, i.e. $\sum_{\mathbf{x}\in \mathrm{S}_j} \mathrm{G}(\mathbf{x}) > 0.5$, which is verified almost every time for high-density crowds and for scales which remain relevant for a physical interpretation.

In order to compute the values required by Eqs. (8.8) and (8.10), the process may be accelerated significantly by using the Integral Histogram [213] trick, given that the most intensive task is to compute sums over rectangular supports defined in the bounded image space.

8.4.1 Multi-scale error statistics

For each scale, some relevant statistics are computed on the elements of the corresponding error vector, E_{σ_i}. Assuming the absence of a significant bias of the density estimator, the elements of E_{σ_i} should follow a folded normal distribution. However, in order to preserve the generality of the analysis, we compute the median m_{σ_i} along with the lower and upper quartiles $q^{25}_{\sigma_i}$ and $q^{75}_{\sigma_i}$, as well as the maximum value observed in E_{σ_i} denoted as M_{σ_i}, which corresponds thus to the largest overestimation or underestimation observed for the considered scale in all the analyzed locations.

The corresponding statistics $\tilde{m}_{\sigma_i}, \tilde{q}^{25}_{\sigma_i}, \tilde{q}^{25}_{\sigma_i}, \tilde{\mathrm{M}}_{\sigma_i}$ for the relative error vector $\tilde{\mathrm{E}}_{\sigma_i}$ are computed in a similar manner.

Although this type of multi-scale analysis is more informative than the single value provided by MAE, offering a statistical analysis based on the aggregation of multiple boxes information at the same scale, it is not able as such to provide real uncertainty bounds that can be tied to an error rate.

8.4.2 Uncertainty bounds

An additional limitation of current density estimators is the absence of an uncertainty range provided along with the scalar density. Ranges on the pedestrian count are greatly needed, as the trade-off between safety concerns and optimal use of infrastructure capacity promotes different

levels of congestion in different contexts. Count estimation with an uncertainty range has been proposed in [9] for the similar task of counting penguins in the wild, but the ranges are derived by the fact that the dataset has been dot-annotated by several people.

We instead propose a generic approach for evaluating the uncertainty about the output of a crowd density estimator. Then, we apply the proposed evaluation on a multi-scale domain derived from the image lattice, which allows us to characterize the estimator performance locally as well.

Until now, we created two ensembles in the context of BFT, an SVM-ensemble and a CNN-ensemble, obtained though BBA allocations that account for different types of imprecision that may arise from the specific base classifier. Whatever this latter, the result of the fusion among the ensemble members is a multiple layers map of BBAs $\mathscr{M}$ from which we derive the BetP(H) map.

Now, we propose a multi-scale evaluation strategy which computes for each considered scale $\mathscr{S}$ indicators based on all squared subdomains $S \in \mathscr{S}_i$. These indicators use the derived upper and lower density bounds $\underline{s}(S), \overline{s}(S)$ such that:

$$\underline{s}(S) = \hat{w} \sum_{\mathbf{x} \in S} \mathrm{Bel}_{\mathbf{x}}(H), \tag{8.11}$$

$$\overline{s}(S) = \hat{w} \sum_{\mathbf{x} \in S} \mathrm{Pl}_{\mathbf{x}}(H). \tag{8.12}$$

The factor $\hat{w}$ relating the numerical output to the actual pedestrian count may be determined with Eq. (8.1) on a validation set consisting of BetP(H) maps.

We then calculate for $\mathscr{S}_i$ the *Prediction Error Probability* (PEP) as:

$$\mathrm{PEP}_i = \Big| \{ S \in \mathscr{S}_i | G(S) \notin [\underline{s}(S), \overline{s}(S)] \} \Big| / |\mathscr{S}_i|, \tag{8.13}$$

and the *Relative Imprecision* (RI) interval as:

$$\mathrm{RI}_i = \Big(\sum_{S \in \mathscr{S}_i} (\overline{s}(S) - \underline{s}(S)) / G(S) \Big) / |\mathscr{S}_i|, \tag{8.14}$$

where G(S) is the ground-truth count over S. We take $\mathscr{S}_1$ as the set of the largest possible squares which fit the image space, and then we use a scale factor δ to reduce the square side for subsequent scales, according to Eq. (8.9)

The RI criterion highlights the size of the imprecision interval around the estimated count, while the PEP criterion indicates the error rate of the prediction, namely whether the ground-truth count for the considered region is outside the estimated interval. Thus, a two-axis plot presenting the evolution of RI vs. PEP across multiple scales and for different estimators allows one to compare them and to select an operating point with an explicit uncertainty tied to a desired error rate.

8.5 Results

8.5.1 Multi-scale statistical evaluation

In order to perform a multi-scale evaluation of the density estimators, we apply the error functions proposed in Eqs. (8.8) and (8.10) and we perform the multi-scale analysis presented in Sec. 8.4.1.

Figures 8.3 and 8.4 show the results obtained with SVM classifier and FE+LFE network respectively. Statistics are computed for 20 scales, where scale 1 represents the largest one (i.e. given by the largest square that fits into the image). Results are computed for every box of the image at a given scale, and then they are shown in terms of the median m_{σ_i} of the obtained error distribution at every scale, along with lower and upper quartiles $q^{25}_{\sigma_i}$ and $q^{75}_{\sigma_i}$.

Figure 8.3 shows the comparison of multi-scale error statistics for SVM-ensemble classifiers obtained with the proposed evidential QBC AL method using Lamata and Moral's entropy definition (explained in Chapter 5), with respect to the baseline that builds a training set with random

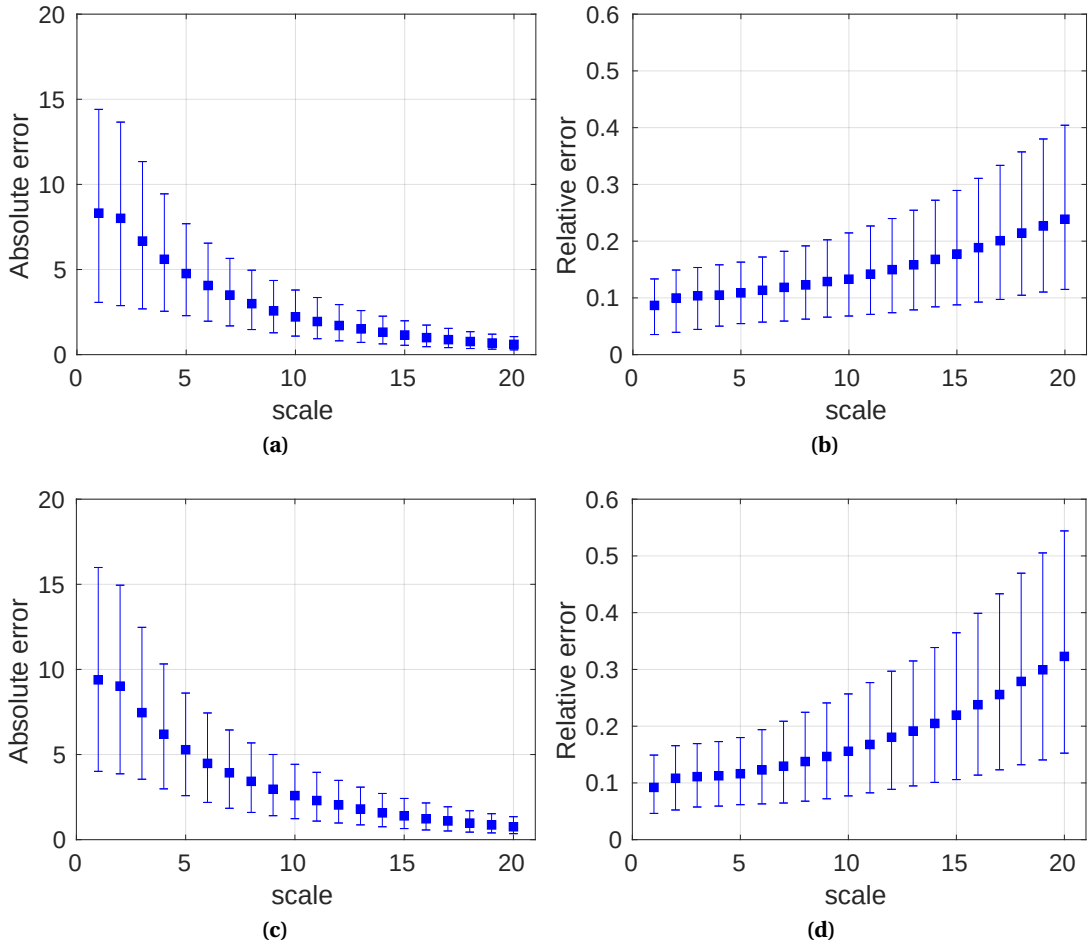


Figure 8.3: Comparison of multi-scale error statistics for SVM-ensemble classifiers obtained with the proposed evidential QBC active learning using Lamata and Moral's entropy definition (first row) vs. random sample selector (second row), in terms of absolute errors E_{σ_i} (first column) and relative errors $\tilde{E}_{\sigma_i}$ (second column).

samples. Overall, the benefits introduced by the active learning procedure are not only useful to perform microscopic analysis through pedestrian detection, but also to more accurately perform macroscopic analysis through density estimation. Specifically, active learning is better at estimating *local* density, proved by the fact that the relative errors at larger scales (i.e. smaller boxes) for the active learning solution are lower than the errors of the random baseline, presenting at the same time smaller variation around the median value among the various boxes at a given scale.

Figure 8.4 shows the comparison of multi-scale error statistics computed with the proposed FE+LFE network trained with a limited amount of data (i.e. the pool of the available samples for the active learning approach), and trained with all the available data. Compared to SVM, we notice that with deep-learning based solutions the errors are more consistent at every scale, proven by the almost constant median relative error throughout the different scales. This is indeed a very desirable property for a density estimator. Training the network with more data, besides reducing the overall errors at every scale, seems to be particularly important in order to obtain more consistent results for every box at a given scale, since the bounds given by the quartiles values are smaller.

However, although being more informative than traditionally employed global metrics such as MAE, this type of multi-scale evaluation is only able to offer a statistical analysis based on the aggregation of multiple boxes information at the same scale, without providing real uncertainty bounds around the estimation that are more useful for a complete analysis of the scene, especially for the synthesis community that could tie the uncertainty bounds to an error rate.

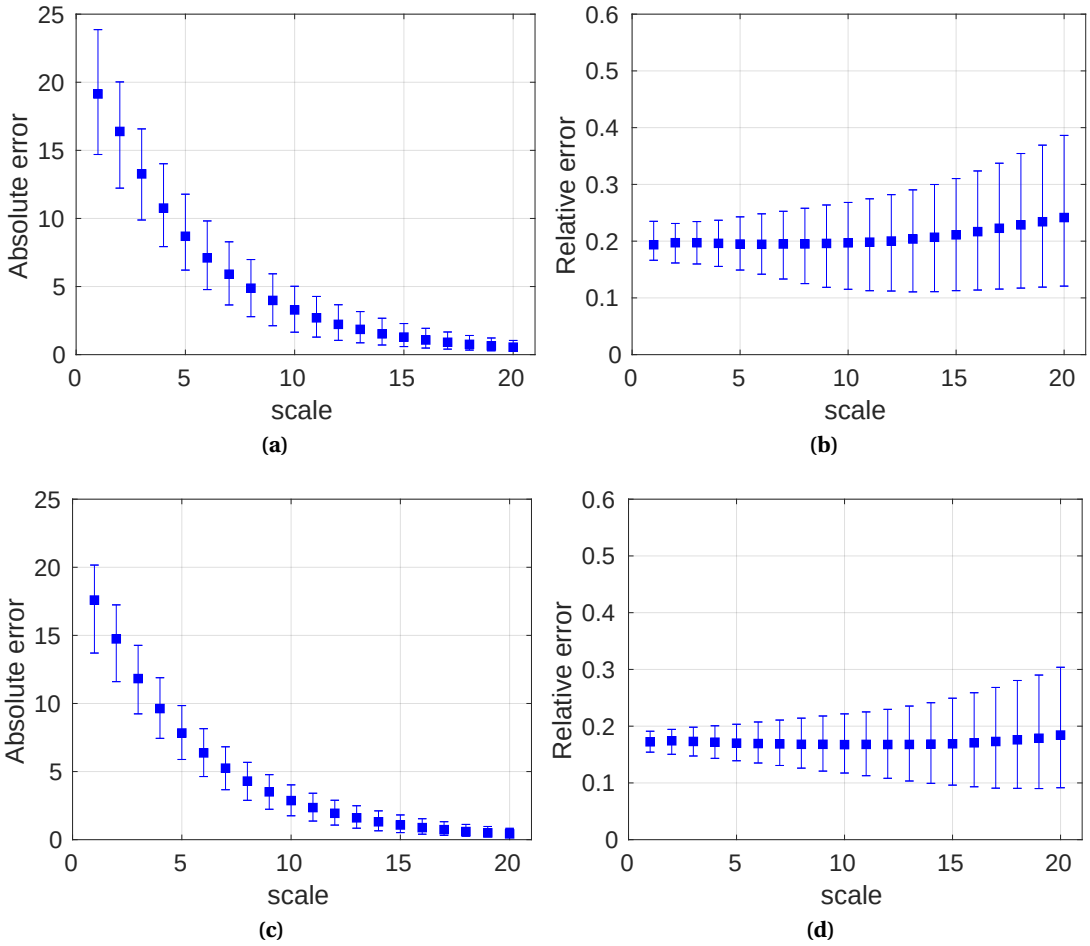


Figure 8.4: Comparison of multi-scale error statistics for the proposed FE+LFE network trained on a limited amount of data (first row) and trained with all the available data (second row), in terms of absolute errors E_{σ_i} (first column) and relative errors $\tilde{E}_{\sigma_i}$ (second column).

8.5.2 Multi-scale uncertainty bounds evaluation

To overcome the limitations of the multi-scale statistical analysis performed in the previous Section, we evaluate now the different density estimators based on the proposed two-axis plot evaluation explained in Sec. 8.4.2, that allows us to evaluate not only density point estimates but also uncertainty bounds associated to the estimations.

Note that only in presence of an ensemble of classifiers, one is able to apply the evidential combination among them and thus obtain besides the probabilistic output map Bel(H) also upper and lower bounds provided by Pl(H) and Bel(H) maps respectively. To this extent, we did experiments using the proposed evidential SVM-ensemble and CNN-ensemble. Concerning CNN-ensemble, note that to obtain it the probability of dropout p^{drop} is set to 0.5, a larger value with respect to the one set in the previous Chapter to perform pedestrian detection, but that resulted to be more appropriate for the density estimation task.

Figure 8.5 shows the results of the density estimator evaluation with the proposed RI vs. PEP plot at multiple scales and with respect to different discounting amounts, regulated by the α parameter of Eq. (7.6) for the CNN-ensemble based approaches, and performing additional global discounting for the SVM-ensemble. In the figures, each horizontal cluster of points corresponds to a different discounting factor.

Specifically, Figs. 8.5a and 8.5b show the results when applying the proposed uncertainty bound evaluation to ensembles obtained with the FE+LFE and U-Net networks respectively. Ideally, an estimator should predict with a high confidence (low PEP) that the estimated count is within a

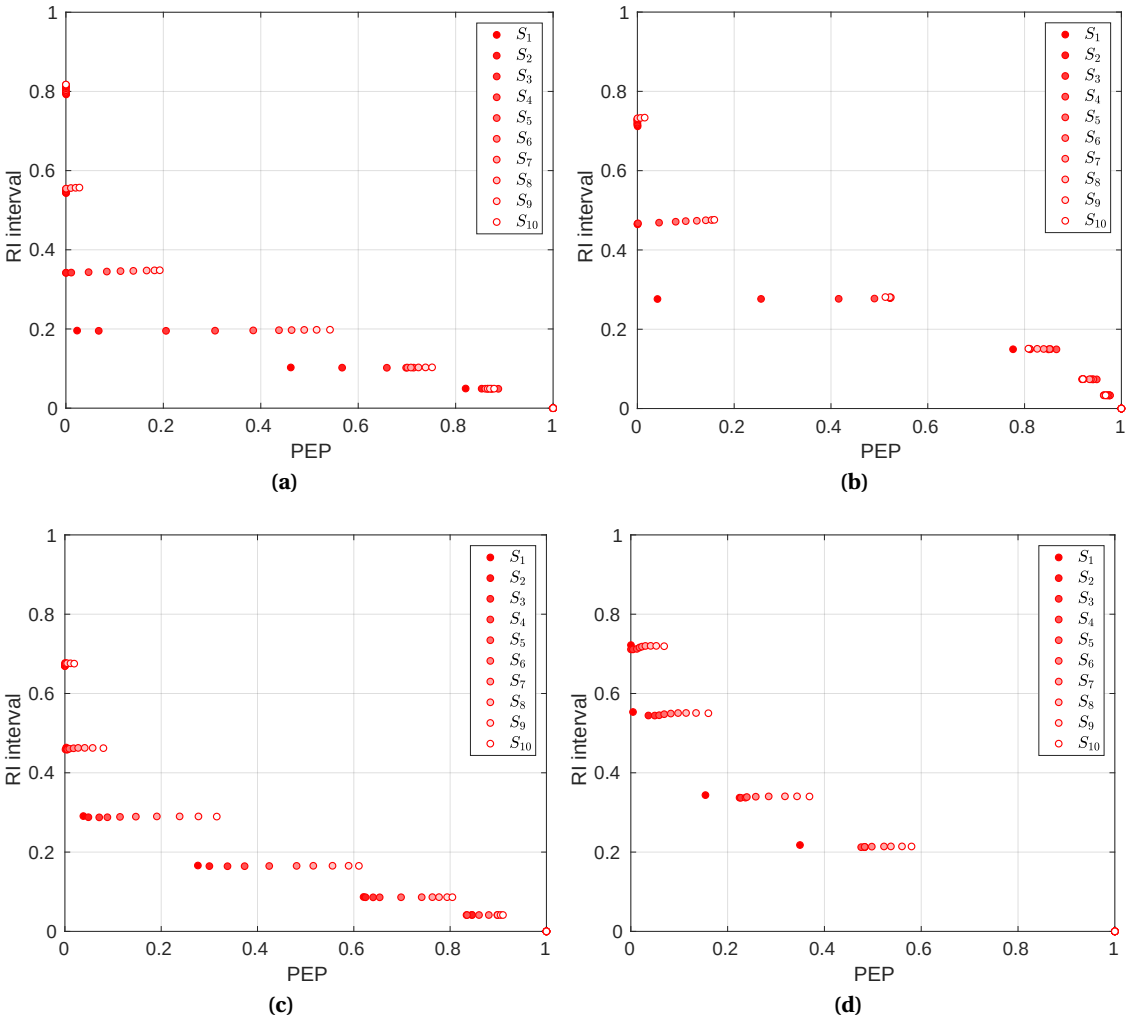
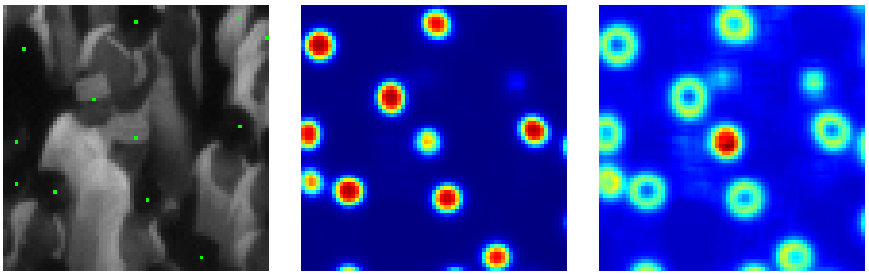


Figure 8.5: Density estimator evaluation with the proposed RI vs. PEP plot at multiple scales and with different discounting amounts. Each horizontal cluster corresponds to a different discounting factor. (a) CNN-ensemble based on FE+LFE network; (b) CNN-ensemble based on U-Net; (c) CNN-ensemble based on FE+LFE network trained on a limited amount of data; (d) SVM-ensemble.

small RI interval. One may increase the size of the RI interval by decreasing the α parameter in Eq. (7.6), in order to obtain better prediction accuracy (at the expense of a larger RI). We tested different discounting factors, corresponding to horizontally aligned clusters of dots. For each cluster, each dot depicts the performance obtained at a different scale, with a scale factor $\delta = 1.1$, $\mathscr{S}_1$ being the largest scale. Both networks perform better at larger scales, due to error compensation. The proposed FE+LFE network outperforms U-Net, showing the importance of preserving spatial information without pooling operations in presence of small targets while increasing at the same time the contextual information with dilations, not only to perform pedestrian detection but also to perform density estimation.

Figure 8.5d shows the results of the density estimation obtained with the SVM-ensemble obtain with the evidential QBC active learning procedure. Moreover, Fig. 8.5c shows the results obtained training the proposed FE+LFE network with a smaller amount of data (i.e. the pool of unlabeled samples $\mathscr{U}$ available for AL). This allows us to perform two different types of analysis. Firstly, we can perform a fairer comparison between the two ensemble of classifiers. To this extent, we notice that FE+LFE, even when trained on less data, outperforms the SVM-based approach, especially at larger scales. Nonetheless, the two methods exhibit almost identical performance when considering the smaller scales. Secondly, it is interesting to evaluate the same network trained with different amounts of data. According to Figs. 8.5a and 8.5c, we see that a larger training set is

beneficial for density estimation especially at larger scales. However, considering smaller scales, the performance gap is consistently reduced, indicating thus an implicit limit in the network capacity (increasing the number of layers and/or filters per layer could help, paying attention to overfitting).



(a) Image patch S, G(S) = 12.3

(b) BetP(H) output map, $s(\mathrm{S}) = 12.01$

(c) $\mathscr{M}(\Theta)$ output map, $\overline{s}(\mathrm{S}) - \underline{s}(\mathrm{S}) = 3.2$

Figure 8.6: Visual results of the density estimation map along with the estimated uncertainty bounds.

Figure 8.6 provides a visual example of uncertainty bounds computation around the estimated count, with respect to CNN-ensemble obtained with the FE+LFE network. Figure 8.6a shows an image patch with corresponding ground-truth count (obtained after Gaussian smoothing). Figure 8.6b shows the resulting BetP(H) map which represents the scalar density estimation map, while Fig. 8.6c shows the imprecision map $\mathscr{M}(\Theta)$ (in our case for pixel $\mathbf{x}$ the imprecision value $\mathrm{Pl}_{\mathbf{x}}(\mathrm{H}) - \mathrm{Bel}_{\mathbf{x}}(\mathrm{H})$ is equal to $m_{\mathbf{x}}(\Theta)$). The values in $\mathscr{M}(\Theta)$ may be interpreted as the predictive uncertainty, and provide a bound for the density estimation itself. For the given region S indeed, by integrating over the BetP(H) map we obtain the estimated number of people within it. Similarly, integrating over the $\mathscr{M}(\Theta)$ map we obtain the imprecision interval $\overline{s}(\mathrm{S}) - \underline{s}(\mathrm{S})$. Then, the corresponding RI interval is given by $(\overline{s}(\mathrm{S}) - \underline{s}(\mathrm{S}))/g(\mathrm{S}) = 0.26$, so that we can conclude that in S there are $12.01 \pm 13\%$ heads, i.e. $s(\mathrm{S}) \in [10.4, 13.6]$. Moreover, from Fig. 8.6c we can notice that, in addition to head edges, ignorance is particularly high on heads with lower gradient on the borders and strong clutter, reflecting in a smaller confidence about the prediction. Finally as expected, we underline the desirable effect of ignorance being higher in circularly-shaped areas (e.g. shoulders, or round dark blobs) which are similar to heads, even if they have a low corresponding score.

Conclusion and future work

Conclusion

In this work we address the problem of high-density crowd understanding by proposing a Multiple Classifier System in a mono-camera setting that can be employed to perform both microscopic and macroscopic analysis of the scene.

Chapter 1 introduces state of the art methods about crowd research, which comprises two main fields of study, i.e. crowd analysis, mostly performed by the Computer Vision community through the analysis of real scenes, and crowd synthesis, mostly performed by Mathematics, Physics and Computer Graphics communities, that deals with crowd simulations. We point out the fact that although focusing on the same entity, i.e. a *crowd*, the two fields have evolved almost independently over the years. Nevertheless, we highlight some works where crowd synthesis and analysis benefit from each other, and we aim at placing our work in this category since we propose methods to perform complete analysis of specific scenes, which can be easily fed into simulations and may be later exploited by the synthesis community.

Chapter 2 introduced the theory related to two major aspects of this work, i.e. the supervised learning techniques that have been exploited (namely logistic regression, SVM and neural networks) and a general introduction about ensemble methods. We indeed propose the use of ensemble methods to perform supervised detection, through the definition of a heterogeneous MCS based on both SVM and CNN classifiers.

Chapter 3 firstly highlights the SVM descriptors that are more adapted to perform pedestrian detection in high-density crowds, and then provides their results on the Makkah dataset that we use to validate our methods. By analysing the results using a single descriptor we understand that a single descriptor is not enough in presence of difficult applications. We nonetheless underline the complementarity among the chosen descriptors, a highly desirable property required in order to be able to perform a successful fusion among them.

To this extent, in Chapter 4 we propose a fusion method in the context of Belief Function Theory which is able to consider the imprecision in addition to the uncertainty value provided by the classifiers. After an introduction about fundamental concepts of BFT, we propose a BBA allocation which is based on the fact that imprecision in SVM learning can come from two different sources, namely during the logistic calibration procedure to derive probabilistic outputs out of the SVM scores (distance from the hyperplane margin), and in the image space due to neighborhood heterogeneity which is caused by the close resolution of the objects (heads) and descriptor respectively which is computed at every pixel through a sliding window.

Noting that the obtained results are generally highly dependent on the training set, especially in presence of few data, we put in place in Chapter 5 a QBC active learning procedure to allow the algorithm to automatically choose the data from which to learn. We propose three criteria in the evidential framework which are based on maximizing an evidential entropy measure (several definitions are tested), conflict or ignorance components. While the Maximum Conflict criterion adds to the training set points about which the initial sources, i.e. descriptors, completely disagree, the Maximum Ignorance criterion adds to the training set points about which all classifiers agree about the fact that they are not at all certain about the true label. The Maximum Entropy criterion results to be a trade-off between the two, being able to select automatically informative training

samples balancing exploration of the feature space and exploitation of the current hyperplane margins. We show that, by integrating the evidential combination of the SVM classifiers in the learning loop, the results are much better because the information coming from the combination itself is exploited in the choice of the training samples for the subsequent iteration. The proposed SVM-ensemble is therefore composed of the four SVM detectors described, trained on the training set defined through the proposed active learning strategy.

We then investigate deep learning solutions for our problem. In Chapter 6 we propose to cast the detection problem as a segmentation problem in presence of soft labels, since we do not have at our disposal labeled segmentation ground-truth but just sample coordinates. We thus model each head as a Gaussian in a cumulative context such that the evidences of a head's presence weighted by the Gaussian are summed up for each pixel. Regarding the chosen network architecture, we propose the use of a Front End module with increasing dilation factors in convolutional layers to consider more context, followed by a Local Feature Extractor module which aggregates the features by decreasing the convolution dilation factor. The absence of pooling layers then allows for the detection of extremely small objects.

In Chapter 7 we point out a major criticism that is often made to deep learning techniques, namely the fact that they often act as black-boxes, rendering difficult the interpretation of the final results. We therefore cast the network as a BNN and propose the use of the MC-dropout method to draw samples from a Bernoulli distribution over the network weights. In this way we once more rely on ensemble methods, obtaining a CNN-ensemble which is composed of multiple dropout-perturbed versions of the same network. Still in the context of BFT, we propose a BBA allocation which is based for each pixel on its distance to the median value of the realizations, and we perform an evidential combination of the sources. Then, we show that the proposed BBA allocation provides better results in terms of detections on the BetP(H) map rather than the traditional averaging operation. In the same way, predictive uncertainty estimated after the evidential combination as the ignorance mass results to be visually more indicative than traditional standard deviation to highlight possibly problematic areas for the learned model. Lastly, after Yager's normalization of the two ensembles (SVM and CNN based ones), they are fused together and show competitive performance even in the presence of a small training set.

Finally, in Chapter 8 we tackle crowd analysis at a macroscopic level to perform density estimation and people counting. After an introduction about the related state of the art, we highlight a major limitation in the evaluation of common density estimators, namely the fact that the evaluation is performed at a global scale, allowing for local compensations, and without providing any uncertainty bounds about the estimated count. From our part, we propose a new evaluation method which exploits a generic ensemble of classifiers in order to obtain evidential upper and lower bounds to the actual count with plausibility and belief functions respectively.

The designed MCS based on the joint use of both SVM-based and CNN-based ensembles allows us to obtain high levels of performance even in the presence of a limited amount of training data, as it is often the case in the field of specific crowd studies. The output of the MCS is a probabilistic map, and this allows us to be able to easily perform both microscopic and the macroscopic analysis. Indeed, detections can be obtained by a simple thresholding operation (e.g. PRBEP is common operative threshold), while density estimation can be performed by integrating over each (local) region of interest. For this reason, we find that the proposed approach helps in reaching a complete understanding of a given scene and could be easily integrated into simulation models concerning both levels of granularity. The analysis of specific scenes is indeed very important for many applications to avoid more stampedes and crowd disasters in the future. It can be useful both for an a-priori study of urban space and architectural design, to validate simulated models, and for an a-posterior analysis of crowded situations, to be able to model more realistic scenarios.

However, in order to reach a complete understanding of a given scene while at the same time being robust to scene variations, several challenges still remain open and will be listed in the following section.

Future work

The dataset

First of all, there is the need of validating the proposed approach with other datasets. The availability of dot-annotated dataset for high-density crowds composed of video sequences of specific scenes is still very scarce, but we hope that in the future more data will be available to the community thanks to the involvement of more and more researchers in the field of high-density crowd studies.

Concerning the SVM-ensemble, the applicability of the method to different datasets remains to be validated. We think that a multi-scale extension of the considered descriptors should be explored in presence of possibly strong perspective variation, possibly considering dimensionality reduction techniques on the final feature vectors. Concerning the CNN-ensemble instead, the deep learning solutions are notably more robust to different data, and the performed validation of the proposed network on the Regent's Park dataset (which presents generally larger heads and medium-density crowd) shows the robustness of the method to scale and density variations.

Still concerning the training data, we could think to extend the proposed method on the whole images of the Makkah dataset (cf. Fig. A.1 in Appendix A). However, performing dot-annotation for the upper part of the Makkah images is sometimes impossible even for a trained human (besides being a hard and time-consuming task implying the availability of trained people to perform it). Nevertheless, we could think about taking into account another source of imprecision in the system, namely the imprecision about ground-truth labeling information which is used for the training. However, it is still not clear how to model this type of imprecision in the context of BFT at inference time.

Lastly, in presence of datasets with synchronized multiple views, it will be interesting to exploit multiple cameras to perform a fusion of the detections obtained with our single camera-based approach. Multiple cameras have been successfully exploited in [71] and more recently in [205], proving the potential of multiple views in presence of occluded scenes. However, in these works the crowd density is not extreme and extending this type of solutions to very dense scenarios will not be trivial, besides the fact that up to our knowledge at present no annotated datasets composed of multiple cameras in presence of a high-density crowd exist.

Toward tracking

In order to really reach a complete understanding of the scene, temporal information has to be exploited. This work lays the foundation for the design of a tracking-by-detection approach that, starting from the probability map of the detections at time t, tracks each person individually in the high-density crowd for the subsequent frames. This would be very useful to obtain precise insights about abnormal behaviours, or to study possible bottlenecks that prevent the other people to advance.

As done in [226], density estimation information can be directly exploited in the system, e.g. through the minimization of a joint energy function incorporating both probability scores of individual detections and *local* density estimation. From our part, we dispose also of uncertainty bounds around the estimated density, and of an associated prediction error probability that could be exploited as well.

Figure 8.7 shows the preliminary results of data association performed on two consecutive frames through Hungarian Algorithm [145]. The preliminary results obtained show that this is nonetheless a promising avenue to be better examined.

The Hungarian Algorithm (known also as Kuhn-Munkres algorithm) is a combinatorial optimization algorithm that solves the assignment problem in polynomial time. The problem is represented by a cost matrix, which in our case indicates the cost of associating each detection at time t_0 with each detection and time t_1, and the algorithm finds the optimal associations on a one-to-one basis in order to minimize the overall cost. We design the cost of association of two detections

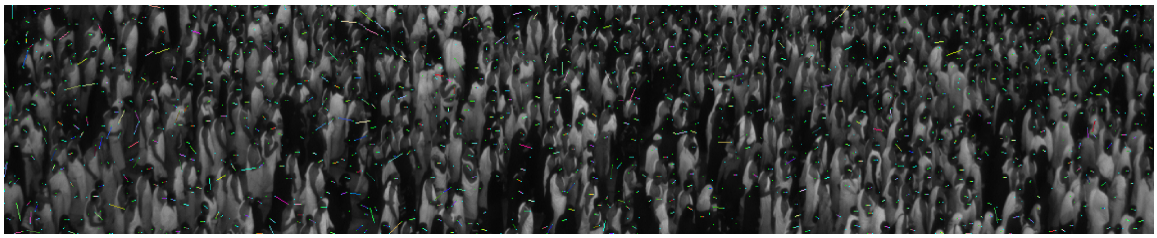


Figure 8.7: Preliminary data association results.

at frame t_0 and t_1 respectively as the distance between the detection at frame t_1 and the projection of the detection at frame t on t_1 through Optical Flow.

Specifically, we firstly employ the proposed MCS in order to obtain probability output maps (i.e. BetP(H) maps) both at frame t_0 and t_1. Detections are then obtained through NMS at the PRBEP threshold. Denoting by $\mathbf{p}_{0,i} = (x_{0,i}, y_{0,i})$ the coordinates of the i^{th} detection at frame t_0, and by $\mathbf{p}_{1,j} = (x_{1,j}, y_{1,j})$ the coordinates of the j^{th} detection at frame t_1, the cost of associating the two detections is given by:

$$C(i,j) = \left\| \mathbf{p}_{1,j} - \mathbf{p}_{1,i}^{\text{proj}} \right\| \tag{8.15}$$

where $\mathbf{p}_{1,i}^{\text{proj}}$ is the projection of $\mathbf{p}_{0,i}$ at frame t_1 through Optical Flow computed following [28].

Hungarian algorithm has been successfully employed to perform object tracking through data association, e.g. in [115] where hierarchical association is performed at different levels, firstly among reliable tracklets and then refining the final trajectories. The concept of weak and strong detections on the basis of a confidence score has been exploited also in [232] in the context of online multi-target tracking, resulting to be beneficial for real-time performance. For this reason, we think that information coming from the final BBA map should be exploited in order to perform tracking taking into account the reliability of each detection. Even further, we can think about associating a single BBA to every detection starting from the BBAs at pixel level which compose the detection itself, and then performing BBA tracking. Nonetheless, tracking-by-detection approaches in the context of BFT have been successfully proposed e.g. in [222, 223], being able to model the imprecision about the location of each (possibly fragmented) detection.

Finally, note that in order to precisely evaluate the tracking algorithm, annotations for each frame of the sequence are needed and should be performed (for the moment the annotated images of the Makkah dataset are temporally independent frames), or a new dataset which has this type of temporally-consistent annotations should be considered.

Exploring deep-learning

Over the last years, deep learning made advances that allowed for major breakthroughs in computer vision and image processing fields. However, there is still much room of improvement and the methodologies are bound to continue to evolve in the near future. For this reason, we think that several paths exist in order to improve our method taking into account deep learning solutions.

Firstly, concerning the loss function, the use of the optimal transport or Wasserstein distance when confronting the output to the ground-truth map is a very promising perspective, but the computational cost is significantly higher. However, recent works [249] explore the applicability of this family of distances for 2D domains. With respect to the simple L2 loss function, the Wasserstein distance should be more adapted to data which represent density information.

Secondly, an important avenue to be considered is the possible learning of spatio-temporal representations, e.g. with the use of 3D convolutions [260], providing as input of the network temporal sequences (3D volumes) instead of 2D images, to jointly perform detection and tracking. The problem of 3D convolutions resides in the higher number of training parameters required. They are usually applied only in presence of small 3D volumes due to the high computational

cost required to compute and store the gradients and high memory consumption. The training of a 3D CNN is thus very computationally expensive and the model size has a quadratic growth with respect to the number of layers compared to 2D CNNs, making it extremely difficult to train a very deep 3D CNN. However, recent advances propose the use of pseudo-3D convolutions [217] by simulating $3\times3\times3$ convolutions with $1\times3\times3$ convolutional filters in the spatial domain plus $3\times1\times1$ convolutions to construct temporal connections on adjacent feature maps in time. They show performance improvements over traditional techniques by encapsulating pseudo 3D convolutions inside residual blocks, obtaining thus a Pseudo-3D (P3D) ResNet, in the contexts of video action recognition, action similarity labeling and scene recognition. The applicability of the P3D ResNet in presence of 3D volumes with large 2D extent is still to be proven but constitutes an interesting research path.

Lastly, we could think of a joint use of deep learning techniques and Belief Function framework. Instead of deriving final BBAs starting from Bayesian BBAs and then applying a discounting which is based on the availability of an ensemble of maps, we could design a network that directly outputs a BBA associated to each pixel which intrinsically contains information about the reliability of pixel's prediction in the mass on Θ. How to directly measure the reliability of pixel's prediction within the forward pass is however still unclear and deserves better investigation.

Exploring active learning

In this work we employed active learning with an ensemble of SVM classifiers to automatically select the most informative training samples on the basis of evidential measures of disagreement computed after having performed the combination among committee members. We devised three different criteria, namely Maximum Conflict, Maximum Ignorance and Maximum (evidential) Entropy. The conflict we consider is simply given by the mass on the empty set. However, other conflict measures exist and it would be interesting to test them. For example, in [60] the authors proposed a conflict measurements based on contour functions, making no prior assumptions about the possible dependence between sources.

Active learning is classically employed in presence of traditional classifiers (or ensemble). It is indeed based on iteratively training the classifier by adding the new selected data to the training set, so that the training procedure must be somehow lightweight. For this reason it has seldom been applied to deep-learning based methods, which require the optimization of millions of parameters during the training step. However, in recent years, the applicability of active learning techniques in presence of deep networks is being more and more explored.

For example [131] proposes a fine tuning in a continuous learning scenario. Fine tuning is an approach which is included in transfer learning methods, i.e. methods where knowledge gained during training of one type of problem is used to train another related task or domain. In fine-tuning, a model is learned based on some training data, and then it is specialized by retraining it (or part of it) on some more data. This is often done for image classification tasks related to specific domains, where a network is trained on the huge ImageNet dataset and then it is specialized by training the last layers on the new data proper to the particular application (which possibly include other classes as well). In this way, the first layers are kept frozen because they deals with generic feature representation, while we let the network update the weights of the final layers which are related to specific feature extraction. The amount of layers to re-train depends on the quantity of new data available and also on the specific task. Continuous fine tuning becomes thus a way to perform a lightweight partial re-training of the network including the new samples which can be chosen in an active learning scenario.

How to select the samples to be added to the network training set is still unclear. In [130] the authors propose a new generalization of the Expected Model Output Change (EMOC) principle [236] for deep architectures to actively select relevant batches of unlabeled examples for annotation. In [268] instead, new samples are added based on least confidence or marginal confidence criteria, i.e. in a multi-class context selecting the samples whose Softmax score for the predicted class is the the lowest, or selecting the samples with smallest difference between the scores asso-

ciated to the first and second most probable class labels predicted by the classifiers.

Finally, an interesting approach is proposed in [280] where active and reinforcement learning are jointly exploited. Classically, methods for active learning involve strategies such as selecting the data points for which the model is the most uncertain, or for which an ensemble of classifiers mostly disagrees, in order to pick the examples which are *expected to be* the most informative ones based on some heuristics. The authors of [280] go even further, proposing the use of reinforcement learning to *learn* which are the best samples to consider based on the choice of a reward function which rewards accurate predictions and penalizes incorrect predictions and label requests.

We thus think that it will be interesting to evaluate active learning solutions not only related to SVM classifier but also with respect to the CNN-based one. In our context however, since a fully convolutional network is employed, the active learning task would imply the selection of informative images (or region of interests) rather than single pixels as is done with SVM. This could be done e.g. considering areas with highest relative imprecision interval performing density estimation, in order to link both microscopic and macroscopic analysis.

Bibliography

[1] Abdel-Hakim, A. E. and Farag, A. A. (2006). Csift: A sift descriptor with color invariant characteristics. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1978–1983. IEEE. 36

[2] Aghajan, H. and Cavallaro, A. (2009). *Multi-camera networks: principles and applications*. Academic press. 4

[3] Ahonen, T., Hadid, A., and Pietikäinen, M. (2004). Face recognition with local binary patterns. *ECCV*, pages 469–481. 45

[4] Aldea, E. and Kiyani, K. H. (2014). Hybrid focal stereo networks for pattern analysis in homogeneous scenes. In *ACCV 2014 Workshops*, pages 695–710. 121

[5] Ali, I. and Dailey, M. N. (2012). Multiple human tracking in high-density crowds. *Image and vision computing*, 30(12):966–977. 5

[6] Ali, S. and Shah, M. (2007). A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6. IEEE. 6

[7] Allain, P., Courty, N., and Corpetti, T. (2009). Crowd flow characterization with optimal control theory. In *Asian Conference on Computer Vision*, pages 279–290. Springer. 7

[8] Andrade, E. and Fisher, B. (2005). Simulation of crowd problems for computer vision. In *First International Workshop on Crowd Simulation (V-CROWDS '05)*, pages 71–80. 7

[9] Arteta, C., Lempitsky, V., and Zisserman, A. (2016). Counting in the wild. In *European Conference on Computer Vision*. 124

[10] Badrinarayanan, V., Kendall, A., and Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*. 87, 104

[11] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer. 36

[12] Bazzani, L., Cristani, M., and Murino, V. (2013). Symmetry-driven accumulation of local features for human characterization and re-identification. *Computer Vision and Image Understanding*, 117(2):130–144. 36

[13] Bellman, R. E. and Zadeh, L. A. (1970). Decision-making in a fuzzy environment. *Management science*, 17(4):B–141. 28, 49

[14] Beucher, S. and Meyer, F. (1992). The morphological approach to segmentation: the watershed transformation. *Optical Engineering-New York-Marcel Dekker Incorporated-*, 34:433–433. 94

[15] Bisagno, N., Conci, N., and Zhang, B. (2017). Data-driven crowd simulation. In *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*, pages 1–6. IEEE. 7

[16] Blake, A. and Isard, M. (1997). The condensation algorithm-conditional density propagation and applications to visual tracking. In *Advances in Neural Information Processing Systems*, pages 361–367. 5

[17] Bloch, I. (2008). Defining belief functions using mathematical morphology–application to image fusion under imprecision. *Int. journal of approximate reasoning*, 48(2):437–465. 54

[18] Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM. 14

[19] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*. 103

[20] Bordes, A., Ertekin, S., Weston, J., and Bottou, L. (2005). Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research*, 6(Sep):1579–1619. 66

[21] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM. 17

[22] Boughorbel, S., Tarel, J.-P., and Boujemaa, N. (2005). Conditionally positive definite kernels for svm based image recognition. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 113–116. IEEE. 21

[23] Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140. 26, 29, 68

[24] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32. 29

[25] Brinker, K. (2003). Incorporating diversity in active learning with support vector machines. In *ICML*, pages 59–66. 68, 69, 120, 121

[26] Brostow, G. J. and Cipolla, R. (2006). Unsupervised bayesian detection of independent motion in crowds. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 594–601. IEEE. 5

[27] Brown, G., Wyatt, J., Harris, R., and Yao, X. (2005). Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20. 30

[28] Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, pages 25–36. Springer. 132

[29] Bruno, L., Corbetta, A., and Tosin, A. (2016). From individual behaviour to an evaluation of the collective evolution of crowds along footbridges. *Journal of Engineering Mathematics*, 101(1):153–173. 3

[30] Cavazza, J. and Murino, V. (2015). People counting by Huber loss regression. In *ICCV Workshops*. 116

[31] Cebron, N. and Berthold, M. R. (2009). Active learning for object classification: from exploration to exploitation. *Data Mining and Knowledge Discovery*, 18(2):283–299. 67

[32] Chan, A. B., Liang, Z.-S. J., and Vasconcelos, N. (2008). Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7. IEEE. 116, 117

[33] Chang, M.-C., Krahnstoever, N., and Ge, W. (2011). Probabilistic group-level motion analysis and scenario recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 747–754. IEEE. 4

[34] Change Loy, C., Gong, S., and Xiang, T. (2013). From semi-supervised to transfer counting of crowds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2256–2263. 8

[35] Chao, C., Cakmak, M., and Thomaz, A. L. (2010). Transparent active learning for robots. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, pages 317–324. IEEE. 66

[36] Chen, K., Gong, S., Xiang, T., and Change Loy, C. (2013). Cumulative attribute space for age and crowd density estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2467–2474. 116

[37] Chen, K., Loy, C. C., Gong, S., and Xiang, T. (2012). Feature mining for localised crowd counting. In *BMVC*, volume 1, page 3. 116

[38] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848. 87

[39] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM. 30

[40] Cheriyadat, A. M. and Radke, R. J. (2008). Detecting dominant motions in dense crowds. *IEEE Journal of Selected Topics in Signal Processing*, 2(4):568–581. 5

[41] Cobb, B. R. and Shenoy, P. P. (2006). On the plausibility transformation method for translating belief function models to probability models. *International journal of approximate reasoning*, 41(3):314–330. 52, 70

[42] Cohn, D., Atlas, L., and Ladner, R. (1994). Improving generalization with active learning. *Machine learning*, 15(2):201–221. 66

[43] Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1995). Active learning with statistical models. In *Advances in neural information processing systems*, pages 705–712. 67

[44] Colombo, R. M. and Rosini, M. D. (2005). Pedestrian flows and non-classical shocks. *Mathematical methods in the applied sciences*, 28(13):1553–1567. 2

[45] Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619. 4

[46] Conde, C., Moctezuma, D., De Diego, I. M., and Cabello, E. (2013). Hogg: Gabor and hog-based human detection for surveillance in non-controlled environments. *Neurocomputing*, 100:19–30. 41

[47] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297. 22

[48] Courty, N., Allain, P., Creusot, C., and Corpetti, T. (2014). Using the agoraset dataset: Assessing for the quality of crowd video analysis methods. *Pattern Recognition Letters*, 44:161–170. 7, 9

[49] Courty, N. and Corpetti, T. (2007). Crowd motion capture. *Computer Animation and Virtual Worlds*, 18(4-5):361–370. 7

[50] Dagan, I. and Engelson, S. P. (1995). Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings 1995*, pages 150–157. Elsevier. 67

[51] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893. IEEE. xvii, 3, 36, 37

[52] Damianou, A. and Lawrence, N. (2013). Deep gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215. 103

[53] Daniel, M. (2010). Conflicts within and between belief functions. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 696–705. Springer. 71

[54] Daniel, M. (2011). Non-conflicting and conflicting parts of belief functions. In *ISIPTA*, volume 11, pages 149–158. Citeseer. 71

[55] Daugman, J. G. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *JOSA A*, 2(7):1160–1169. 39

[56] Dehghan, A., Modiri Assari, S., and Shah, M. (2015). Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4091–4099. 5

[57] Deng, Y. (2016). Deng entropy. *Chaos, Solitons & Fractals*, 91:549–553. 70, 71

[58] Denœux, T. (2006). The cautious rule of combination for belief functions and some extensions. In *2006 9th International Conference on Information Fusion*, pages 1–8. IEEE. 107

[59] Denoeux, T. (2016). 40 years of Dempster-Shafer theory. *International Journal of Approximate Reasoning*, 79:1–6. 28, 49

[60] Destercke, S. and Burger, T. (2013). Toward an axiomatic definition of conflict between belief functions. *IEEE transactions on cybernetics*, 43(2):585–596. 71, 133

[61] Dezert, J. (2002). Foundations for a new theory of plausible and paradoxical reasoning. *Information and Security*, 9:13–57. 71

[62] Dezert, J. and Smarandache, F. (2004). Presentation of DSmT. In *Advances and Applications of DSmT for Information Fusion*, pages 3–35. American Research Press. 71

[63] Dezert, J., Smarandache, F., and Tchamova, A. (2003). On the Blackman's association problem. In *Proceedings of the 6th Annual Conference on Information Fusion*, pages 1371–1378. 71

[64] Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer. 26

[65] Dietterich, T. G. and Bakiri, G. (1994). Solving multiclass learning problems via error-correcting output codes. *Journal of artificial intelligence research*, 2:263–286. 31

[66] Dollar, P., Wojek, C., Schiele, B., and Perona, P. (2012). Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):743–761. 116

[67] Donmez, P., Carbonell, J. G., and Bennett, P. N. (2007). Dual strategy active learning. In *European Conference on Machine Learning*, pages 116–127. Springer. 67

[68] Dubois, D. and Prade, H. (1987). Properties of measures of information in evidence and possibility theories. *Fuzzy sets and systems*, 24(2):161–182. 70, 71

[69] Dupont, C., Tobías, L., and Luvison, B. (2017). Crowd-11: A dataset for fine grained crowd behaviour analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 9–16. 9

[70] Ellis, A. and Ferryman, J. (2010). Pets2010: Dataset and challenge. *AVSS, 00 (undefined)*, pages 143–150. 8

[71] Eshel, R. and Moses, Y. (2010). Tracking in a dense crowd using multiple cameras. *International journal of computer vision*, 88(1):129–143. 4, 131

[72] Fabbri, M., Lanzi, F., Calderara, S., Palazzi, A., Vezzani, R., and Cucchiara, R. (2018). Learning to detect and track visible and occluded body joints in a virtual world. In *European Conference on Computer Vision (ECCV)*. 9

[73] Fei-Fei, L. and Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE. 36

[74] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645. 3, 116

[75] Ferryman, J. and Shahrokni, A. (2009). Pets2009: Dataset and challenge. In *2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 1–6. IEEE. 8

[76] Fiaschi, L., Koethe, U., Nair, R., and Hamprecht, F. A. (2012). Learning to count with regression forest and structured labels. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2685–2688. IEEE. 117, 121

[77] Fischer, P. and Brox, T. (2014). Image descriptors based on curvature histograms. In *German Conference on Pattern Recognition*, pages 239–249. Springer. 36

[78] Fradi, H., Luvison, B., and Pham, Q.-C. (2017). Crowd behavior analysis using local mid-level visual descriptors. *IEEE Trans. Circuits Syst. Video Techn.*, 27(3):589–602. 6

[79] Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285. 26, 29

[80] Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139. 29, 68

[81] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232. 30

[82] Fu, M., Xu, P., Li, X., Liu, Q., Ye, M., and Zhu, C. (2015). Fast crowd density estimation with convolutional neural networks. *Engineering Applications of Artificial Intelligence*, 43:81–88. 117

[83] Fumera, G., Pillai, I., and Roli, F. (2004). A two-stage classifier with reject option for text categorisation. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 771–779. Springer. 29

[84] Gal, Y. (2016). Uncertainty in deep learning. *University of Cambridge*. 104, 105

[85] Gal, Y. and Ghahramani, Z. (2015). Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*. 105

[86] Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. 102, 103

[87] Gao, T. and Koller, D. (2011). Active classification based on value of classifier. In *Advances in Neural Information Processing Systems*, pages 1062–1070. 31

[88] Gao, Y., Liu, H., Sun, X., Wang, C., and Liu, Y. (2016). Violence detection using oriented violent flows. *Image and vision computing*, 48:37–41. 6

[89] Goatin, P., Colombo, R. M., and Rosini, M. D. (2009). A macroscopic model for pedestrian flows in panic situations. In *4th Polish-Japan Days*, volume 32, pages 255–272. 2

[90] Gönen, M. and Alpaydın, E. (2011). Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12(Jul):2211–2268. 29, 64

[91] Gong, S., Cristani, M., Yan, S., and Loy, C. C. (2014). *Person re-identification*. Springer. 8

[92] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572. 104

[93] Graves, A. (2011). Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356. 103

[94] Gray, D. and Tao, H. (2008). Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *European conference on computer vision*, pages 262–275. Springer. 36

[95] Guerrero, R., Qin, C., Oktay, O., Bowles, C., Chen, L., Joules, R., Wolz, R., Valdés-Hernández, M., Dickie, D., Wardlaw, J., et al. (2018). White matter hyperintensity and stroke lesion segmentation and differentiation using convolutional neural networks. *NeuroImage: Clinical*, 17:918–934. 90

[96] Hall, E. T. et al. (1959). *The silent language*, volume 3. Doubleday New York. ix

[97] Hamaguchi, R., Fujita, A., Nemoto, K., Imaizumi, T., and Hikosaka, S. (2018). Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1442–1450. IEEE. xxii, 84, 90, 91, 92

[98] Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001. 33

[99] Hassner, T., Itcher, Y., and Kliper-Gross, O. (2012). Violent flows: Real-time detection of violent crowd behavior. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 1–6. IEEE. 6

[100] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE. 94

[101] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034. 94

[102] Helbing, D., Farkas, I., and Vicsek, T. (2000). Simulating dynamical features of escape panic. *Nature*, 407(6803):487. 2

[103] Helbing, D., Johansson, A., and Al-Abideen, H. Z. (2007). Dynamics of crowd disasters: An empirical study. *Physical review E*, 75(4):046109. x, 8

[104] Helbing, D. and Molnar, P. (1995). Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282. 1

[105] Helbing, D. and Mukerji, P. (2012). Crowd disasters as systemic failures: analysis of the love parade disaster. *EPJ Data Science*, 1(1):7. x, 8

[106] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*. 24

[107] Ho, T. K., Hull, J. J., and Srihari, S. N. (1994). Decision combination in multiple classifier systems. *IEEE transactions on pattern analysis and machine intelligence*, 16(1):66–75. 28

[108] Höhle, U. (1982). Entropy with respect to plausibility measures. In *Proceedings of the 12th IEEE international symposium on multiple-valued logic*, pages 167–169. 70, 71

[109] Hoi, S. C., Jin, R., Zhu, J., and Lyu, M. R. (2006). Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd international conference on Machine learning*, pages 417–424. ACM. 66

[110] Hosang, J., Omran, M., Benenson, R., and Schiele, B. (2015). Taking a deeper look at pedestrians. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4073–4082. 4, 83

[111] Howley, T. and Madden, M. G. (2006). An evolutionary approach to automatic kernel construction. In *International Conference on Artificial Neural Networks*, pages 417–426. Springer. 21

[112] Hu, P. and Ramanan, D. (2017). Finding tiny faces. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1522–1530. IEEE. 84, 91

[113] Hu, R., Wang, R., Shan, S., and Chen, X. (2014). Robust head-shoulder detection using a two-stage cascade framework. In *ICPR*, pages 2796–2801. 36

[114] Hu, X. (2001). Using rough sets theory and database operations to construct a good ensemble of classifiers for data mining applications. In *icdm*, page 233. IEEE. 30

[115] Huang, C., Wu, B., and Nevatia, R. (2008). Robust object tracking by hierarchical association of detection responses. In *European Conference on Computer Vision*, pages 788–801. Springer. 132

[116] Huang, S.-J., Jin, R., and Zhou, Z.-H. (2010). Active learning by querying informative and representative examples. In *Advances in neural information processing systems*, pages 892–900. 67

[117] Huang, Y. S. and Suen, C. Y. (1993). The behavior-knowledge space method for combination of multiple classifiers. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 347–352. IEEE. 28

[118] Hughes, R. L. (2003). The flow of human crowds. *Annual review of fluid mechanics*, 35(1):169–182. 2

[119] Idrees, H., Saleemi, I., Seibert, C., and Shah, M. (2013). Multi-source multi-scale counting in extremely dense crowd images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2547–2554. 6, 8, 117

[120] Iglovikov, V. and Shvets, A. (2018). Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. *arXiv preprint arXiv:1801.05746*. 93

[121] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*. 24

[122] ISO, I. and OIML, B. (1995). Guide to the expression of uncertainty in measurement. *Geneva, Switzerland*. 122

[123] J Mercer, B. (1909). Xvi. functions of positive and negative type, and their connection the theory of integral equations. *Phil. Trans. R. Soc. Lond. A*, 209(441-458):415–446. 21

[124] Javed, O. and Shah, M. (2008). *Automated multi-camera surveillance: algorithms and practice*, volume 10. Springer Science & Business Media. 4

[125] Jin, Z., An, L., and Bhanu, B. (2017). Group structure preserving pedestrian tracking in a multicamera video network. *IEEE Transactions on Circuits and Systems for Video Technology*. 4

[126] Jiroušek, R. and Shenoy, P. P. (2018). A new definition of entropy of belief functions in the Dempster–Shafer theory. *International Journal of Approximate Reasoning*, 92:49–65. 70, 71

[127] Jones, M. J. and Snow, D. (2008). Pedestrian detection using boosted features over many frames. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE. 4

[128] Jousselme, A.-L., Liu, C., Grenier, D., and Bossé, É. (2006). Measuring ambiguity in the evidence theory. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 36(5):890–903. 71

[129] Junior, J. C. S. J., Musse, S. R., and Jung, C. R. (2010). Crowd analysis using computer vision techniques. *IEEE Signal Processing Magazine*, 27(5):66–77. x

[130] Käding, C., Rodner, E., Freytag, A., and Denzler, J. (2016a). Active and continuous exploration with deep neural networks and expected model output changes. *arXiv preprint arXiv:1612.06129*. 133

[131] Käding, C., Rodner, E., Freytag, A., and Denzler, J. (2016b). Fine-tuning deep neural networks in continuous learning scenarios. In *Asian Conference on Computer Vision*, pages 588–605. Springer. 133

[132] Ke, Y. and Sukthankar, R. (2004). Pca-sift: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE. 36

[133] Kee, S., del Castillo, E., and Runger, G. (2018). Query-by-committee improvement with diversity and density in batch active learning. *Information Sciences*, 454:401–418. 68

[134] Kendall, A., Badrinarayanan, V., and Cipolla, R. (2015). Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*. 104, 105, 109

[135] Kendall, A. and Cipolla, R. (2016). Modelling uncertainty in deep learning for camera relocalization. In *2016 IEEE international conference on Robotics and Automation (ICRA)*, pages 4762–4769. IEEE. 104

[136] Khan, S. D., Saqib, M., and Blumenstein, M. (2017). Towards a dedicated computer vision tool set for crowd simulation models. *arXiv preprint arXiv:1709.02243*. 7

[137] Khan, S. M. and Shah, M. (2009). Tracking multiple occluding people by localizing on multiple scene planes. *IEEE transactions on pattern analysis and machine intelligence*, 31(3):505–519. 4

[138] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 94

[139] Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*. 103

[140] Klaser, A., Marszałek, M., and Schmid, C. (2008). A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, pages 275–1. British Machine Vision Association. 4

[141] Klein, J., Albardan, M., Guedj, B., and Colot, O. (2018). Decentralized learning with budgeted network load using gaussian copulas and classifier ensembles. *arXiv preprint arXiv:1804.10028*. 53

[142] Kratz, L. and Nishino, K. (2012). Tracking pedestrians using local spatio-temporal motion patterns in extremely crowded scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(5):987–1002. 6

[143] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105. 117

[144] Krogh, A. and Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. In *Advances in neural information processing systems*, pages 231–238. 30

[145] Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97. 131

[146] Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86. 68

[147] Kuncheva, L. I. (2000). Clustering-and-selection model for classifier combination. In *Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on*, volume 1, pages 185–188. IEEE. 31

[148] Kuncheva, L. I. (2004). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons. xvii, 26, 29, 33

[149] Kuncheva, L. I. and Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207. 31

[150] Lachaize, M., Le Hégarat-Mascle, S., Aldea, E., Maitrot, A., and Reynaud, R. (2018a). Evidential framework for error correcting output code classification. *Engineering Applications of Artificial Intelligence*, 73:10–21. 54

[151] Lachaize, M., Le Hégarat-Mascle, S., Aldea, E., Maitrot, A., and Reynaud, R. (2018b). Evidential split-and-merge: Application to object-based image analysis. *International Journal of Approximate Reasoning*, 103:303–319. 107, 112

[152] Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413. 104

[153] Lamata, M. T. and Moral, S. (1988). Measures of entropy in the theory of evidence. *International Journal Of General System*, 14(4):297–305. 70, 71

[154] Leibe, B., Seemann, E., and Schiele, B. (2005). Pedestrian detection in crowded scenes. In *null*, pages 878–885. IEEE. 4

[155] Lempitsky, V. and Zisserman, A. (2010). Learning to count objects in images. In *Advances in Neural Information Processing Systems*, pages 1324–1332. 6, 117, 119

[156] Lewis, D. D. and Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the eleventh international conference on machine learning*, pages 148–156. 67

[157] Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc. 67

[158] Li, J., Liang, X., Shen, S., Xu, T., Feng, J., and Yan, S. (2018a). Scale-aware fast r-cnn for pedestrian detection. *IEEE Transactions on Multimedia*, 20(4):985–996. 4, 83

[159] Li, M., Bao, S., Dong, W., Wang, Y., and Su, Z. (2013a). Head-shoulder based gender recognition. In *ICIP*, pages 2753–2756. xvii, 11, 36

[160] Li, M., Zhang, Z., Huang, K., and Tan, T. (2008). Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE. 5, 116

[161] Li, M., Zhang, Z., Huang, K., and Tan, T. (2009). Rapid and robust human detection and tracking based on omega-shape features. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 2545–2548. IEEE. 4, 11

[162] Li, P., Samorodnitsk, G., and Hopcroft, J. (2013b). Sign cauchy projections and chi-square kernel. In *Advances in Neural Information Processing Systems*, pages 2571–2579. 45

[163] Li, T., Chang, H., Wang, M., Ni, B., Hong, R., and Yan, S. (2015). Crowded scene analysis: A survey. *IEEE transactions on circuits and systems for video technology*, 25(3):367–386. x

[164] Li, Y., Zhang, X., and Chen, D. (2018b). Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1091–1100. 6, 118

[165] Lim, M. K., Kok, V. J., Loy, C. C., and Chan, C. S. (2014). Crowd saliency detection via global similarity structure. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 3957–3962. IEEE. 9

[166] Lin, H.-T., Lin, C.-J., and Weng, R. C. (2007). A note on plattâs probabilistic outputs for support vector machines. *Machine learning*, 68(3):267–276. 45

[167] Lin, S.-F., Chen, J.-Y., and Chao, H.-X. (2001). Estimation of number of people in crowded scenes using perspective transformation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 31(6):645–654. 4, 7

[168] Lloyd, K., Rosin, P. L., Marshall, D., and Moore, S. C. (2017). Detecting violent and abnormal crowd activity using temporal analysis of grey level co-occurrence matrix (glcm)-based texture measures. *Machine Vision and Applications*, 28(3-4):361–371. 6

[169] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440. 84, 86, 87, 90

[170] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee. 36

[171] Loy, C. C., Chen, K., Gong, S., and Xiang, T. (2013). Crowd counting and profiling: Methodology and evaluation. In *Modeling, Simulation and Visual Analysis of Crowds*, pages 347–382. Springer. 116

[172] Lu, B.-L. and Ito, M. (1999). Task decomposition and module combination based on class relations: a modular neural network for pattern classification. *IEEE Transactions on Neural Networks*, 10(5):1244–1256. 33

[173] Lughofer, E. (2012). Single-pass active learning with conflict and ignorance. *Evolving Systems*, 3(4):251–271. 71

[174] Ma, B., Su, Y., and Jurie, F. (2012). Local descriptors encoded by fisher vectors for person re-identification. In *European Conference on Computer Vision*, pages 413–422. Springer. 36

[175] Ma, B., Su, Y., and Jurie, F. (2014). Covariance descriptor based on bio-inspired features for person re-identification and face verification. *Image and Vision Computing*, 32(6-7):379–390. 36

[176] Ma, L., Destercke, S., and Wang, Y. (2016). Online active learning of decision trees with evidential data. *Pattern Recognition*, 52:33–45. 29

[177] MacKay, D. J. (1992). A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472. 31, 103

[178] Marana, A. N., Velastin, S. A., Costa, L. d. F., and Lotufo, R. (1998). Automatic estimation of crowd density using texture. *Safety Science*, 28(3):165–175. 117

[179] Marsden, M., McGuinness, K., Little, S., and O'Connor, N. E. (2016). Fully convolutional crowd counting on highly congested scenes. *arXiv preprint arXiv:1612.00220*. 118

[180] McCallumzy, A. K. and Nigamy, K. (1998). Employing EM and pool-based active learning for text classification. In *Proc. International Conference on Machine Learning (ICML)*, pages 359–367. Citeseer. 68

[181] Mehran, R., Oyama, A., and Shah, M. (2009). Abnormal crowd behavior detection using social force model. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 935–942. IEEE. 6, 7

[182] Melville, P., Yang, S. M., Saar-Tsechansky, M., and Mooney, R. (2005). Active learning for probability estimation using Jensen-Shannon divergence. In *European Conference on Machine Learning*, pages 268–279. Springer. 68

[183] Merad, D., Aziz, K. E., and Thome, N. (2010). Fast people counting using head detection from skeleton graph. In *AVSS*, pages 151–156. 4

[184] Mercier, D., Quost, B., and Denoeux, T. (2008). Refined modeling of sensor reliability in the belief function framework using contextual discounting. *Information Fusion*, 9(2):246–258. 51

[185] Micchelli, C. A. and Pontil, M. (2005). Learning the kernel function via regularization. *Journal of machine learning research*, 6(Jul):1099–1125. 21

[186] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–1630. 41, 43

[187] Morel, J.-M. and Yu, G. (2009). Asift: A new framework for fully affine invariant image comparison. *SIAM journal on imaging sciences*, 2(2):438–469. 36

[188] Morerio, P., Marcenaro, L., and Regazzoni, C. S. (2012). People count estimation in small crowds. In *Advanced video and signal-based surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 476–480. IEEE. 117

[189] Moussaïd, M., Perozo, N., Garnier, S., Helbing, D., and Theraulaz, G. (2010). The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PloS one*, 5(4):e10047. 2

[190] Musse, S. R. and Thalmann, D. (1997). A model of human crowd behavior: Group interrelationship and collision detection analysis. In *Computer Animation and Simulationâ97*, pages 39–51. Springer. ix

[191] Neal, R. M. (2012). *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media. 31, 103

[192] Ngai, G. and Yarowsky, D. (2000). Rule writing or annotation: Cost-efficient resource usage for base noun phrase chunking. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 117–125. Association for Computational Linguistics. 68

[193] Nguyen, H. T. (1987). On entropy of random sets and possibility distributions. *The Analysis of Fuzzy Information*, 1:145–156. 70, 71

[194] Nicolas, A., Bouzat, S., and Kuperman, M. N. (2017). Pedestrian flows through a narrow doorway: Effect of individual behaviours on the global flow and microscopic dynamics. *Transportation Research Part B: Methodological*, 99:30–43. 2

[195] Nicolas, A., Kuperman, M., Ibanez, S., Bouzat, S., and Appert-Rolland, C. (2018). Mechanical response of dense pedestrian crowds to the crossing of intruders. *arXiv preprint arXiv:1810.03343*. 2

[196] Ojala, T., Pietikäinen, M., and Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59. 36, 38

[197] Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987. 38, 39

[198] Okuma, K., Taleghani, A., De Freitas, N., Little, J. J., and Lowe, D. G. (2004). A boosted particle filter: Multitarget detection and tracking. In *European conference on computer vision*, pages 28–39. Springer. 5

[199] Onoro-Rubio, D. and López-Sastre, R. J. (2016). Towards perspective-free object counting with deep learning. In *European Conference on Computer Vision*, pages 615–629. Springer. 6, 117

[200] Pal, N. R., Bezdek, J. C., and Hemasinha, R. (1992). Uncertainty measures for evidential reasoning I: A review. *International Journal of Approximate Reasoning*, 7(3-4):165–183. 70, 71

[201] Pal, N. R., Bezdek, J. C., and Hemasinha, R. (1993). Uncertainty measures for evidential reasoning II: A new measure of total uncertainty. *International Journal of Approximate Reasoning*, 8(1):1–16. 70, 71

[202] Paragios, N. and Ramesh, V. (2001). A mrf-based approach for real-time subway monitoring. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE. 116

[203] Pasolli, E., Melgani, F., Tuia, D., Pacifici, F., and Emery, W. J. (2014). SVM active learning approach for image classification using spatial information. *IEEE Transactions on Geoscience and Remote Sensing*, 52(4):2217–2233. 68

[204] Pellegrini, S., Ess, A., Schindler, K., and Van Gool, L. (2009). You'll never walk alone: Modeling social behavior for multi-target tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 261–268. IEEE. 8

[205] Pellicanò, N., Aldea, E., and Hégarat-Mascle, S. L. (2018). Geometry-based multiple camera head detection in dense crowds. *arXiv preprint arXiv:1808.00856*. 4, 131

[206] Peng, P., Tian, Y., Wang, Y., Li, J., and Huang, T. (2015). Robust multiple cameras pedestrian detection with multi-view bayesian network. *Pattern Recognition*, 48(5):1760–1772. 4

[207] Pham, V.-Q., Kozakaya, T., Yamaguchi, O., and Okada, R. (2015). Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3253–3261. 117

[208] Piccoli, B. and Tosin, A. (2011). Time-evolving measures and macroscopic modeling of pedestrian flow. *Archive for Rational Mechanics and Analysis*, 199(3):707–738. 3

[209] Pichon, F., Dubois, D., and Denoeux, T. (2012). Relevance and truthfulness in information correction and fusion. *Int. J. Approx. Reasoning*, 53(2):159–175. 51

[210] Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. *Technical Report MSR-TR-98-14*. 18

[211] Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74. 44

[212] Polikar, R. (2006). Ensemble based systems in decision making. *Circuits and Systems Magazine, IEEE*, 6(3):21–45. 26

[213] Porikli, F. (2005). Integral histogram: A fast way to extract histograms in cartesian spaces. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 829–836. IEEE. 123

[214] Possegger, H., Sternig, S., Mauthner, T., Roth, P. M., and Bischof, H. (2013). Robust real-time tracking of multiple objects by volumetric mass densities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2395–2402. 4

[215] Prodromidis, A. L., Stolfo, S. J., and Chan, P. K. (1999). Effective and efficient pruning of metaclassifiers in a distributed data mining system. *Knowledge Discovery and Data Mining Journal. submitted for publication*. 32

[216] Qi, Z., Ting, R., Husheng, F., and Jinlin, Z. (2012). Particle filter object tracking based on harris-sift feature matching. *Procedia Engineering*, 29:924–929. 5

[217] Qiu, Z., Yao, T., and Mei, T. (2017). Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541. 133

[218] Rabaud, V. and Belongie, S. (2006). Counting crowded moving objects. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 705–711. IEEE. 5, 117

[219] Rahman, A. F. R. and Fairhurst, M. C. (1999). Serial combination of multiple experts: A unified evaluation. *Pattern Analysis & Applications*, 2(4):292–311. 29

[220] Ramirez-Loaiza, M. E., Sharma, M., Kumar, G., and Bilgic, M. (2017). Active learning: an empirical study of common baselines. *Data Mining and Knowledge Discovery*, 31(2):287–313. 75

[221] Reineking, T. (2016). Active classification using belief functions and information gain maximization. *International Journal of Approximate Reasoning*, 72:43–54. 31

[222] Rekik, W., Le Hégarat-Mascle, S., and Aldea, E. (2017). A novel approach for multi-object tracking using evidential representation for objects. In *2017 20th International Conference on Information Fusion (Fusion)*, pages 1–8. IEEE. 132

[223] Rekik, W., Le Hégarat-Mascle, S., André, C., Kallel, A., Reynaud, R., and Hamida, A. B. (2014). Data association for object enumeration using belief function theory. In *International Conference on Belief Functions*, pages 383–392. Springer. 132

[224] Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH computer graphics*, 21(4):25–34. 2

[225] Riesenhuber, M. and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature neuroscience*, 2(11):1019. 36

[226] Rodriguez, M., Laptev, I., Sivic, J., and Audibert, J.-Y. (2011). Density-aware person detection and tracking in crowds. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2423–2430. IEEE. 5, 117, 131

[227] Rokach, L. (2009). Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Computational Statistics & Data Analysis*, 53(12):4046–4072. 27

[228] Rokach, L. and Maimon, O. (2005). Feature set decomposition for decision trees. *Intelligent Data Analysis*, 9(2):131–158. 28, 33

[229] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer. 84, 87, 90

[230] Ryan, D., Denman, S., Fookes, C., and Sridharan, S. (2011). Textures of optical flow for real-time anomaly detection in crowds. In *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, pages 230–235. IEEE. 6

[231] Ryan, D., Denman, S., Sridharan, S., and Fookes, C. (2015). An evaluation of crowd counting methods, features and regression models. *Computer Vision and Image Understanding*, 130:1–17. 116

[232] Sanchez-Matilla, R., Poiesi, F., and Cavallaro, A. (2016). Online multi-target tracking with strong and weak detections. In *European Conference on Computer Vision*, pages 84–99. Springer. 132

[233] Schapire, R. E. (1990). The strength of weak learnability. *Machine learning*, 5(2):197–227. 29

[234] Schohn, G. and Cohn, D. (2000). Less is more: Active learning with support vector machines. In *ICML*, pages 839–846. Citeseer. 67, 119

[235] Schröder, G., Senst, T., Bochinski, E., and Sikora, T. (2018). Optical flow dataset and benchmark for visual crowd analysis. *arXiv preprint arXiv:1811.07170*. 9

[236] Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114. 66, 68, 133

[237] Seung, H. S., Opper, M., and Sompolinsky, H. (1992). Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM. 68

[238] Sexton, J. and Laake, P. (2008). Logitboost with errors-in-variables. *Computational Statistics & Data Analysis*, 52(5):2549–2559. 33

[239] Shafer, G. (1976). *A mathematical theory of evidence*, volume 1. Princeton university press Princeton. 49, 51

[240] Sharkey, A. J. (2002). Types of multinet system. In *International Workshop on Multiple Classifier Systems*, pages 108–117. Springer. 26

[241] SHARKEY, A. J. C. (1996). On combining artificial neural nets. *Connection Science*, 8(3-4):299–314. 33

[242] Sharma, M. and Bilgic, M. (2017). Evidence-based uncertainty sampling for active learning. *Data Mining and Knowledge Discovery*, 31(1):164–202. 72

[243] Shi, J. and Tomasi, C. (1993). Good features to track. Technical report, Cornell University. 5

[244] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. 87, 92

[245] Sindagi, V. A. and Patel, V. M. (2017). Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*, pages 1–6. IEEE. 6, 118

[246] Sindagi, V. A. and Patel, V. M. (2018). A survey of recent advances in cnn-based single image crowd counting and density estimation. *Pattern Recognition Letters*, 107:3–16. 116

[247] Smets, P. and Kennes, R. (1994). The transferable belief model. *Artificial intelligence*, 66(2):191–234. 28, 49, 52

[248] Solmaz, B., Moore, B. E., and Shah, M. (2012). Identifying behaviors in crowd scenes using stability analysis for dynamical systems. *IEEE transactions on pattern analysis and machine intelligence*, 34(10):2064–2070. 6

[249] Solomon, J., De Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., and Guibas, L. (2015). Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (TOG)*, 34(4):66. 132

[250] Song, H., Sun, S., Akhtar, N., Zhang, C., Li, J., and Mian, A. (2018). Benchmark data and method for real-time people counting in cluttered scenes using depth sensors. *arXiv preprint arXiv:1804.04339*. 9

[251] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958. 24, 104

[252] Sulman, N., Sanocki, T., Goldgof, D., and Kasturi, R. (2008). How effective is human video surveillance performance? In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–3. IEEE. x

[253] Sun, T., Jiao, L., Liu, F., Wang, S., and Feng, J. (2013). Selective multiple kernel learning for classification with ensemble strategy. *Pattern Recognition*, 46(11):3081–3090. 31

[254] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*. 104

[255] Tan, X. and Triggs, B. (2007). Fusing gabor and lbp feature sets for kernel-based face recognition. In *International Workshop on Analysis and Modeling of Faces and Gestures*, pages 235–249. Springer. 41

[256] Tao, D., Tang, X., Li, X., and Wu, X. (2006). Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 28(7):1088–1099. 33

[257] Tian, Y., Luo, P., Wang, X., and Tang, X. (2015). Pedestrian detection aided by deep learning semantic tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5079–5087. 4, 83

[258] Tola, E., Lepetit, V., and Fua, P. (2010). Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE TPAMI*, 32(5):815–830. 36, 41

[259] Tong, S. and Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66. 67

[260] Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497. 132

[261] Treuille, A., Cooper, S., and Popović, Z. (2006). Continuum crowds. *ACM Transactions on Graphics (TOG)*, 25(3):1160–1168. 2

[262] Tumer, K. and Ghosh, J. (1996). Error correlation and error reduction in ensemble classifiers. *Connection science*, 8(3-4):385–404. 30

[263] Ujjwal, U., Dziri, A., Leroy, B., and Bremond, F. (2018). Late fusion of multiple convolutional layers for pedestrian detection. In *15th IEEE International Conference on Advanced Video and Signal-based Surveillance*. 4

[264] Valentini, G. and Masulli, F. (2002). Ensembles of learning machines. In *Italian Workshop on Neural Nets*, pages 3–20. Springer. 33

[265] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *CVPR*, pages 511–518. 35

[266] Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., and Fergus, R. (2013). Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*, pages 1058–1066. 104

[267] Wang, C., Zhang, H., Yang, L., Liu, S., and Cao, X. (2015). Deep people counting in extremely dense crowds. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1299–1302. ACM. 117

[268] Wang, K., Zhang, D., Li, Y., Zhang, R., and Lin, L. (2017a). Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600. 133

[269] Wang, S., Zhang, J., and Miao, Z. (2013). A new edge feature for head-shoulder detection. In *ICIP*, pages 2822–2826. 36

[270] Wang, X. (2013). Intelligent multi-camera video surveillance: A review. *Pattern recognition letters*, 34(1):3–19. 36

[271] Wang, X., Fan, B., Chang, S., Wang, Z., Liu, X., Tao, D., and Huang, T. S. (2017b). Greedy batch-based minimum-cost flows for tracking multiple objects. *IEEE Transactions on Image Processing*, 26(10):4765–4776. 5

[272] Wang, X., Han, T. X., and Yan, S. (2009a). An hog-lbp human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 32–39. IEEE. 39

[273] Wang, X., Liu, X., Japkowicz, N., and Matwin, S. (2014). Ensemble of multiple kernel svm classifiers. In *Canadian Conference on Artificial Intelligence*, pages 239–250. Springer. 30

[274] Wang, X., Ma, X., and Grimson, W. E. L. (2009b). Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE Transactions on pattern analysis and machine intelligence*, 31(3):539–555. 5

[275] Warde-Farley, D., Goodfellow, I. J., Courville, A., and Bengio, Y. (2013). An empirical analysis of dropout in piecewise linear networks. *arXiv preprint arXiv:1312.6197*. 24

[276] Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52. 36

[277] Wolpert, D. and Macready, W. G. (1996). Combining stacking with bagging to improve a learning algorithm. *Santa Fe Institute, Technical Report*. 30

[278] Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2):241–259. 30

[279] Wolpert, D. H. (1996). The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390. 26

[280] Woodward, M. and Finn, C. (2017). Active one-shot learning. *arXiv preprint arXiv:1702.06559*. 134

[281] Woźniak, M., Graña, M., and Corchado, E. (2014). A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16:3–17. 26, 29

[282] Wu, J., Cui, Z., Sheng, V. S., Zhao, P., Su, D., and Gong, S. (2013). A comparative study of sift and its variants. *Measurement science review*, 13(3):122–131. 36

[283] Wu, X., Liang, G., Lee, K. K., and Xu, Y. (2006). Crowd density estimation using texture analysis and learning. In *Robotics and Biomimetics, 2006. ROBIO'06. IEEE International Conference on*, pages 214–219. IEEE. 6

[284] Xia, H. and Hoi, S. C. (2013). Mkboost: A framework of multiple kernel boosting. *IEEE Transactions on knowledge and data engineering*, 25(7):1574–1586. 31

[285] Xu, P., Davoine, F., Zha, H., and Denoeux, T. (2016). Evidential calibration of binary SVM classifiers. *International Journal of Approximate Reasoning*, 72:55–70. 54

[286] Xu, Y., Zhou, X., Liu, P., and Xu, H. (2018). Rapid pedestrian detection based on deep omega-shape features with partial occlusion handing. *Neural Processing Letters*, pages 1–15. 83

[287] Xu, Z., Yu, K., Tresp, V., Xu, X., and Wang, J. (2003). Representative sampling for text classification using support vector machines. In *European Conference on Information Retrieval*, pages 393–407. Springer. 66

[288] Yager, R. R. (1983). Entropy and specificity in a mathematical theory of evidence. *International Journal of General System*, 9(4):249–260. 70, 71

[289] Yager, R. R. (1987). On the dempster-shafer framework and new combination rules. *Information sciences*, 41(2):93–137. 112

[290] Yu, F. and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*. xviii, 85

[291] Zadeh, L. A. (1999). Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 100(1):9–34. 28, 49

[292] Zadeh, L. A. et al. (1965). Fuzzy sets. *Information and control*, 8(3):338–353. 28, 49

[293] Zhang, C., Kang, K., Li, H., Wang, X., Xie, R., and Yang, X. (2016a). Data-driven crowd understanding: a baseline for a large-scale crowd dataset. *IEEE Transactions on Multimedia*, 18(6):1048–1061. 9

[294] Zhang, L., Lin, L., Liang, X., and He, K. (2016b). Is faster r-cnn doing well for pedestrian detection? In *European Conference on Computer Vision*, pages 443–457. Springer. 4, 83

[295] Zhang, S., Benenson, R., Omran, M., Hosang, J., and Schiele, B. (2016c). How far are we from solving pedestrian detection? In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 1259–1267. IEEE. 4, 83

[296] Zhang, X., Weng, W., Yuan, H., and Chen, J. (2013). Empirical study of a unidirectional dense crowd during a real mass event. *Physica A: Statistical Mechanics and its Applications*, 392(12):2781–2791. 8

[297] Zhang, Y. and Li, S. (2011). Gabor-lbp based region covariance descriptor for person re-identification. In *Image and Graphics (ICIG), 2011 Sixth International Conference on*, pages 368–371. IEEE. 41

[298] Zhang, Y., Zhou, D., Chen, S., Gao, S., and Ma, Y. (2016d). Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 589–597. 6, 9, 117, 118

[299] Zhao, R., Ouyang, W., and Wang, X. (2013). Person re-identification by salience matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2528–2535. 36

[300] Zhao, R., Ouyang, W., and Wang, X. (2014). Learning mid-level filters for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 144–151. 36

[301] Zhao, T. and Nevatia, R. (2003). Bayesian human segmentation in crowded situations. In *null*, page 459. IEEE. 4

[302] Zhao, T. and Nevatia, R. (2004). Tracking multiple humans in crowded environment. In *null*, pages 406–413. IEEE. 4

[303] Zhao, Y., Xu, C., and Cao, Y. (2006). Research on query-by-committee method of active learning and application. In *International Conference on Advanced Data Mining and Applications*, pages 985–991. Springer. 68

[304] Zhou, B., Wang, X., and Tang, X. (2012). Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2871–2878. IEEE. 8

Appendix A

Ground-truth labeling

The Makkah dataset used in this project is composed by sequences of gray-scale images acquired by multiple cameras at Makkah during very congested times of the Hajj period, in October 2012. The cameras used for the recordings are robotic cameras (AVT Guppy PRO) mounted statically in different places of the Great Mosque's central square in order to observe the high-density pilgrim crowd from different perspectives. Each camera recorded video sequences of the crowd (at a frame-rate of 8Hz), providing gray-level regular image frames of the visible spectrum.

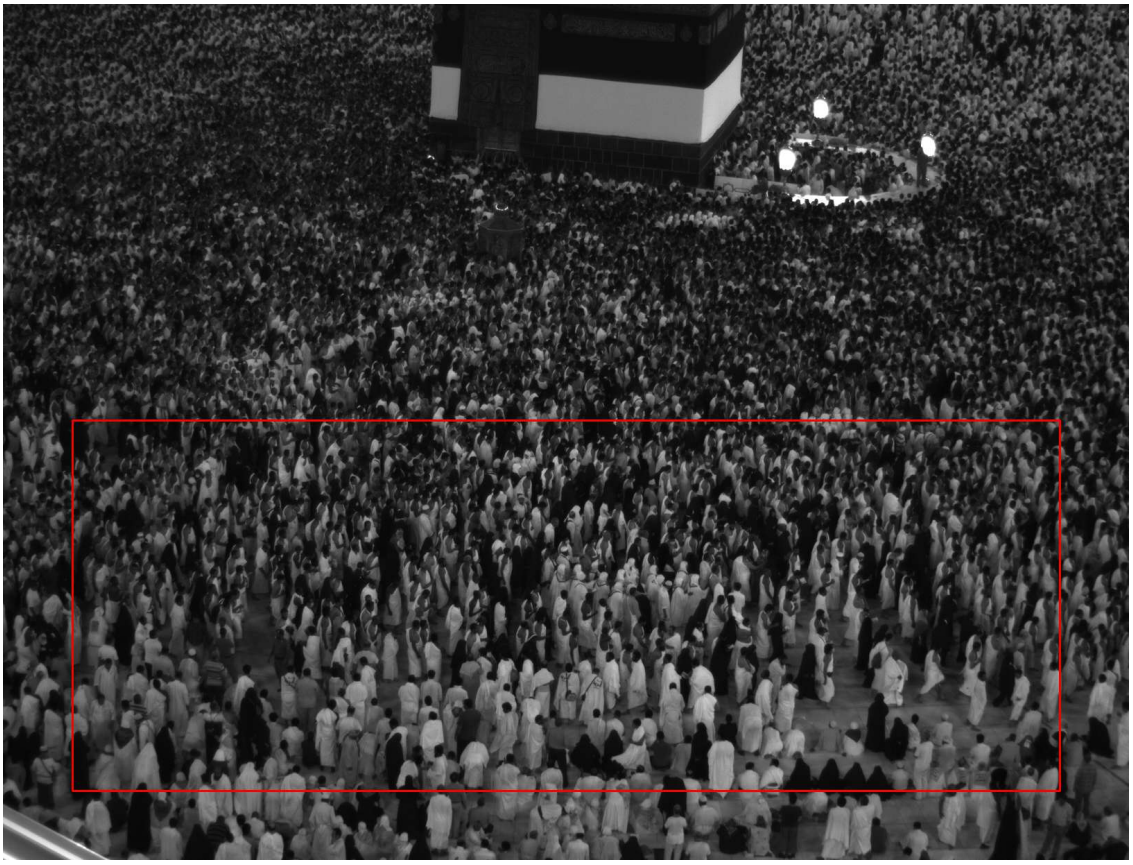


Figure A.1: Example image from the Makkah dataset. The region of interest considered in this work is highlighted in red.

Figure A.1 shows an image from the dataset considered in the context of this mono-camera based work. The highlighted region of interest allows us to perform both a microscopic and a macroscopic analysis, and roughly contains 900-1000 heads per image.

In order to obtain the ground-truth information which is useful both to build the training set and to perform evaluation on the validation and testing sets, annotations are needed. Dot annotations have been manually performed through a web interface especially designed for the task,

where the user is asked to choose the image to label and then to discriminate between the various types of heads.

In anticipation of the possible extension of the work to multi-class analysis indeed, different classes of pedestrians have been identified specifically for this dataset, i.e. women with white veil, women with black veil, men with hat, men without hat. Figure A.2 shows an example of labeling, where the user has clicked in the center of each head to perform the annotations.

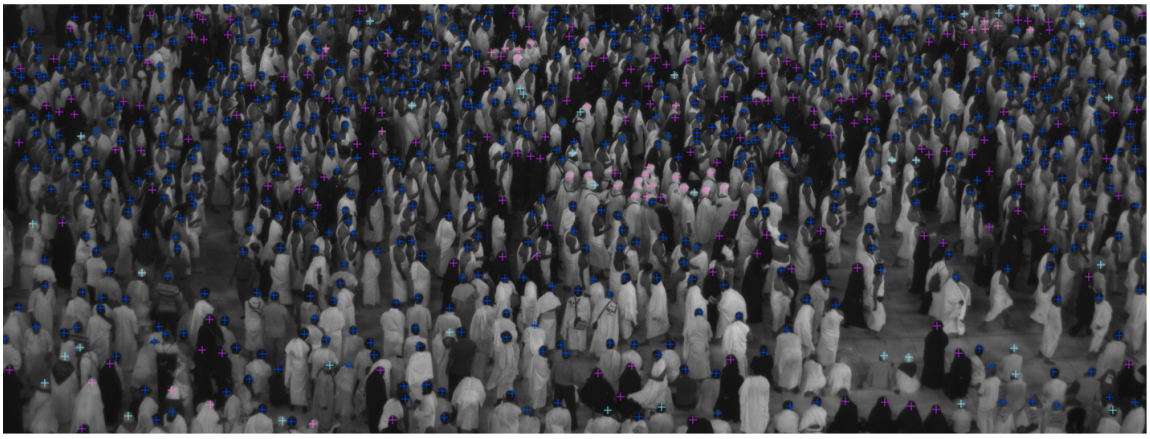


Figure A.2: Example of ground-truth labeling.

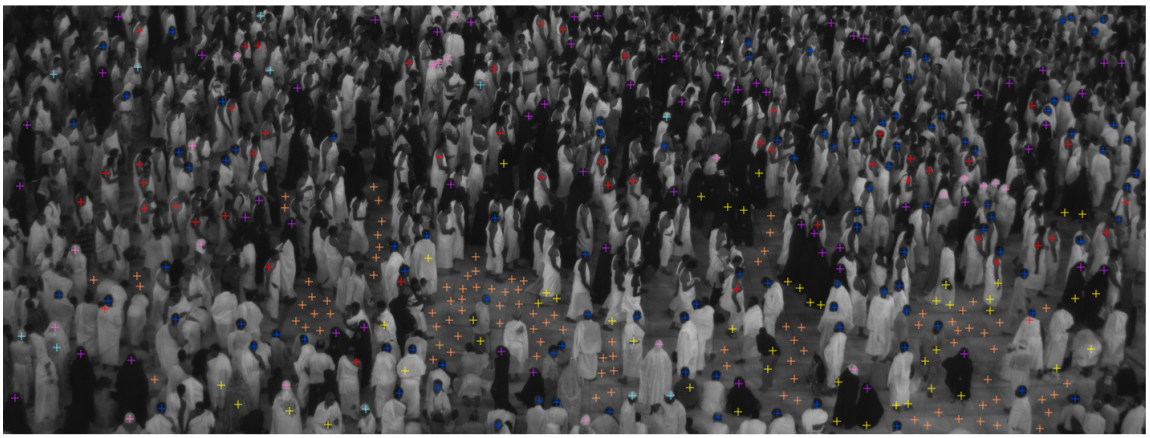


Figure A.3: Example of positive and negative sample labeling to manually select samples to add to the training set.

The interface can be used both to perform ground-truth labeling for an entire image, as shown in Fig. A.2, and also to manually build a training set with positive and negative samples. To this extent, Fig. A.3 shows an example of an image where the user clicked on some specific locations both to obtain positive and negative samples to be added to a training set. Different classes of negative samples have been identified as well, i.e. floor, clothes and shoulders (which are particularly difficult for their peculiar shape which is similar to a head in some cases).

Note that to maintain the method's generalization ability, in the context of this work we considered the generic binary problem *head* vs. *not head*. Having defined the different classes allows us nevertheless to easily obtain diverse training sets, where the appearance of both positive and negative samples is able to span over the different types. Note that for the proposed active learning procedure only ground-truth maps are needed, decreasing the burden of the annotation that has to be performed by the user only for the positive class.

The web interface allows us finally to download the annotations, which are saved as lists of triplets "*label x y*" for each image, where *label* is an integer corresponding to the sample's class, while x and y corresponds to the sample's coordinates (head's center).

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Méthodes d'ensembles pour la détection de piétons en foules denses
Mots clés: Analyse de foules, fusion d'informations, théorie des fonctions de croyance, traitement d'images, apprentissage

Résumé: Cette thèse s'intéresse à la détection des piétons dans des foules très denses depuis un système mono-camera, avec comme but d'obtenir des détections localisées de toutes les personnes. Ces détections peuvent être utilisées soit pour obtenir une estimation robuste de la densité, soit pour initialiser un algorithme de suivi. Les méthodologies classiques utilisées pour la détection de piétons s'adaptent mal au cas où seulement les têtes sont visibles, de part l'absence d'arrière-plan, l'homogénéité visuelle de la foule, la petite taille des objets et la présence d'occultations très fortes. En présence de problèmes difficiles tels que notre application, les approches à base d'apprentissage supervisé sont bien adaptées. Nous considérons un système à plusieurs classifieurs (Multiple Classifier System, MCS), composé de deux ensembles différents, le premier basé sur les classifieurs SVM (SVM-ensemble) et le deuxième basé sur les CNN (CNN-ensemble), combinés dans le cadre de la Théorie des Fonctions de Croyance (TFC). L'ensemble SVM est composé de plusieurs SVM exploitant les données issues d'un descripteur différent. La TFC nous permet de prendre en compte une valeur d'imprécision supposée correspondre soit à une imprécision dans la procédure de calibration, soit à une imprécision spatiale. Cependant, le manque de données labellisées pour le cas des foules très denses nuit à la génération d'ensembles de données d'entrainement et de validation robustes. Nous avons proposé un algorithme d'apprentissage actif de type Query-by-Committee (QBC) qui permet de sélectionner automatiquement de nouveaux échantillons d'apprentissage. Cet algorithme s'appuie sur des mesures évidentielles déduites des fonctions de croyance. Pour le second ensemble, pour exploiter les avancées de l'apprentissage profond, nous avons reformulé notre problème comme une tâche de segmentation en soft labels. Une architecture entièrement convolutionelle a été conçue pour détecter les petits objets grâce à des convolutions dilatées. Nous nous sommes appuyés sur la technique du dropout pour obtenir un ensemble CNN capable d'évaluer la fiabilité sur les prédictions du réseau lors de l'inférence. Les réalisations de cet ensemble sont ensuite combinées dans le cadre de la TFC. Pour conclure, nous montrons que la sortie du MCS peut être utile aussi pour le comptage de personnes. Nous avons proposé une méthodologie d'évaluation multi-échelle, très utile pour la communauté de modélisation car elle lie incertitude (probabilité d'erreur) et imprécision sur les valeurs de densité estimées.

Title: Ensemble Methods for Pedestrian Detection in Dense Crowds.
Key Words: crowd analysis, information fusion, belief function theory, image processing, machine learning

Summary: This study deals with pedestrian detection in high-density crowds from a mono-camera system. The detections can be then used both to obtain robust density estimation, and to initialize a tracking algorithm. One of the most difficult challenges is that usual pedestrian detection methodologies do not scale well to high-density crowds, for reasons such as absence of background, high visual homogeneity, small size of the objects, and heavy occlusions. We cast the detection problem as a Multiple Classifier System (MCS), composed by two different ensembles of classifiers, the first one based on SVM (SVM-ensemble) and the second one based on CNN (CNN-ensemble), combined relying on the Belief Function Theory (BFT) to exploit their strengths for pixel-wise classification. SVM-ensemble is composed by several SVM detectors based on different gradient, texture and orientation descriptors, able to tackle the problem from different perspectives. BFT allows us to take into account the imprecision in addition to the uncertainty value provided by each classifier, which we consider coming from possible errors in the calibration procedure and from pixel neighbor's heterogeneity in the image space. However, scarcity of labeled data for specific dense crowd contexts reflects in the impossibility to obtain robust training and validation sets. By exploiting belief functions directly derived from the classifiers' combination, we propose an evidential Query-by-Committee (QBC) active learning algorithm to automatically select the most informative training samples. On the other side, we explore deep learning techniques by casting the problem as a segmentation task with soft labels, with a fully convolutional network designed to recover small objects thanks to a tailored use of dilated convolutions. In order to obtain a pixel-wise measure of reliability about the network's predictions, we create a CNN-ensemble by means of dropout at inference time, and we combine the different obtained realizations in the context of BFT. Finally, we show that the output map given by the MCS can be employed to perform people counting. We propose an evaluation method that can be applied at every scale, providing also uncertainty bounds on the estimated density.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France