UNIVERSITÉ D'ARTOIS

Evidential calibration and fusion of multiple classifiers: application to face blurring

Calibration et fusion évidentielles de classifieurs: application à l'anonymisation de visages

THÈSE

Présentée et soutenue publiquement le 8 décembre 2017
en vue de l'obtention du

Doctorat de l'Université d'Artois

Spécialité : Génie Informatique et Automatique

par

Pauline Minary

Composition du jury :

M.	Thierry Denœux	Professeur, Université de Technologie de Compiègne	Rapporteur
Mme.	Michèle Rombaut	Professeur, Université Grenoble Alpes	Rapporteur
M.	Olivier Colot	Professeur, Université de Lille 1	Examinateur
Mme.	Sylvie Le Hégarat Mascle	Professeur, Université de Paris Sud	Examinatrice
M.	Franck Luthon	Professeur, Université de Pau et des Pays de l'Adour	Examinateur
M.	Éric Lefèvre	Professeur, Université d'Artois	Directeur
M.	David Mercier	Maître de Conférences (HDR), Université d'Artois	Co-directeur
M.	Frédéric Pichon	Maître de Conférences, Université d'Artois	Co-Encadrant
M.	Benjamin Droit	Chef de projet, SNCF	Invité

Thèse préparée au
Laboratoire de Génie Informatique et d'Automatique de l'Artois
Université d'Artois
Faculté des sciences appliquées
Technoparc Futura
62 400 Béthune

et à
SNCF Réseau
Département des Télécommunications
6 avenue Francois Mitterrand
93 574 La Plaine Saint Denis

Abstract

In order to improve overall performance of a classification problem, a path of research consists in using several classifiers and to fuse their outputs. To perform this fusion, some approaches merge the outputs using a fusion rule. This requires that the outputs be made comparable beforehand, which is usually done using a probabilistic calibration of each classifier. The fusion can also be performed by concatenating the classifier outputs into a vector, and applying a joint probabilistic calibration to it. Recently, extensions of probabilistic calibrations of an individual classifier have been proposed using evidence theory, in order to better represent the uncertainties inherent to the calibration process. In the first part of this thesis, this latter idea is adapted to joint probabilistic calibration techniques, leading to evidential versions. This approach is then compared to the aforementioned ones on classical classification datasets. In the second part, the challenging problem of blurring faces on images, which SNCF needs to address, is tackled. A state-of-the-art method for this problem is to use several face detectors, which return boxes with associated confidence scores, and to combine their outputs using an association step and an evidential calibration. In this report, it is shown that reasoning at the pixel level is more interesting than reasoning at the box-level, and that among the fusion approaches discussed in the first part, the evidential joint calibration yields the best results. Finally, the case of images coming from videos is considered. To leverage the information contained in videos, a classical tracking algorithm is added to the blurring system.

Keywords : Calibration, Face detection, Theory of belief functions, Classification, Information fusion, Logistic regression.

Résumé

Afin d'améliorer les performances d'un problème de classification, une piste de recherche consiste à utiliser plusieurs classifieurs et à fusionner leurs sorties. Pour ce faire, certaines approches utilisent une règle de fusion. Cela nécessite que les sorties soient d'abord rendues comparables, ce qui est généralement effectué en utilisant une calibration probabiliste de chaque classifieur. La fusion peut également être réalisée en concaténant les sorties et en appliquant à ce vecteur une calibration probabiliste conjointe. Récemment, des extensions des calibrations d'un classifieur individuel ont été proposées en utilisant la théorie de l'évidence, afin de mieux représenter les incertitudes. Premièrement, cette idée est adaptée aux techniques de calibrations probabilistes conjointes, conduisant à des versions évidentielles. Cette approche est comparée à celles mentionnées ci-dessus sur des jeux de données de classification classiques. Dans la seconde partie, le problème d'anonymisation de visages sur des images, auquel SNCF doit répondre, est considéré. Une méthode consiste à utiliser plusieurs détecteurs de visages, qui retournent des boites et des scores de confiance associés, et à combiner ces sorties avec une étape d'association et de calibration évidentielle. Il est montré que le raisonnement au niveau pixel est plus intéressant que celui au niveau boite et que, parmi les approches de fusion abordées dans la première partie, la calibration conjointe évidentielle donne les meilleurs résultats. Enfin, le cas des images provenant de vidéos est considéré. Pour tirer parti de l'information contenue dans les vidéos, un algorithme de suivi classique est ajouté au système.

Mots-clés : Calibration, Détection de visages, Théorie des fonctions de croyance, Classification, Fusion d'informations, Régression logistique.

Remerciements

En premier lieu, je voudrais exprimer ma sincère gratitude et mes profonds remerciements à mon directeur Eric Lefevre, pour son soutien continu, sa grande compétence et sa gentillesse. Je suis également très reconnaissante à mon co-directeur David Mercier pour sa motivation, sa disponibilité et ses conseils, ainsi qu'à mon co-encadrant Frédéric Pichon, dont l'enthousiasme pour le sujet traité, mais aussi pour la recherche en général, a été contagieux et très motivant. Je les remercie tous les trois pour tout le temps qu'ils m'ont consacré ainsi que pour tous les précieux conseils et pertinentes remarques qu'ils ont formulés au cours de ces 3 années. Venant d'une école d'ingénieurs, le monde de la recherche était relativement nouveau pour moi et ils m'ont sans nul doute aidé à devenir une meilleure et plus rigoureuse chercheuse. Je n'aurais pas pu imaginer meilleur trio pour mon doctorat.

Mes remerciements vont également à mon encadrant SNCF, Benjamin Droit, pour la confiance qu'il m'a accordée, sa disponibilité, ainsi que l'aide prodiguée tout au long de mon doctorat.

J'aimerais également remercier l'ensemble de mes collègues de la section ES de la SNCF pour leur sympathie et leur bonne humeur, avec une mention toute particulière à Jean-Christophe et à Louis.

De même, j'ai beaucoup apprécié la camaraderie de tous les membres du laboratoire LGI2A et je les remercie tous pour leur accueil très chaleureux.

Je voudrais en outre remercier les rapporteurs de cette thèse, M. Thierry Denœux et Mme Michèle Rombaut, pour l'intérêt qu'ils ont porté à mon travail ainsi que pour toutes leurs remarques et suggestions. J'associe à ces remerciements M. Olivier Colot, Mme Sylvie Le Hégarat-Mascle et M. Franck Luthon, pour avoir accepté d'examiner mon travail.

A titre plus personnel, je voudrais aussi remercier mes amis pour tous les bons moments passés ensemble, à Paris ou ailleurs. Une pensée particulière à mes amis "Enseirbiens" (Braco, Grégo, Max, Les Bos, Les Pons...), mais aussi à Elise, Mélanie, ainsi qu'à mon colocataire Hugues.

De plus, je désire grandement exprimer mes profonds remerciements à mes

parents Maryse et Jean-Pierre et à mes soeurs Manon et Emma pour leur amour, leur soutien et la confiance qu'ils me permettent d'acquérir, que ce soit dans le cadre de ma thèse mais aussi plus globalement dans mon parcours de vie.

Enfin, un merci tout spécial à mon compagnon Louis, dont les encouragements, la patience sans faille et le soutien fidèle au cours de ce doctorat ont représenté une ressource essentielle pour mener à terme le travail engagé.

Contents

List of Tables

List of Figures

Introduction

Using multiple classifiers in order to solve a classification problem is a widely studied subject in the supervised learning field, as it may be significantly more accurate than using a single classifier. Ensemble methods regroup all the approaches that are based on multiple classifiers [68, 107, 126]. Among them, the most common types are the techniques of bagging [16], boosting [93, 45], and combination methods. In bagging (bootstrap aggregation), multiple models are created using the same learning algorithm but trained with different subsets of the original training dataset, randomly created with bootstrap sampling method. Boosting refers to methods, which are able to convert weak models, *i.e.*, which have an error classification rate slightly below random guess, to strong models. They iteratively build an ensemble by training each model with the same dataset but where the weights of samples are adjusted according to the error of the last iteration. The main idea is to force the models to focus on the difficult samples. The third category concerns the methods that train different classifiers, called base classifiers, and use their outputs as input to another classifier, called combiner classifier. This category is the one that interested us the most as it enables for instance to use for combination the outputs returned by already pre-trained base classifiers.

The approaches based on a fusion of multiple classifiers have many applications in classification, especially in the image processing field [17, 40, 87, 75, 114]. SNCF (*Société Nationale des Chemins de Fer*), the French railway company, needs to address an issue of this kind due to legal reasons. Safety is one of the most important challenges for SNCF. With such an infrastructure, there are multiple hazards for the passengers and the agents. For instance, it may happen that the train starts while a passenger is stuck between the train and the platform. Within this scope, a process called EAS (*Exploitation à Agent Seul*) has been developed since the 1980's in around 400 suburb stations, in Ile-de-France. The purpose of this system is to allow the train driver to watch the railway platform in its entirety, so that the train can be started without any problem. This system is composed of cameras positioned on the platforms, and monitors. For the purpose of checking the proper positioning of cameras, a series of videos is regularly recorded. However, according to the French legislation about respect for private life, a video shall not be retained by the company if it contains identifiable people on it. As it is too expensive to bring a train on purpose during off hours, the videos are recorded during commercial hours, and thus, with public (users).

SNCF would like to keep these videos for different purposes, such as for preventive maintenance. Thus, the only viable solution is to make people faces unrecognizable on those videos, *i.e.*, blurring them, by using an automatic, or at least semi-automatic system, as it is clearly too tedious to blur by hand the faces on each image of a video. Yet, the conditions of the application are challenging, as they present several difficulties such as bad image quality, indoor and outdoor situations, variation of lighting, etc. Furthermore, faces are deformable objects, they can have many poses and sizes, and occlusions may occur especially when the platform is crowded. Figure 1 shows two examples of images extracted of some videos, and as it can be noticed the lighting and the scene disposition are very different. These challenging conditions impact the efficiency of a given face detector, which can be used to obtain face positions in a image; given these positions, the blurring can be performed. Thus, using several classifiers in order to obtain different information and combining these outputs seems to be an interesting path of research to solve the face blurring issue.

Figure 1 – Examples of images extracted from two different videos.

This report is composed of two main parts. The first part concerns the combiner classifier, which fuses the outputs returned by several classifiers. To perform this fusion, some approaches merge the outputs using a fusion rule. This requires that the outputs be made comparable beforehand, which is usually done using a probabilistic calibration of each classifier. The fusion can also be performed by concatenating the classifier outputs into a vector, and applying a joint probabilistic calibration to this vector. Recently, extensions of probabilistic calibration techniques of an individual classifier have been proposed using evidence theory, in order to better represent the uncertainties inherent to the calibration process. In this part of this thesis, this latter idea is adapted to probabilistic joint calibration techniques, leading to evidential versions of joint calibration techniques. Chapter 1 exposes the main concepts of the belief function theory and in particular its application to statistical inference and forecasting, which is necessary to extend calibration techniques to the evidential framework. The probabilistic version of the calibration techniques of a single classifier are exposed in

Chapter 2, followed by their extension to the evidential framework. Then, the probabilistic and evidential approaches of joint calibration of multiple classifiers that we propose are described. Some performed experiments on classical datasets are exposed.

The second main part of this report deals with the problem of face blurring in images, which SNCF needs to address due to legal reasons. A state-of-the-art method for this problem is to use several face detectors, which return bounding boxes with associated confidence scores, and to combine their outputs using an association step and an evidential calibration of the detector scores. This classical approach is based on the box-level. In this report, a reasoning at pixel-level, *i.e.*, viewing this problem as a binary classification of the pixels where each detector classifies each pixel as belonging or not to a face, is proposed. Furthermore, the application of the joint calibration is applied to this face blurring issue. Chapter 3 describes the classical box-based approach as well as the pixel-based approach that we propose. A comparison between these approaches is given, in terms of concepts and performances. Then, the case of images extracted from videos is considered in Chapter 4, where a tracking algorithm is integrated to the blurring system in order to leverage the information contained in videos with respect to the face blurring problem. This report ends with a general conclusion and some directions for future work.

Part I

Evidential joint calibration

Chapter 1

Belief function theory

Contents

1.1 Introduction

There are uncertainties in many systems and applications [2]. These uncertainties may have different origins, for instance they may come from unreliable sources. Based on the previous work of Dempster [23], Shafer established the basis of a theory [97], called the Dempster-Shafer theory, which was further popularized and developed in particular by Smets [100]. This theory, also known as the belief function theory or evidence theory, has proved to be an effective theoretical framework for reasoning with uncertain information. Indeed, this theory offers a convenient formalism to represent, merge and propagate uncertainty. Starting from the 1990's, it has been applied in a growing number of applications and in many different fields, such as in

data classification [86, 27], data clustering [31, 30], information fusion [10, 69, 114], computer vision [41, 87], etc.

In this chapter, we first expose in Section 1.2 how to represent information with belief functions. In Section 1.3, the most commonly used combination rule of this framework is presented. The issue of decision-making using belief functions is described in Section 1.4. Belief function theory can also be used for statistical inference and prediction, as detailed in Section 1.5. This latter part is useful to extend probabilistic calibrations to the evidential framework, as it will be seen in Chapter 2.

1.2 Information representation

This section exposes how to represent knowledge under the form of a belief function, *i.e.*, a function which allows one to take into account the imprecision and uncertainty that might be contained in a piece of information.

1.2.1 Mass function

Let $\boldsymbol{\omega}$ be a variable whose possible values belong to the finite set $\Omega = \{\omega_1, \cdots, \omega_K\}$. In the belief function theory, uncertainty with respect to the actual value ω_0 taken by $\boldsymbol{\omega}$ is represented using a *Mass Function* (MF) defined as a mapping $m^\Omega : 2^\Omega \to [0, 1]$ verifying $m^\Omega(\emptyset) = 0$ and

$$\sum_{A \subseteq \Omega} m^\Omega(A) = 1. \tag{1.1}$$

The quantity $m^\Omega(A)$ corresponds to the part of belief committed exactly to the hypothesis $\omega_0 \in A$ and nothing more specific.

Definition 1.2.1 *Any subset A of Ω such that $m^\Omega(A) > 0$ is called a focal set of m^Ω.*

Definition 1.2.2 *A mass function is said to be vacuous if Ω is the only focal set,* i.e., *$m^\Omega(\Omega) = 1$. As $m^\Omega(\Omega)$ represents the degree of ignorance, a vacuous mass function represents the case of total ignorance. We denote by m_Ω the corresponding mass function.*

Definition 1.2.3 *When the focal sets $A_1, \ldots, A_N$, of a mass function m^Ω are nested,* i.e., *$A_1 \subseteq A_2 \subseteq \cdots \subseteq A_N$, the mass function m^Ω is said to be consonant.*

Definition 1.2.4 *A mass function is said to be Bayesian if its focal sets are singletons,* i.e.*, when any subset A of Ω such that $m^\Omega(A) > 0$ implies $|A| = 1$.*

Example 1.2.1 *A train driver saw in the distance a silhouette in the vicinity of railways. There are three possibilities: it is either a working SNCF employee (e), or a senseless person who should not be there (p), or an animal (a). The set $\Omega = \{e, p, a\}$ can be chosen as the frame of discernment. The train driver only saw the silhouette briefly but he is convinced at 70% that he saw a human face. This information is not fully certain as the driver did not see well, but if he is correct, we know that it was either an employee or an unreasonable person. Otherwise, we know nothing, as it might be a human or an animal. This piece of evidence can be represented by the following mass function:*

$$m_1^\Omega(\{e, p\}) = 0.7, \quad m_1^\Omega(\Omega) = 0.3. \tag{1.2}$$

For instance, the quantity $m_1^\Omega(\{e, p\})$ corresponds to the share of belief committed exactly to the hypothesis $\omega_0 \in \{e, p\}$.

1.2.2 Belief and plausibility functions

The belief and the plausibility functions are equivalent representations of a mass function. They are respectively defined by

$$Bel^\Omega(A) = \sum_{B \subseteq A} m^\Omega(B), \quad \forall A \subseteq \Omega, \tag{1.3}$$

$$Pl^\Omega(A) = \sum_{B \cap A \neq \emptyset} m^\Omega(B), \quad \forall A \subseteq \Omega. \tag{1.4}$$

The degree of belief $Bel^\Omega(A)$ measures the amount of evidence strictly in favour of the hypothesis $\omega_0 \in A$, while the plausibility $Pl^\Omega(A)$ is the amount of evidence not contradicting it.

Property 1.2.1 $Bel^\Omega(A) \leq Pl^\Omega(A), \quad \forall A \subseteq \Omega$.

Property 1.2.2 *These two quantities are linked by*

$$Pl^\Omega(A) = 1 - Bel^\Omega(\overline{A}), \quad \forall A \subseteq \Omega, \tag{1.5}$$

$$Bel^\Omega(A) = 1 - Pl^\Omega(\overline{A}), \quad \forall A \subseteq \Omega, \tag{1.6}$$

where $\overline{A}$ is the complement of A.

Property 1.2.3 *A mass function m^Ω can be retrieved from Bel^Ω or Pl^Ω using the following equations:*

$$m^\Omega(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|} Bel^\Omega(B), \quad \forall A \subseteq \Omega, \tag{1.7}$$

$$m^\Omega(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|+1} Pl^\Omega(\overline{B}), \quad \forall A \subseteq \Omega. \tag{1.8}$$

Property 1.2.4 *If the mass function m^Ω is Bayesian, then $Bel^\Omega(A) = Pl^\Omega(A) \quad \forall A \subseteq \Omega$, and Bel^Ω and Pl^Ω are a probability measure.*

Property 1.2.5 *The plausibility function restricted to singletons is called the contour function, denoted pl^Ω and defined by*

$$pl^\Omega(\omega) = Pl^\Omega(\{\omega\}), \quad \forall \omega \in \Omega. \tag{1.9}$$

Property 1.2.6 *When a mass function is consonant, the plausibility function can be recovered from its contour function as follows:*

$$Pl^\Omega(A) = \sup_{\omega \in A} pl^\Omega(\omega), \quad \forall A \subseteq \Omega. \tag{1.10}$$

Example 1.2.2 *Table 1.1 gives the mass function m_1^Ω of Example 1.2.1, as well as the belief function Bel_1^Ω and plausibility function Pl_1^Ω associated to m_1^Ω.*

A	$\{e\}$	$\{p\}$	$\{e,p\}$	$\{a\}$	$\{e,a\}$	$\{p,a\}$	Ω
$m_1^\Omega(A)$	0	0	0.7	0	0	0	0.3
$Bel_1^\Omega(A)$	0	0	0.7	0	0	0	1
$Pl_1^\Omega(A)$	1	1	1	0.3	1	1	1

Table 1.1 – Mass, belief and plausibility functions for Example 1.2.1.

If we take the hypothesis "it was an Animal", we have no information that is strictly in favour of this hypothesis, so $Bel^\Omega(\{a\}) = 0$. Furthermore, the evidence that does not contradict this hypothesis has a confidence of 0.3, so $Pl^\Omega(\{a\}) = 0.3$.

1.3 Combination of evidence

An important aspect of belief function theory concerns the combination of some pieces of evidence provided by different sources. Several combination rules exist to merge two given mass functions, such as the disjunctive rule [99] or the cautious

rule [28]. In this section, we will only introduce the most commonly used one, which is Dempster's rule of combination [23, 97], also known as orthogonal sum.

Given two mass functions m_1^Ω and m_2^Ω obtained from two distinct and reliable sources, the fusion of these two pieces of evidence with Dempster's rule of combination, denoted $\oplus$, results in a mass function $m_{1\oplus 2}^\Omega$ defined by

$$m_{1\oplus 2}^\Omega(A) = (m_1^\Omega \oplus m_2^\Omega)(A) = \frac{1}{1-\kappa} \sum_{B\cap C=A} m_1^\Omega(B) m_2^\Omega(C), \qquad \forall A \neq \emptyset, \tag{1.11}$$

where

$$\kappa = \sum_{B\cap C=\emptyset} m_1^\Omega(B) m_2^\Omega(C), \tag{1.12}$$

represents the degree of conflict between m_1^Ω and m_2^Ω, and $m_{1\oplus 2}^\Omega(\emptyset) = 0$. If $\kappa = 1$, there is a total conflict between the two pieces of evidence and they cannot be combined.

Property 1.3.1 *Dempster's rule of combination is commutative,* i.e., $m_{1\oplus 2}^\Omega = m_{2\oplus 1}^\Omega$.

Property 1.3.2 *Dempster's rule of combination is associative,* i.e., $m_{1\oplus 2}^\Omega \oplus m_3^\Omega = m_1^\Omega \oplus m_{2\oplus 3}^\Omega$.

Property 1.3.3 *The vacuous mass function m_Ω is the unique neutral element,* i.e., $m^\Omega \oplus m_\Omega = m^\Omega$.

Example 1.3.1 *A new piece of evidence is added to Example 1.2.1: the ticket inspector affirms that if they are in a protected natural area, it was necessarily an animal. As they were not in a wooded environment, we judge that there is only* 30% *of chance that it was a protected natural area. This information can be represented by the following mass function:*

$$m_2^\Omega(\{a\}) = 0.3, \quad m_2^\Omega(\Omega) = 0.7. \tag{1.13}$$

The two masses m_1^Ω and m_2^Ω are obtained from two different sources that we assume to be reliable. Thus, they can be combined following Eq. (1.11) *and this fusion results in the mass function $m_{1\oplus 2}^\Omega$, presented in Table 1.2.*

A	$\{e\}$	$\{p\}$	$\{e,p\}$	$\{a\}$	$\{e,a\}$	$\{p,a\}$	Ω
$m_{1\oplus 2}^\Omega(A)$	0	0	0.62	0.11	0	0	0.27

Table 1.2 – Mass function $m_{1\oplus 2}^\Omega$ resulting from the combination of m_1^Ω and m_2^Ω.

1.4 Decision-making

After representing and merging imperfect information about a given problem, one may need to make a decision about the considered problem. Different strategies exist in the evidential formalism to make a decision about the actual value ω_0 of $\boldsymbol{\omega}$ given knowledge about ω_0 represented by a mass function m^Ω [26]. This section exposes some of them, in particular the decision strategies based on maximum of belief (or plausibility), based on costs and finally, using a reject option.

A first simple strategy consists in choosing the value $\omega \in \Omega$ corresponding to the singleton with the highest belief. The same strategy can be used choosing the singleton with maximum plausibility. In a binary situation, *i.e.*, when $|\Omega| = 2$, using the maximum of mass, belief or plausibility leads to the same decision.

Yet, making a wrong decision about a class can have more or less important consequence depending on the considered class, and this can be taken into account using decision costs. The value $\omega \in \Omega$ having the smallest so-called *upper* or *lower expected costs* may be selected. The upper and lower expected costs of some value $\omega \in \Omega$, respectively denoted by $R^*(\omega)$ and $R_*(\omega)$, are defined as

$$R^*(\omega) = \sum_{A \subseteq \Omega} m^\Omega(A) \max_{\omega' \in A} c(\omega, \omega'), \tag{1.14}$$

$$R_*(\omega) = \sum_{A \subseteq \Omega} m^\Omega(A) \min_{\omega' \in A} c(\omega, \omega'), \tag{1.15}$$

where $c(\omega, \omega')$ is the cost of deciding ω when the true answer is ω'. Choosing the value ω minimizing the lower (resp. upper) expected costs is called the optimistic (resp. pessimistic) strategy.

Let us consider a particular situation when the set of focal elements is reduced to singletons and Ω, and with

$$c(\omega, \omega') = \begin{cases} 0, & \text{if } \omega = \omega', \\ 1, & \text{otherwise.} \end{cases} \tag{1.16}$$

In that case, the upper and lower expected costs are respectively and simply defined as

$$R^*(\omega) = 1 - m^\Omega(\{\omega\}) = 1 - Bel^\Omega(\{w\}), \tag{1.17}$$

$$R_*(\omega) = 1 - m^\Omega(\{\omega\}) - m^\Omega(\Omega) = 1 - Pl^\Omega(\{w\}). \tag{1.18}$$

This amounts to choosing the singleton with the highest belief or, equivalently, plausibility.

Example 1.4.1 *Let us consider again our example, but simplified. The question is now only to determine if it was a human (h) or an animal (a); the frame of discernment*

becomes thus in that case $\Omega = \{h, a\}$*. We consider that the final obtained mass function* m^Ω *is the following:*

$$m^\Omega(\{h\}) = 0.35, \quad m^\Omega(\{a\}) = 0.55, \quad m^\Omega(\Omega) = 0.1. \tag{1.19}$$

Using the maximum of belief or plausibility rules leads to the decision $\{Animal\}$*. The belief and plausibility functions are given in Table 1.3.*

We now consider that it is twice more serious to decide $\{Animal\}$ *when the true answer is* $\{Human\}$ *than the opposite,* i.e., $c(a, h) = 2$ *and* $c(h, a) = 1$*. Plus, we have* $c(a, a) = c(h, h) = 0$*. The corresponding risks are given in Table 1.3. In that case, the smallest upper expected cost* R^* *corresponds to the decision* $\{Human\}$*. The same conclusion can be made using the smallest lower expected costs. It illustrates the fact that using the decision costs can change a final decision.*

Decisions	Bel^Ω	Pl^Ω	R^*	R_*
$\{Human\}$	0.35	0.45	0.65	0.55
$\{Animal\}$	0.55	0.65	0.9	0.7

Table 1.3 – Belief and plausibility functions, risks.

To avoid making wrong decisions in the risky cases, *i.e.*, when the expected costs are high, a decision of rejection may be introduced. Formally, a reject cost $R_{rej} \geq 0$ is introduced and is compared to the chosen expected costs, for instance the upper expected costs. In that case, the decision to reject is made when R_{rej} is lower than the upper expected costs.

Example 1.4.2 *For instance, if we fix* $R_{rej} = 0.5$ *and we use the upper expected costs in Table 1.3, then using the preceding Example 1.4.1 the decision is now neither* $\{Animal\}$ *nor* $\{Human\}$ *but to reject.*

1.5 Statistical inference and forecasting

Belief function theory can also be used for estimation and forecasting. Estimation consists in representing the knowledge about an unknown parameter after observing some data while forecasting (also known as prediction) consists in making statements about a not yet observed data based on available data [97, 29, 25, 63, 62].

1.5.1 Estimation

Consider $\theta \in \Theta$ an unknown parameter, $x \in \mathbb{X}$ some observed data and $f_\theta(x)$ the density function generating the data. Statistical inference consists in making state-

ments about θ after observing the data x. Shafer [97] proposed to represent knowledge about θ given x by a consonant belief function Bel_x^Θ based on the likelihood function $L_x : \theta \to f_\theta(x)$ (see also justifications by Denoeux in [29]), whose contour function is the normalized likelihood function:

$$pl_x^\Theta(\theta) = \frac{L_x(\theta)}{\sup_{\theta' \in \Theta} L_x(\theta')}, \qquad \forall \theta \in \Theta. \tag{1.20}$$

Example 1.5.1 *Let us consider an important particular case. Assume that each train leaving Gare du Nord is either on time or not. We denote by θ the probability that a train is on time. Furthermore, we assume that a delayed train does not impact the departure of the other trains. Assume further that n trains have been observed and that $x \leq n$ of these trains were on time.*

Let X denote the number of trains that are on time out of n trains. X has thus a binomial distribution with parameters $n \in \mathbb{N}$ and $\theta \in [0,1]$, i.e., *$X \sim \mathcal{B}(n, \theta)$. If $x \leq n$ trains have been observed to be on time, the likelihood of value $\theta \in [0,1]$ is*

$$L_x(\theta) = \binom{n}{x} \theta^x (1-\theta)^{n-x}. \tag{1.21}$$

Thus, the likelihood-based belief function has the following contour function:

$$pl_x^\Theta(\theta) = \frac{\theta^x (1-\theta)^{n-x}}{\hat{\theta}^x (1-\hat{\theta})^{n-x}}, \tag{1.22}$$

for all $\theta \in \Theta = [0,1]$, where $\hat{\theta} = \frac{x}{n}$ is the Maximum Likelihood Estimate (MLE) of θ. Figure 1.1 shows the contour function of the binomial distribution, for $n = 30$ and $x = 10$, i.e., *we have observed 30 trains during the day and 10 of them were on time.*

1.5.2 Prediction

Let us now suppose that we have some knowledge about $\theta \in \Theta$ after observing some data x, given under a form of a consonant belief function Bel_x^Θ. The aim of forecasting is to make statements about a not yet observed data $Y \in \mathbb{Y}$, whose conditional distribution $g_{x,\theta}(y)$ given $X = x$ depends on θ. A solution to this problem, proposed by Kanjanatarakul *et al.* [63, 62], consists in using the fact that Bel_x^Θ is equivalent to a random set, and in using the sampling model of Dempster [25] to deduce a belief function on $\mathbb{Y}$. We detail these two points below.

Let us recall that the focal sets of Bel_x^Θ are the level sets of pl_x^Θ [79], defined by

$$\Gamma_x(\gamma) = \{\theta \in \Theta | pl_x^\Theta(\theta) \geq \gamma\}, \qquad \forall \gamma \in [0,1]. \tag{1.23}$$

Example 1.5.2 *For instance in Figure 1.1, for* $\gamma = \gamma_0 = 0.4$*, the set* $\Gamma_x(\gamma_0)$ *is defined as the set of all values of* $\theta \in \Theta$ *such that* $pl_x^\Theta(\theta) \geq 0.4$, i.e., $\Gamma_x(\gamma_0) = [a, b] \approx [0.225, 0.454]$.

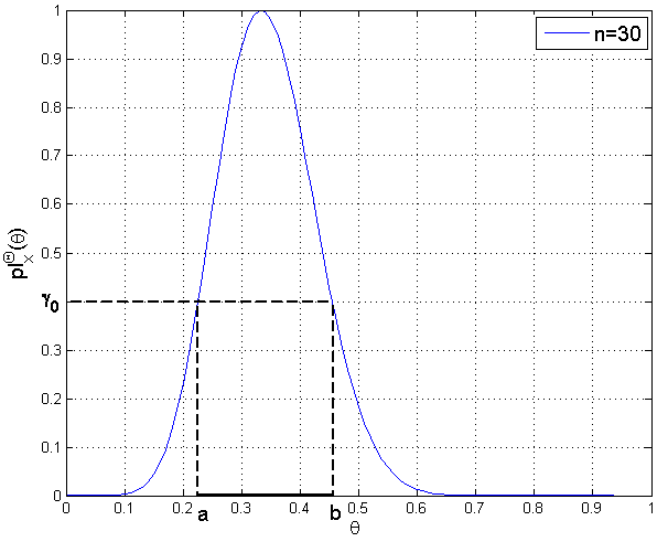


Figure 1.1 – Contour function of a binomial distribution, with $n = 30$ and $x = 10$.

Moreover, the belief function Bel_x^Θ is equivalent to the random set induced by the Lebesgue measure λ on $[0, 1]$ and the multi-valued mapping $\Gamma_x : [0, 1] \to \Theta$ [79]. Thus, we have

$$Bel_x^\Theta(A) = \lambda(\{\gamma \in [0, 1] | \Gamma_x(\gamma) \subseteq A\}), \tag{1.24}$$

$$Pl_x^\Theta(A) = \lambda(\{\gamma \in [0, 1] | \Gamma_x(\gamma) \cap A \neq \emptyset\}), \tag{1.25}$$

for all $A \subseteq \Theta$.

The sampling model of Dempster proposes to express Y using a function φ depending on the parameter θ and some unobserved variable $Z \in \mathbb{Z}$, whose probability distribution μ is known and independent of θ:

$$Y = \varphi(\theta, Z). \tag{1.26}$$

From Eqs. (1.23) and (1.26), for a given $(\gamma, z) \in [0, 1] \times \mathbb{Z}$, we can assert that $Y \in \varphi(\Gamma_x(\gamma), z)$. This can be represented by a multi-valued mapping $\Gamma'_x : [0, 1] \times \mathbb{Z} \to \mathbb{Y}$ defined by composing Γ_x with φ, *i.e.*, $\Gamma'_x(\gamma, z) = \varphi(\Gamma_x(\gamma), z), \forall (\gamma, z) \in [0, 1] \times \mathbb{Z}$. The product measure $\lambda \otimes \mu$ on $[0, 1] \times \mathbb{Z}$ and the multi-valued mapping Γ'_x induce the belief and plausibility functions on $\mathbb{Y}$, which are defined by

$$Bel_x^\mathbb{Y}(A) = (\lambda \otimes \mu)(\{(\gamma, z) | \varphi(\Gamma_x(\gamma), z) \subseteq A\}), \tag{1.27}$$

$$Pl_x^\mathbb{Y}(A) = (\lambda \otimes \mu)(\{(\gamma, z) | \varphi(\Gamma_x(\gamma), z) \cap A \neq \emptyset\}), \tag{1.28}$$

for all $A \subseteq \mathbb{Y}$.

Let us consider a binary case, which will be useful hereafter. Let $Y \in \mathbb{Y} = \{0, 1\}$ be a random variable with a Bernoulli distribution, *i.e.*, $Y \sim \mathcal{B}(\theta)$. In that case, the function φ can be defined as follows:

$$Y = \varphi(\theta, Z) = \begin{cases} 1, & \text{if } Z \leq \theta, \\ 0, & \text{otherwise,} \end{cases} \tag{1.29}$$

with Z having a uniform distribution on $[0, 1]$. Assume that the consonant belief function Bel_x^Θ has a unimodal and continuous contour function pl_x^Θ. In that case, each level set of Bel_x^Θ is a closed interval, *i.e.*, $\Gamma_x(\gamma) = [U(\gamma), V(\gamma)]$ [24], and the multi-valued mapping Γ'_x defined by composing Γ_x with φ, is given by

$$\Gamma'_x(\gamma, z) = \varphi([U(\gamma), V(\gamma)], z) = \begin{cases} \{1\}, & \text{if } z \leq U(\gamma), \\ \{0\}, & \text{if } z > V(\gamma), \\ \{0, 1\}, & \text{otherwise.} \end{cases} \tag{1.30}$$

By applying Eq. (1.27), we get

$$Bel_x^{\mathbb{Y}}(\{1\}) = (\lambda \otimes \mu)(\{(\gamma, z) | z \leq U(\gamma)\}), \tag{1.31}$$

$$Bel_x^{\mathbb{Y}}(\{0\}) = (\lambda \otimes \mu)(\{(\gamma, z) | z > V(\gamma)\}). \tag{1.32}$$

Kanjanatarakul *et al.* [63] showed that in this situation, Eq. (1.31) is equivalent to

$$Bel_x^{\mathbb{Y}}(\{1\}) = \int_0^{+\infty} (1 - F_U(u)) du, \tag{1.33}$$

where $F_U(u)$ is the cumulative distribution function of U. By definition, we have

$$F_U(u) = P(U \leq u), \tag{1.34}$$

$$= P([U, V] \cap]-\infty, u] \neq \emptyset), \tag{1.35}$$

$$= Pl_x^\Theta(]-\infty, u]), \tag{1.36}$$

$$= \begin{cases} pl_x^\Theta(u) & \text{if } u \leq \hat{\theta}, \\ 1 & \text{otherwise.} \end{cases} \tag{1.37}$$

Finally, using Eqs. (1.33) and (1.37), we thus obtain the following belief function:

$$Bel_x^{\mathbb{Y}}(\{1\}) = \hat{\theta} - \int_0^{\hat{\theta}} pl_x^\Theta(u) du, \tag{1.38}$$

where $\hat{\theta}$ maximizes pl_x^Θ. The same reasoning can be applied for the plausibility function and we obtain

$$Pl_x^{\mathbb{Y}}(\{1\}) = \hat{\theta} + \int_{\hat{\theta}}^1 pl_x^\Theta(u) du. \tag{1.39}$$

These functions are equivalent to the mass function illustrated in Figure 1.2 and which is defined by

$$m_x^{\mathbb{Y}}(\{0\}) = 1 - \hat{\theta} - \int_{\hat{\theta}}^{1} pl_x^{\Theta}(u)du, \tag{1.40}$$

$$m_x^{\mathbb{Y}}(\{1\}) = \hat{\theta} - \int_{0}^{\hat{\theta}} pl_x^{\Theta}(u)du, \tag{1.41}$$

$$m_x^{\mathbb{Y}}(\{0,1\}) = \int_{0}^{1} pl_x^{\Theta}(u)du, \tag{1.42}$$

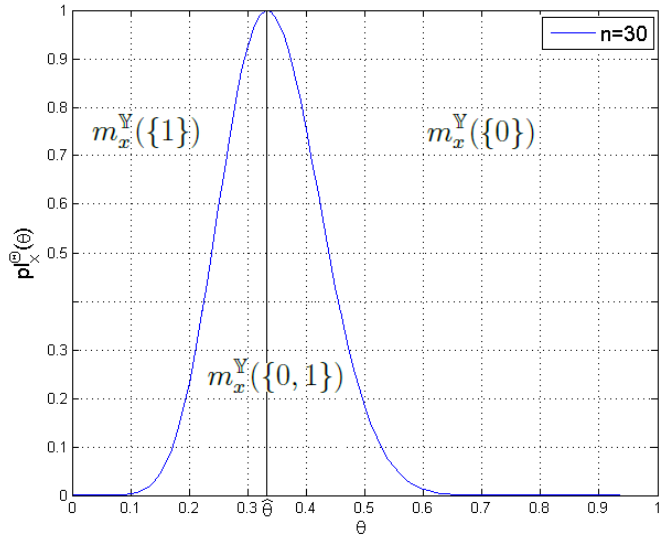


Figure 1.2 – Illustration of the mass function $m_x^{\mathbb{Y}}$, which depends on the contour function pl_x^{Θ}.

The part of ignorance in $m_x^{\mathbb{Y}}$ depends on the area under the contour function curve. Figure 1.3 shows the different obtained contour functions of the binomial distribution, for $n \in \{3, 30, 300\}$ and a fixed $\hat{\theta} = 0.33$. As it can be seen, the bigger n, *i.e.*, the bigger the number of trials, the smaller the uncertainty, as the area under the curve is lower.

Example 1.5.3 *Let us consider again the particular case of Section 1.5.1, where* $X \sim \mathcal{B}(n, \theta)$*. After observing some trains, we now have some knowledge about the probability* θ *that a train departure is on time. In that case, the contour function on* Θ *defined in Eq.* (1.22) *is unimodal and continuous, as illustrated in Figure 1.1. Thus, to represent knowledge about the on-time departure of an unobserved train,* i.e.*, the data* $Y \in \mathbb{Y}$*, with* $Y \sim \mathcal{B}(\theta)$*, we can apply Eqs.* (1.38) *and* (1.39)*. Xu* et al. *showed that the obtained belief and plausibility functions boil down in that case to [116]:*

$$Bel_x^{\mathbb{Y}}(\{1\}) = \begin{cases} 0, & \text{if } \hat{\theta} = 0, \\ \hat{\theta} - \frac{B(\hat{\theta}; x+1, n-x+1)}{\hat{\theta}^x (1-\hat{\theta})^{n-x}}, & \text{if } 0 < \hat{\theta} < 1, \\ \frac{n}{n+1}, & \text{if } \hat{\theta} = 1, \end{cases} \tag{1.43}$$

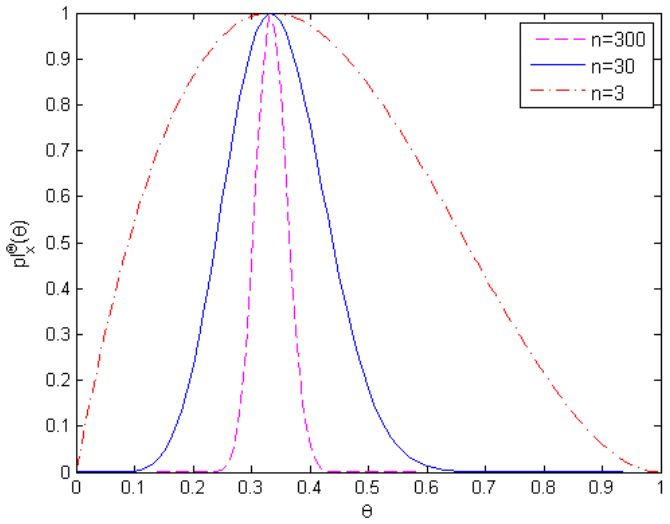


Figure 1.3 – Contour function of a binomial distribution, with $n \in \{3, 30, 300\}$ and $\hat{\theta} = 0.33$.

$$Pl_x^{\mathbb{Y}}(\{1\}) = \begin{cases} \frac{1}{n+1}, & \text{if } \hat{\theta} = 0, \\ \hat{\theta} + \frac{\overline{B}(\hat{\theta}; x+1, n-x+1)}{\hat{\theta}^x (1-\hat{\theta})^{n-x}}, & \text{if } 0 < \hat{\theta} < 1, \\ 1, & \text{if } \hat{\theta} = 1, \end{cases} \tag{1.44}$$

where $\underline{B}$ *and* $\overline{B}$ *are respectively the lower and upper incomplete beta functions, defined when* a *and* b *are integers and* $0 < z < 1$ *by*

$$\underline{B}(z; a, b) = \sum_{j=a}^{a+b-1} \frac{(a-1)!(b-1)!}{j!(a+b-1-j)!} z^j (1-z)^{a+b-1-j}, \tag{1.45}$$

and

$$\overline{B}(z; a, b) = \underline{B}(1-z; b, a). \tag{1.46}$$

To continue the example where we observed 30 trains where 10 were on time, and thus $\hat{\theta} \approx 0.33$*, the obtained values of the belief and plausibility functions using Eqs.* (1.43) *and* (1.44) *are* $Bel_x^{\mathbb{Y}}(\{On\ time\}) = 0.23$ *and* $Pl_x^{\mathbb{Y}}(\{On\ time\}) = 0.44$*. Let us note that the chosen data are not the real ones and that it does not reflect the reality of on-time trains.*

1.6 Conclusion

In this chapter, we have introduced the fundamental concepts of the theory of belief functions, including the representation of evidence, the combination of pieces

of evidence, the decision-making process, and the application of belief functions to inference and forecasting. We gave some simple examples to illustrate all these concepts.

Belief function theory is a powerful tool to handle uncertainties. For instance, in the following chapter, it is used to improve some probabilistic calibration techniques and specifically to better represent the uncertainties inherent to the calibration process.

Chapter 2

Evidential calibration of scores

Contents

2.1 Introduction

As recalled in Introduction, the combination methods regroup the approaches using the outputs of several base classifiers as input to another classifier. Since the base classifiers do not necessarily give the same output after observing a given object,

a central issue in this approach consists in figuring out how to exploit these outputs to classify this object. These various combination methods are usually separated into two categories: the non-trainable and trainable combiners.

In the first category, the outputs returned by the classifiers after observing a given object are combined using a predetermined rule of combination. As the used classifiers are different, *i.e.*, they may be trained with different data or based on different training models, their outputs are not scaled with respect to each other and thus have to be made comparable before being combined. A step called calibration [85] is thus usually performed to transform each output into a probability. In particular, the three calibration techniques the most commonly used are based on binning [120], isotonic regression [121] and logistic regression [85]. These calibration techniques suffer from an over-fitting problem, especially when only few training data are available. Within this scope, Xu *et al.* [116] recently proposed a refinement of the main calibration procedures using evidence theory [97, 100]. This theory allows Xu *et al.* to model more precisely the uncertainties inherent to such calibration process and thus to prevent the over-fitting issue. Xu *et al.* used this refinement to propose in [116] an approach of the non-trainable kind for binary classification problems. This latter approach consists in: using several SVM classifiers returning confidence scores, calibrating each of the returned scores using an evidential calibration technique, hence transforming each of the score into a belief function, and finally merging them using Dempster's rule of combination [97].

The second category regroups the approaches using the concatenation of the outputs of the classifiers as an input vector for another classifier. In particular, the approach defined in [125] is a member of that category as a vector of scores obtained from an ensemble of classifiers is provided as an input vector to a probabilistic classifier based on multiple isotonic regression. Note that such kind of approach may be regarded as a probabilistic *joint* calibration as it learns how to convert a vector of scores into a probability, that is it calibrates jointly the classifiers. In addition, as logistic regression can also be defined with multiple inputs [52], one may envisage to extend this kind of approach to the logistic model.

Both categories present some disadvantages. As already mentioned, the calibration techniques used in the non-trainable combiners are prone to uncertainties. In addition, non-trainable combiners rely on a fixed rule of combination; as explained in particular by Duin [36], an improved result may be obtained using an approach of the trainable combiner category. For the trainable combiners, a training set common to all classifiers is required, and the combiner must be re-learned each time a new classifier is added to the system. Furthermore, trainable combiner approaches corresponding to a probabilistic joint calibration may also be prone to the uncertainties problem inherent to probabilistic calibration.

Within this scope, we propose in this chapter to study the application of the appealing element of Xu *et al.*'s approach [116], *i.e.*, the evidential extension of cali-

bration, to joint calibration techniques. As a result, we obtain methods that transform the vector of scores returned by the classifiers for a given object into a belief function. Let us note that we only consider binary classification problems.

This chapter is organized as follows. In Section 2.2, probabilistic calibration methods of a single classifier are presented, followed by their extension using the evidence theory. Then, probabilistic joint calibrations and their extension to the evidential framework that we propose, are exposed in Section 2.3. In Section 2.4, the proposed approach is compared experimentally to other approaches, and in particular to Xu *et al.* non-trainable combiner approach relying on evidential calibration of individual classifiers and to probabilistic joint calibration. Finally, conclusions are given in Section 2.5.

2.2 Calibration of a single classifier

Let us consider an object, whose true label y is such that $y \in \mathbb{Y} = \{0, 1\}$, and a confidence score $s \in \mathbb{R}$ returned by a classifier after observing this object. To learn how to interpret what this score represents with respect to y, a step called calibration may be used. This step relies on a training set $\mathcal{X}$, which contains n other objects for which the label is known, and for which we observed the score that the classifier returned, *i.e.*, $\mathcal{X} = \{(s_1, y_1), ..., (s_n, y_n)\}$ where s_i represents the score given by the classifier for the i^{th} object whose true label is y_i. The calibration procedures commonly used are the binning [120], isotonic regression [121] and logistic regression [85]. The probabilistic version of these calibrations is described in Section 2.2.1, followed by their extension to the evidential framework in Section 2.2.2.

2.2.1 Probabilistic calibration of a single classifier

Given a score $s \in \mathbb{R}$ returned by a classifier after observing a given object, the aim of the calibration in the probabilistic framework consists in estimating the probability distribution $p^{\mathbb{Y}}(\cdot|s)$.

Binning The binning approach consists in dividing the score spaces into B_U different bins, for example $]-3; -2],]-2; -1]$, etc. For each bin j, the number k_j of couples $(s_i, y_i) \in \mathcal{X}$ such that $y_i = 1$ and s_i is in bin j, and the number n_j of couples $(s_i, y_i) \in \mathcal{X}$ such that s_i is in bin j can be obtained. Then, for a score s such that s belongs to bin j, we have

$$P^{\mathbb{Y}}(y = 1|s) = \frac{k_j}{n_j}. \tag{2.1}$$

There are different ways of building the bins, *i.e.*, choosing the size and the boundaries position of each bin. For instance, one may find the lowest and highest scores in the training set and divide the interval by the desired total number of bins.

Isotonic regression The second main calibration is the method based on the isotonic regression. It was proposed in [121] and consists in fitting a non-decreasing stepwise-constant function g, *i.e.*, an isotonic function, according to the mean-squared error criterion:

$$\hat{g} = \arg\min_{g} \frac{1}{n} \sum_{i=1}^{n} [g(s_i) - y_i]^2, \tag{2.2}$$

such that $g(s_1) < \cdots < g(s_n)$ and where $\hat{g}$ is the vector of calibrated probability estimates. An iterative algorithm called the pair-adjacent violators (PAV) algorithm [4] can be used in order to find the optimal function $\hat{g}$ that best fits the data: first, the couples $(s_i, y_i) \in \mathcal{X}$ are ranked with respect to s_i in increasing order. Then, the algorithm analyzes all the data looking for violations of the monotonicity constraint, *i.e.*, a situation where $g(s_{i-1}) > g(s_i)$. The examples s_{i-1} and s_i are thus called pair-adjacent violators, and the values of $g(s_{i-1})$ and $g(s_i)$ are replaced. This PAV algorithm is presented in Algorithm 1 (taken from [116]).

Algorithm 1 PAV algorithm for isotonic calibration

Input: training set $\mathcal{X} = \{(s_1, y_1), ..., (s_n, y_n)\}$ sorted according to s_i
$\hat{g}_{i,i} \leftarrow 0, w_i \leftarrow 0$
$\hat{g}_1 \leftarrow 1, w_1 \leftarrow 1$
$i \leftarrow 1$
for j=2:n **do**
  $i \leftarrow i + 1$
  $\hat{g}(s_i) \leftarrow y_j$
  $w_i \leftarrow w_j$
  while $i > 2$ and $\hat{g}(s_{i-1}) > \hat{g}(s_i)$ **do**
    $\hat{g}(s_{i-1}) \leftarrow \frac{w_{i-1}\hat{g}(s_{i-1}) + w_i \hat{g}(s_i)}{w_{i-1} + w_i}$
    $w_{i-1} \leftarrow w_{i-1} + w_i$
    $i \leftarrow i - 1$
  end while
end for
Output: $\hat{g}(s) = \hat{g}_{i,j}$ for $s_i < s < s_j$

The output is thus a set of intervals and a probability estimate associated to each interval. For a given score s to be calibrated, the interval k such that s is between the lowest and highest scores in this interval is found, and the probability estimate $P^{\mathbb{Y}}(y = 1|s)$ is thus $\hat{g}(k)$. Let us note that this calibration can be seen as a form of binning, where the position of the boundaries and the size of the bins are dynamically calculated instead of being fixed, and which entirely depends on the training set $\mathcal{X}$.

Logistic regression The third calibration is similar to the isotonic regression-based calibration, but the difference is the function being fit. It is similar in the sense that both have a non-decreasing constraint, *i.e.*, the higher the score the higher the probability of having the positive class, that binning does not have. Thus, the calibration based on isotonic regression can be seen as an intermediary approach between binning and logistic regression [121].

The calibration based on logistic regression proposed by Platt [85] is based on fitting a sigmoid function h defined by

$$P^{\mathbb{Y}}(y = 1|s) \approx h_s(\sigma) = \frac{1}{1 + e^{-(\sigma_0 + \sigma_1 s)}}, \tag{2.3}$$

where the parameter $\sigma = (\sigma_0, \sigma_1) \in \mathbb{R}^2$ is chosen as the one maximizing the following likelihood function:

$$L_{\mathcal{X}}(\sigma) = \prod_{i=1}^{n} p_i^{y_i}(1 - p_i)^{1-y_i}, \tag{2.4}$$

with

$$p_i = \frac{1}{1 + e^{-(\sigma_0 + \sigma_1 s_i)}}. \tag{2.5}$$

To find the optimal parameters $\hat{\sigma} = (\hat{\sigma}_0, \hat{\sigma}_1)$, usually a maximisation algorithm such as gradient ascent is used. Since the logarithm function is a strictly increasing function, maximizing the logarithm of the likelihood is the same as maximizing the likelihood, except that it is usually easier to do it. The log-likelihood is defined by

$$\ell(\sigma) = \log L_{\mathcal{X}}(\sigma) \tag{2.6}$$

$$= \sum_{i=1}^{n} \Big(y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \Big). \tag{2.7}$$

The computation of the gradient of the log-likelihood at iteration k is denoted by $\nabla \ell(\sigma) = [\frac{\partial \ell(\sigma)}{\partial \sigma_0}, \frac{\partial \ell(\sigma)}{\partial \sigma_1}]$ and gives:

$$\frac{\partial \ell(\sigma)}{\partial \sigma_0} = \sum_{i}^{n} (y_i - p_i), \quad \frac{\partial \ell(\sigma)}{\partial \sigma_1} = \sum_{i}^{n} (y_i - p_i) s_i. \tag{2.8}$$

The gradient ascent is an iterative method and so the parameters are updated at each iteration k, until convergence, with:

$$\sigma_j^{(k+1)} \leftarrow \sigma_j^{(k)} + \eta \frac{\partial \ell(\sigma^{(k)})}{\partial \sigma_j}, \tag{2.9}$$

where η is called the learning rate. The whole gradient ascent algorithm is presented in Algorithm 2.

It may happen that the training data are perfectly (linearly) separable, *i.e.*, all data of the first class are in one half-space and those of the second class are in the

Algorithm 2 Batch gradient ascent to find the optimal parameters

Input: training set $\mathcal{X} = \{(s_1, y_1), ..., (s_n, y_n)\}$, error criterion ϵ.
Initialization: $\sigma^{(0)} = (0, 0)$.
while $\left\|\nabla \ell(\sigma^{(k)})\right\| \geq \epsilon$ **do**
  Compute the gradient: $\nabla \ell(\sigma^{(k)}) = [\frac{\partial \ell(\sigma^{(k)})}{\partial \sigma_0}, \frac{\partial \ell(\sigma^{(k)})}{\partial \sigma_1}]$
  Update the parameters : $\sigma_j^{(k+1)} \leftarrow \sigma_j^{(k)} + \eta \frac{\partial \ell(\sigma^{(k)})}{\partial \sigma_j}, \quad j = 0, 1$
end while
Output: $\hat{\sigma} = (\hat{\sigma}_0, \hat{\sigma}_1)$

other half-space. It especially happens when only few data are available. In that case, the gradient ascent does not converge and the parameter σ tends to infinity. To solve this issue, Platt [85] proposed to change the labels y_i by

$$t_i = \begin{cases} \frac{N_+ + 1}{N_+ + 2} & \text{if } y_i = 1, \\ \frac{1}{N_- + 2} & \text{if } y_i = 0, \end{cases} \tag{2.10}$$

where N_+ and N_- are respectively the number of positive and negative samples in the training set $\mathcal{X}$. In order to illustrate this point, we trained a logistic-based calibration with, then without, this change of labels. It was trained with 10 examples of the Australian dataset[1] that we selected so that they had the particularity to be linearly separable. Then, the probability was computed, in both cases, for score range between -3 and 3. Figure 2.1 illustrates the impact of this change of label, that enables to smooth the function.

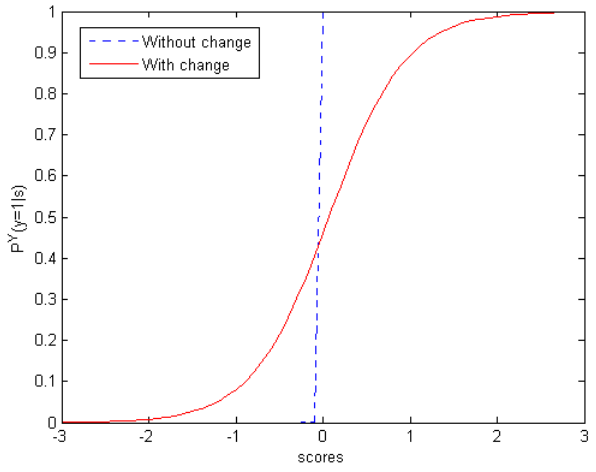


Figure 2.1 – Illustration of the impact of the label change. Calibration trained with 10 linearly separable examples of Australian dataset.

[1] UCI dataset available at http://archive.ics.uci.edu/ml.

We may remark that the less training samples are available, the more the estimated probabilities are uncertain. For instance, if a bin contains only few examples, the uncertainty is higher than a bin containing more data. Within this scope, the above calibrations have recently been refined using the theory of evidence, in order to better handle the uncertainties [116]. The following section recalls the evidential versions of these calibration procedures.

2.2.2 Evidential calibration of a single classifier

The calibration of a given score s can be seen as a prediction problem of a Bernoulli variable $Y \in \mathbb{Y} = \{0, 1\}$ with parameter θ, where uncertainty on θ depends on s. Different models to estimate the uncertainty on θ have been studied in [116], and the authors highlighted in particular the benefits of the so-called likelihood-based model. Thus, this chapter focuses on the evidential extension of the calibrations based on this likelihood model. These evidential calibration procedures yields a MF $m^{\mathbb{Y}}(\cdot|s)$ (rather than a probability distribution), equivalently represented by the belief and plausibility functions $Bel^{\mathbb{Y}}(\cdot|s)$ and $Pl^{\mathbb{Y}}(\cdot|s)$.

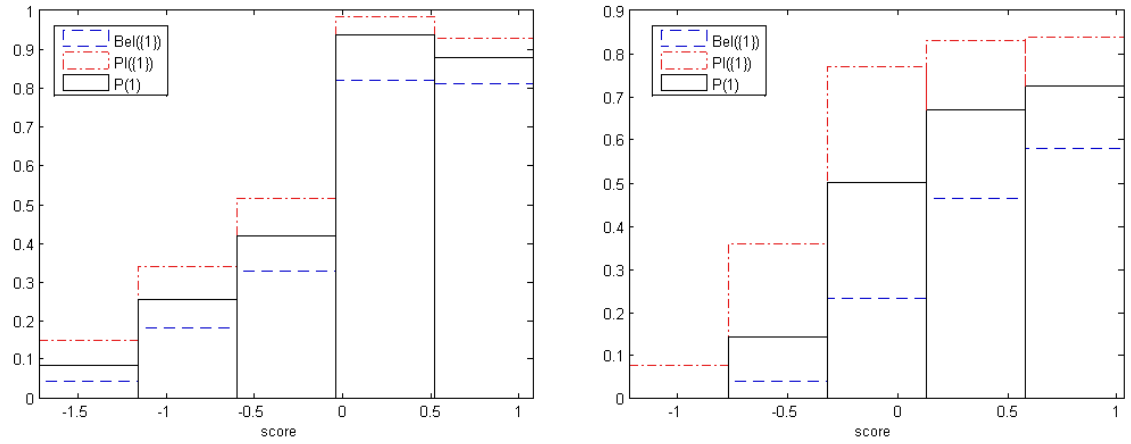
Binning For a given bin j, binning can be seen as a binomial experiment, where the number of examples n_j corresponds to the number of trials and the number of positive examples k_j represents the number of successes. Thus, it corresponds to the particular case of estimation considered in Section 1.5.1, and used for forecasting in Section 1.5.2. Considering that the given score s is in bin j, the likelihood-based contour function defined in Eq. (1.22) becomes

$$pl_{\mathcal{X}}^{\Theta}(\theta|s) = \frac{\theta^{k_j}(1-\theta)^{n_j-k_j}}{\hat{\theta}^{k_j}(1-\hat{\theta})^{n_j-k_j}}, \tag{2.11}$$

where $\hat{\theta} = \frac{k_j}{n_j}$ is the Maximum Likelihood Estimate (MLE) of θ. The belief and plausibility functions $Bel^{\mathbb{Y}}(\cdot|s)$ and $Pl^{\mathbb{Y}}(\cdot|s)$ are then simply obtained using Eq. (1.43) and (1.44) with $x = k_j$ and $n = n_j$.

Example 2.2.1 *Figure 2.2 illustrates the probability, belief and plausibility of having a positive example given a confidence score returned by a SVM classifier trained with a UCI dataset [5] called Australian. In Figure 2.2a (resp. Figure 2.2b), the training set of calibration is composed of 200 (resp. 50) examples. As it can be noticed, the interval between $Bel^{\mathbb{Y}}(\cdot|s)$ and $Pl^{\mathbb{Y}}(\cdot|s)$ is higher when there are less examples in the training set,* i.e.*, there is more ignorance, as should be. Thus, if a bin contains many training examples, the ignorance is low, and vice versa. This information cannot be obtained with the probabilistic calibration, as it is represented by only one value. Thus, the calibration based on evidence theory better reflects the uncertainties. It can be noticed*

in Figure 2.2a that the calibration based on binning is not a strictly increasing function, unlike the other two calibrations.



(a) Calibration trained with 200 examples. (b) Calibration trained with 50 examples.

Figure 2.2 – Illustration of calibration based on evidential binning and trained with 200 (left) and 50 (right) examples with the Australian dataset.

Isotonic regression As noticed in Section 2.2.1, the calibration based on isotonic regression can be seen as a form of binning. Thus, the extension to the evidential framework used for binning can be straightforwardly applied to this calibration [116].

Logistic regression Logistic-based calibration can also be extended in the evidential framework. Specifically, Xu *et al.* [116] express uncertainty on the parameter $\sigma = (\sigma_0, \sigma_1)$ of the sigmoid function, by a consonant belief function Bel^{Σ}, whose contour function is defined by

$$pl_{\mathcal{X}}^{\Sigma}(\sigma) = \frac{L_{\mathcal{X}}(\sigma)}{L_{\mathcal{X}}(\hat{\sigma})}, \quad \forall \sigma \in \Sigma, \tag{2.12}$$

where $\hat{\sigma} = (\hat{\sigma}_0, \hat{\sigma}_1)$ is the MLE of σ and $L_{\mathcal{X}}$ is the likelihood function defined in Eq. (2.4). The corresponding plausibility function is defined as

$$Pl_{\mathcal{X}}^{\Sigma}(A) = \sup_{\sigma \in A} pl_{\mathcal{X}}^{\Sigma}(\sigma), \quad \forall A \subseteq \Sigma. \tag{2.13}$$

As seen in Section 1.5.2, the belief and plausibility functions on $\mathbb{Y}$ can be deduced from the contour function $pl_{\mathcal{X}}^{\Theta}$ defined on Θ. Xu *et al.* showed in [116] that this function $pl_{\mathcal{X}}^{\Theta}$ can be computed from $Pl_{\mathcal{X}}^{\Sigma}$. Indeed, as θ is defined by $\theta = h_s(\sigma)$, we get

$$pl_{\mathcal{X}}^{\Theta}(\theta|s) = \begin{cases} 0 & \text{if } \theta \in \{0, 1\}, \\ Pl_{\mathcal{X}}^{\Sigma}(h_s^{-1}(\theta)) & \text{otherwise,} \end{cases} \tag{2.14}$$

with

$$h_s^{-1}(\theta) = \{(\sigma_0, \sigma_1) \in \Sigma | h_s(\sigma) = \theta\}, \tag{2.15}$$

$$= \left\{(\sigma_0, \sigma_1) \in \Sigma | \frac{1}{1 + \exp(-(\sigma_0 + \sigma_1 s))} = \theta\right\}, \tag{2.16}$$

$$= \left\{(\sigma_0, \sigma_1) \in \Sigma | \sigma_0 = -\ln(\theta^{-1} - 1) - \sigma_1 s\right\}. \tag{2.17}$$

Finally, Eqs. (2.14) and (2.17) yield the following function

$$pl_{\mathcal{X}}^{\Theta}(\theta|s) = \sup_{\sigma_1 \in \mathbb{R}} pl_{\mathcal{X}}^{\Sigma}(-\ln(\theta^{-1} - 1) - \sigma_1 s, \sigma_1), \quad \forall \theta \in [0, 1]. \tag{2.18}$$

The value σ_1 maximizing $pl_{\mathcal{X}}^{\Theta}(\theta|s)$ can be obtained using a gradient ascent as in the probabilistic version. In that case, the value of the partial derivative is obtained by

$$\frac{\partial \ell(\sigma^{(k)})}{\partial \sigma_1} = \sum_i (y_i - p'_i)(s_i - s), \tag{2.19}$$

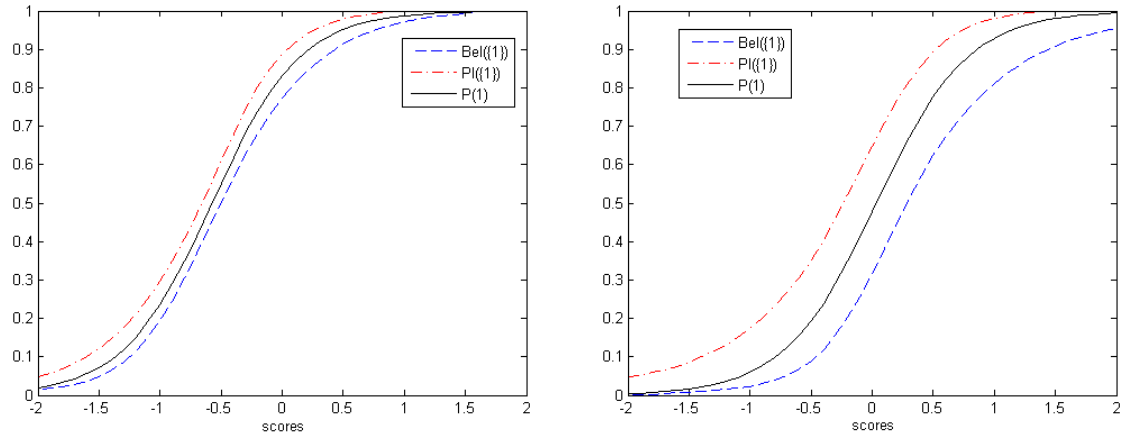
where

$$p'_i = \frac{1}{1 + \exp(-(-\ln(\theta^{-1} - 1) - \sigma_1^{(k)} s + \sigma_1^{(k)} s_i))}. \tag{2.20}$$

After that $pl_{\mathcal{X}}^{\Theta}(\theta|s)$ is computed, the belief and plausibility functions $Bel^{\mathbb{Y}}(\cdot|s)$ and $Pl^{\mathbb{Y}}(\cdot|s)$ can then be calculated using Eqs. (1.38) and (1.39).

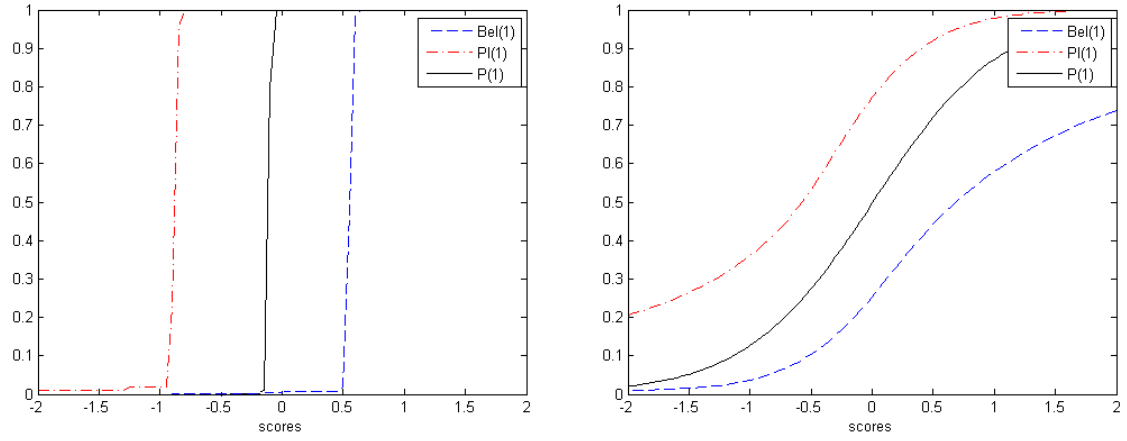
Example 2.2.2 *Figure 2.3 illustrates the probability, belief and plausibility obtained with a calibration using logistic regression and the Australian dataset. In Figure 2.3a, the training set of calibration is composed of 200 examples and we can notice that the ignorance is lower than in Figure 2.3b, where the calibration is trained with 50 examples. This difference of level of ignorance cannot be obtained with the probabilistic calibration.*

Let us consider again the particular situation where the data are linearly separable. Figure 2.4 shows the probability, belief and plausibility functions obtained in that case, without (Figure 2.4a) and with (Figure 2.4b) the change of labels. We may notice that without the change, we obtained either a total ignorance or a null ignorance, depending on the value of the score. This model reflects well the reality, as in the used training data there were no score between -1 and 0.5 and in that case for a given new score, the obtained belief function corresponds to total ignorance; we had no information, so we have no knowledge. Yet, it is reasonable to think that for a score close to one of the class, for instance $s = -0.99$, it is more likely that the sample belongs to the negative class. In that sense, and after performing some tests, we may estimate that the modelling is better with the change of labels for evidential logistic regression, as it is smoother.



(a) Calibration trained with 200 examples. (b) Calibration trained with 50 examples.

Figure 2.3 – Illustration of calibration based on logistic regression and trained with 200 and 50 examples, with the Australian dataset.



(a) Without change. (b) With change.

Figure 2.4 – Logistic-based calibration trained with 10 examples of Australian that are perfectly separable.

2.3 Evidential joint calibration of multiple classifiers

In a context of multiple classifiers, one may independently calibrate the scores given by each classifier after observing an object, using the techniques described in the previous section, and then merge them using a predetermined rule of combination. Yet, using a fixed rule may be the best combination only under very strict conditions, and an improved result may be obtained using an approach of the trainable combiner category

[36]. We propose in this section to use the multivariable versions of the techniques underlying the calibrations, and to apply it to the outputs of multiple classifiers, *i.e.*, to perform a joint calibration of the scores provided by binary classifiers. More specifically, in order to better handle the uncertainties of the calibration process, we propose to perform the joint calibration in the evidential framework. As the isotonic regression can be seen as an intermediary approach between binning and logistic regression [121], only this latter two are considered in this report.

For a given object, we take as input the score vector $\mathbf{s} = (s_1, s_2, ..., s_J)$, with s_j the score returned by the j^{th} classifier after observing the object. The required training set is now defined by $\mathcal{X}' = \{(s_{11}, s_{21}, ..., s_{J1}, y_1), ..., (s_{1n}, s_{2n}, ..., s_{Jn}, y_n)\}$, where s_{ji} corresponds to the score given by the j^{th} classifier for the i^{th} test sample, and y_i the true label of this sample.

We first expose in Section 2.3.1 the joint version of binning calibration, followed by the joint version of the calibration based on logistic regression in Section 2.3.2.

2.3.1 Joint binning

The idea consists in dividing the score space into multi-dimensional bins (cells), or more precisely into J-dimensional bins with J the number of classifiers. Let us illustrate the building of these cells with a $2D$ scenario, *i.e.*, when only two classifiers are considered. If the first classifier has score values between -3 and 3 and the second classifier has score values between -2 and 1, the score space is $[-3, 3] \times [-2, 1]$. This score space can be divided in different ways. In particular, a number of bins per classifier can be chosen and the score space can be divided uniformly based on this number. An illustration of this naive scheme is given in Figure 2.5, where the number of bins by classifier, denoted B_M, is chosen equal to 5.

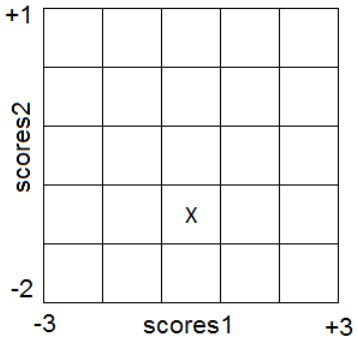


Figure 2.5 – Example of score space for joint binning, with $J = 2$ and $B_M = 5$.

Given a cell c, the number k_c of tuples $(s_{1i}, s_{2i}, ..., s_{Ji}, y_i) \in \mathcal{X}'$ such that $y_i = 1$ and $(s_{1i}, s_{2i}, ..., s_{Ji})$ belongs to cell c, and the number n_c of tuples such

that $(s_{1i}, s_{2i}, ..., s_{Ji})$ belongs to cell c, can be obtained. For a given input vector $\mathbf{s} = (s_1, s_2, ..., s_J)$ such that $\mathbf{s}$ belongs to the cell c, we have

$$P^{\mathbb{Y}}(y = 1|\mathbf{s}) = \frac{k_c}{n_c}. \tag{2.21}$$

For instance, let us consider that we have $\mathbf{s} = (0.5, -1)$, *i.e.*, after observing a given example the first classifier returns the score 0.5 and the second -1. The probability associated to this object can thus be found by looking into the corresponding cell c, which is the one marked by a cross in Figure 2.5.

This probabilistic joint approach of binning can be extended to the evidential framework. Similarly to the single classifier case, the label y of a given score vector $\mathbf{s}$ can be seen as a realization of a random variable with a Bernoulli distribution, and binning can be seen as a binomial experiment for each cell. If the score vector $\mathbf{s}$ is in cell c, the belief and plausibility functions associated to this score vector can be calculated using the following equations:

$$Bel^{\mathbb{Y}}(\{1\}|\mathbf{s}) = \begin{cases} 0, & \text{if } \hat{\theta} = 0, \\ \hat{\theta} - \frac{\underline{B}(\hat{\theta}; k_c+1, n_c-k_c+1)}{\hat{\theta}^{k_c}(1-\hat{\theta})^{n_c-k_c}}, & \text{if } 0 < \hat{\theta} < 1, \\ \frac{n_c}{n_c+1}, & \text{if } \hat{\theta} = 1, \end{cases} \tag{2.22}$$

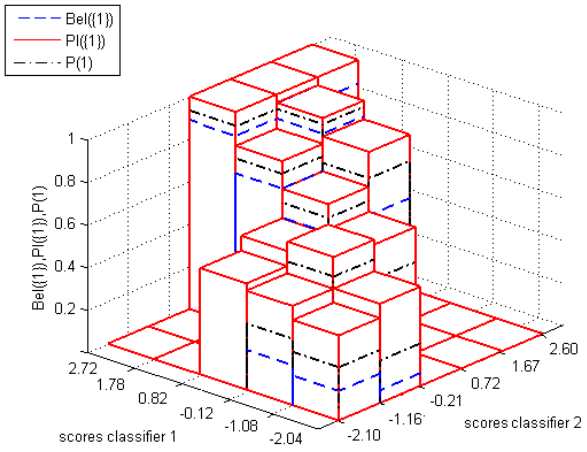
$$Pl^{\mathbb{Y}}(\{1\}|\mathbf{s}) = \begin{cases} \frac{1}{n_c+1}, & \text{if } \hat{\theta} = 0, \\ \hat{\theta} + \frac{\overline{B}(\hat{\theta}; k_c+1, n_c-k_c+1)}{\hat{\theta}^{k_c}(1-\hat{\theta})^{n_c-k_c}}, & \text{if } 0 < \hat{\theta} < 1, \\ 1, & \text{if } \hat{\theta} = 1, \end{cases} \tag{2.23}$$

with $\hat{\theta} = \frac{k_c}{n_c}$. Let us recall that the beta functions $\underline{B}$ and $\overline{B}$ are given in Eqs. (1.45) and (1.46).

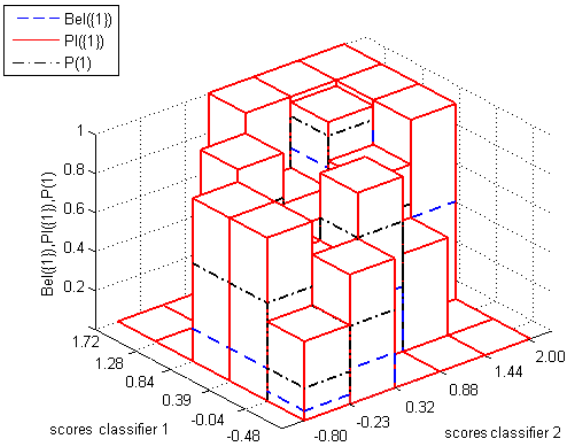
Example 2.3.1 *Figure 2.6 gives an illustration of the multi-dimensional bins obtained using the evidential joint calibration based on binning. The dataset considered in this example is the Diabetes dataset of UCI repository [5]. Two SVM classifiers, trained with 25 examples each, return a score after observing a given example. The number of bins per classifier is 5,* i.e., *$J = 2$ and $B_M = 5$. For the sake of a better visibility, the cells corresponding to the situation where s_1 is high and s_2 is low (the opposite is also true) are set to zero, but it actually corresponds to the case of total ignorance, as there are no tuple (s_1, s_2) in $\mathcal{X}'$ that belongs to these cells. As it can be noticed, the interval between $Bel^{\mathbb{Y}}(.|\boldsymbol{s})$ and $Pl^{\mathbb{Y}}(.|\boldsymbol{s})$ is higher in Figure 2.6a than in Figure 2.6b,* i.e., *when there are less examples in the training set.*

2.3.2 Joint logistic regression

The logistic regression, exposed in Section 2.2, is used to calibrate a score given by a single classifier. Yet, the logistic model works as well when more than one



(a) Calibration trained with 200 examples.



(b) Calibration trained with 50 examples.

Figure 2.6 – Illustration of joint calibration based on binning and trained with 200 and 50 examples, using Diabetes.

input is available: it is then called a multivariable (or multiple) logistic regression [52]. It has been widely used in many applications, such as for instance in the medicine field [6]. We propose to use this multiple version of logistic regression and apply it to the vector of scores returned by different classifiers for a given object, in order to calibrate this vector.

Given a vector of scores $\mathbf{s} = (s_1, s_2, ..., s_J)$, the probabilistic joint calibration based on multiple logistic regression is defined by

$$P^{\mathbb{Y}}(y = 1|\mathbf{s}) = \frac{1}{1 + e^{-(\sigma_0 + \sigma_1 s_1 + \sigma_2 s_2 + ... + \sigma_J s_J)}}, \tag{2.24}$$

where the parameter $\sigma = (\sigma_0, ..., \sigma_J) \in \mathbb{R}^{J+1}$ is obtained by maximizing the likelihood

function $L_{\mathcal{X}'}$ defined by

$$L_{\mathcal{X}'}(\sigma) = \prod_{i=1}^{n} p_i^{t_i}(1-p_i)^{1-t_i}, \tag{2.25}$$

with

$$p_i = \frac{1}{1+\exp(-(\sigma_0 + \sigma_1 s_{1i} + ... + \sigma_J s_{Ji}))}, \tag{2.26}$$

and

$$t_i = \begin{cases} \frac{N_+ +1}{N_+ +2} & \text{if } y_i = 1, \\ \frac{1}{N_- +2} & \text{if } y_i = 0, \end{cases} \tag{2.27}$$

where N_+ and N_- are respectively the number of positive and negative samples in the training set $\mathcal{X}'$. The log-likelihood can be used instead of the likelihood, and following the same gradient ascent algorithm than in Section 2.2 the optimal parameters $\sigma = \{\sigma_0, ..., \sigma_J\}$ can be estimate using the following partial derivatives:

$$\frac{\partial \ell(\sigma)}{\partial \sigma_0} = \sum_i (y_i - p_i), \quad \frac{\partial \ell(\sigma)}{\partial \sigma_j} = \sum_i (y_i - p_i) s_{ji}. \tag{2.28}$$

We propose to extend this joint logistic-based calibration to the evidential framework by following the same reasoning as for the single classifier case. The knowledge about $\sigma = (\sigma_0, ..., \sigma_J)$ can be represented by a consonant belief function whose contour function is defined by

$$pl^{\Sigma}_{\mathcal{X}'}(\sigma) = \frac{L_{\mathcal{X}'}(\sigma)}{L_{\mathcal{X}'}(\hat{\sigma})}, \quad \forall \sigma \in \Sigma. \tag{2.29}$$

Furthermore, $pl^{\Theta}_{\mathcal{X}'}$ can be computed from $Pl^{\Sigma}_{\mathcal{X}'}$:

$$pl^{\Theta}_{\mathcal{X}'}(\theta|\mathbf{s}) = \begin{cases} 0 & \text{if } \theta \in \{0,1\}, \\ Pl^{\Sigma}_{\mathcal{X}'}(h_{\mathbf{s}}^{-1}(\theta)) & \text{otherwise,} \end{cases} \tag{2.30}$$

with

$$h_{\mathbf{s}}^{-1}(\theta) = \{(\sigma_0, \sigma_1, ..., \sigma_J) \in \Sigma | h_s(\sigma) = \theta\}, \tag{2.31}$$

$$= \left\{(\sigma_0, \sigma_1, ..., \sigma_J) \in \Sigma | \frac{1}{1+e^{-(\sigma_0 + \sigma_1 s_1 + ... + \sigma_J s_J)}} = \theta\right\}, \tag{2.32}$$

$$= \left\{(\sigma_0, \sigma_1, ..., \sigma_J) \in \Sigma | \sigma_0 = -\ln(\theta^{-1} - 1) - \sigma_1 s_1 - ... - \sigma_J s_J\right\}. \tag{2.33}$$

Thus, the contour function $pl^{\Theta}_{\mathcal{X}'}(\theta|\mathbf{s})$ is defined by

$$pl^{\Theta}_{\mathcal{X}'}(\theta|\mathbf{s}) = \sup_{\sigma_1,...,\sigma_J \in \mathbb{R}} pl^{\Sigma}_{\mathcal{X}'}(-\ln(\theta^{-1} - 1) - \sigma_1 s_1 - \sigma_2 s_2 - ... - \sigma_J s_J, \sigma_1, ..., \sigma_J), \tag{2.34}$$

for all $\theta \in [0,1]$. The vector of parameters $(\sigma_1, \sigma_2, ..., \sigma_J)$ which maximizes $pl^{\Theta}_{\mathcal{X}'}$ can be approximated using gradient ascent as well, with the following partial derivatives:

$$\frac{\partial \ell(\sigma)}{\partial \sigma_j} = \sum_i (y_i - p_i^{''})(s_{ji} - s_j), \quad j = 1, ...J, \tag{2.35}$$

where

$$p_i'' = \frac{1}{1 + e^{-(-\ln(\theta^{-1}-1)-\sigma_1 s_1 - \ldots - \sigma_J s_J + \sigma_1 s_{1i} + \ldots + \sigma_J s_{Ji})}}, \quad \forall \theta \in [0,1]. \tag{2.36}$$

Finally, the belief and plausibility functions $Bel^{\mathbb{Y}}(\cdot|\mathbf{s})$ and $Pl^{\mathbb{Y}}(\cdot|\mathbf{s})$ can be obtained through Eq. (1.38) and (1.39).

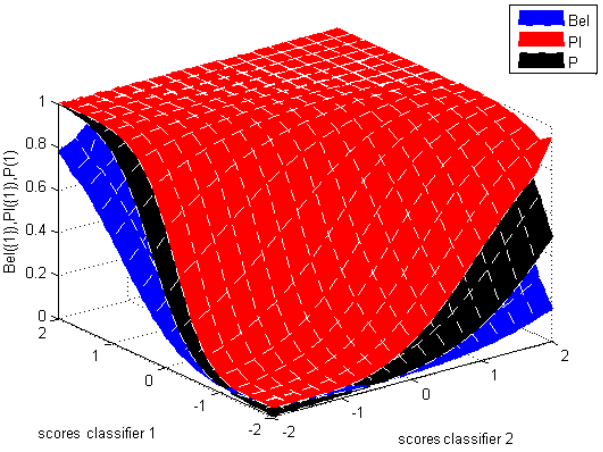
The computational complexity of gradient ascent algorithm is $O(nJ)$ per iteration. Thus, we may notice that the computational complexity is much higher in this joint situation than in the single classifier case. Evaluating the sum-gradient may require expensive computational cost at every iteration, especially when n is large. Within this scope, some techniques have been proposed to speed up the computation. For instance, the use of a dynamic learning rate instead of a fixed one. Indeed, the role of the learning rate is important as if it is too small, the gradient ascent may be very slow, and on the contrary if it is too large, gradient ascent might overshoot the optimum. Thus, to reach the convergence point faster, Barzilai and Borwein [9] proposed to adapt the value of η in each iteration k with:

$$\eta^{(k)} = \frac{(\sigma^{(k)} - \sigma^{(k-1)})^T [\nabla \ell(\sigma^{(k)}) - \nabla \ell(\sigma^{(k-1)})]}{||\nabla \ell(\sigma^{(k)}) - \nabla \ell(\sigma^{(k-1)})||^2}. \tag{2.37}$$

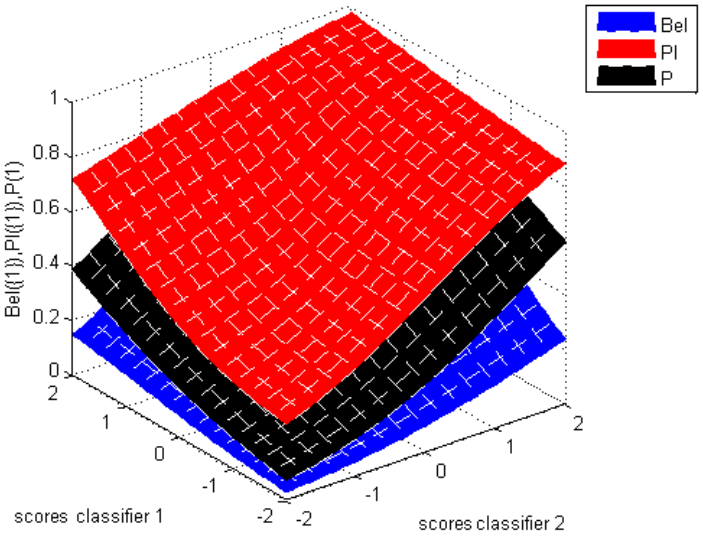
Example 2.3.2 *Figure 2.7 shows an example of an evidential joint calibration based on logistic regression, with $J = 2$ classifiers. This calibration was first trained with 200 examples (Figure 2.7a) then 50 examples (Figure 2.7b). It can be seen that the ignorance is higher in Figure 2.7b than in Figure 2.7a as the three layers are more spaced apart. Furthermore, we can see that, similarly to the joint binning case, the ignorance is high when s_1 is low and s_2 high (or the opposite), and especially when only 50 training examples are taken.*

2.4 Experimental results

In this section, the performance of the proposed evidential joint calibration approach is compared to those of other approaches using different datasets, which are presented in Section 2.4.1. In Section 2.4.2, our approach is compared to the approach of Xu *et al* [116], which consists in transforming the scores provided by different SVM classifiers into belief functions, using the evidential calibration of a single classifier. They are then combined using Dempster's rule of combination. We refer hereafter to this latter approach as the disjoint method. Both binning and logistic regression calibrations are studied. Then, in Section 2.4.3, our joint calibration approaches are compared to a conceptually similar approach, that is a trainable combiner based on an evidential classifier, *i.e.*, a classifier returning a mass function after observing an object. Finally, we focus on the calibration based on multiple logistic regression and



(a) Calibration trained with 200 examples.



(b) Calibration trained with 50 examples.

Figure 2.7 – Illustration of joint calibration based on logistic regression and trained with 200 (Figure 2.7a) and 50 (Figure 2.7b) examples, Diabetes dataset.

we compare the probabilistic and evidential versions of this joint calibration in Section 2.4.4.

2.4.1 Datasets

The experiments are conducted on five binary classification problems provided by UCI repository [5]. They are all of different sizes, and their sample vectors

have various number of features, as presented in Table 2.1.

Dataset	# instance vectors	# features
Australian	690	14
Diabetes	768	8
Heart	270	13
Ionosphere	351	34
Sonar	208	60

Table 2.1 – Number of instance vectors and number of features by vector for different datasets from UCI.

We also simulated a dataset composed of 360 randomly generated instance vectors from two bivariate normal distributions, with means $\mu_0 = (-1, 0)$ in class 0 and $\mu_1 = (1, 1)$ in class 1, and with a covariance matrix equals to $\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$ for both classes. An illustration of these data in the feature space are represented in Figure 2.8, where x and y represent respectively the first and second feature of each instance vector.

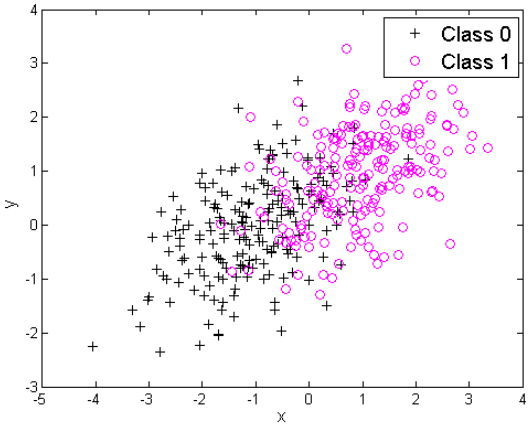


Figure 2.8 – Illustration of 300 instance vectors of the simulated dataset.

2.4.2 Comparison between joint and single calibrations on UCI datasets

The following experiment follows the same protocol as the first experiment detailed in [116]. For each dataset, three SVM classifiers are trained on non-overlapping subsets, using the LIBSVM library [19]. The numbers of examples used for training and testing for each dataset are described in Table 2.2.

Dataset	# Train 1	# Train 2	# Train 3	# Test
Australian	30	70	10-60-190	400
Diabetes	30	70	10-50-200	468
Heart	20	40	10-50-140	70
Ionosphere	20	40	10-80-190	101
Sonar	20	40	10-40-90	58
Simulated data	20	40	10-50-200	100

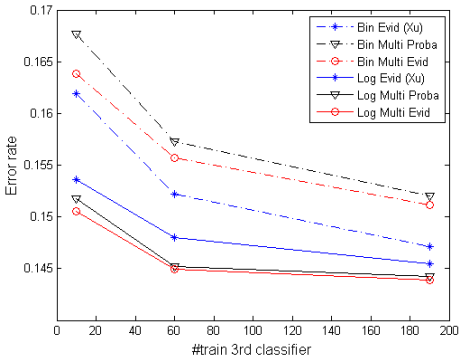
Table 2.2 – Number of examples used for training and testing.

For the first two classifiers, the number of training examples is fixed while different training set sizes are considered for the third one. The training set of each classifier is partitioned into two equal sized-subsets. One of these subsets is for training the classifier, and in Xu *et al.*'s approach the second subset is for training the calibration of the classifier. In the proposed approach, the joint calibration is trained using the set composed of the concatenation of each second subset of each classifier.

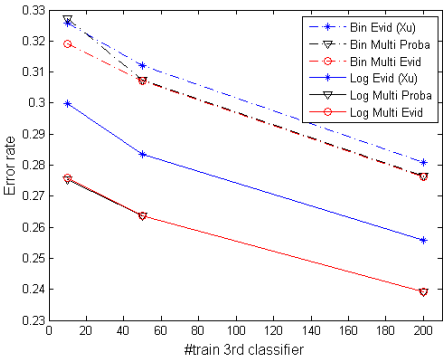
For each sample belonging to the test set, the three classifiers return a score. In the disjoint approach, each of these scores is calibrated using the trained calibration of its corresponding classifier, and the three obtained mass functions are merged into a final mass function using Dempster's rule. In our proposed approach, the scores are grouped into a score vector and this vector is calibrated using a joint calibration, which directly returns a final mass function. In both cases, the decision corresponds to the singleton with the highest belief, since we use $\{0, 1\}$ costs without the possibility to reject, in which case upper and lower expected costs lead to the same decision. The error rate is calculated on the test set and corresponds to the number of samples misclassified over the number of tested samples. The whole process is repeated for 100 rounds of random partitioning, thus the final error rate corresponds to the average of 100 calculated error rates.

For the binning calibration, we may remark that there are in total a number of $B_U \times J$ bins in the disjoint case against $(B_M)^J$ bins for the joint binning. In order to fairly compare our approach to the disjoint one, the number of bins for each classifier is chosen such that each method has the same total number of bins. In particular, as $J = 3$, we chose respectively $B_U = 9$ and $B_M = 3$ for disjoint and joint approaches.

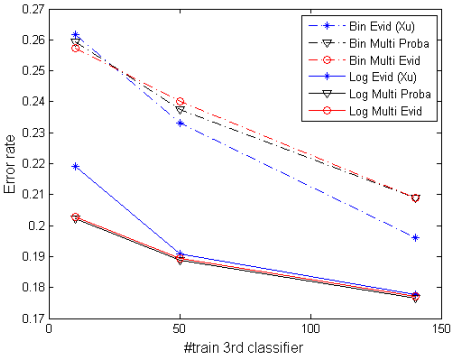
Figure 2.9 shows the results of the experiments for binning and logistic-based approaches, in the evidential framework, and for disjoint and joint cases. Results of the probabilistic version of joint calibrations are also given. As it can be noticed, the approaches based on the logistic regression are always better than those based on binning, as their obtained error rates are lower. For binning approaches, the joint case is not always better than the disjoint case, but it might come from the chosen value for B_M; with a higher value, the results might be better. For logistic regression, the evidential joint approach always presents better results than the evidential disjoint



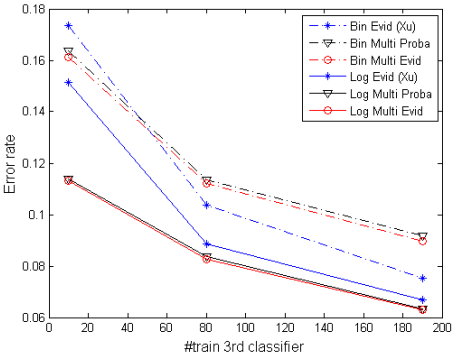
(a) Australian.



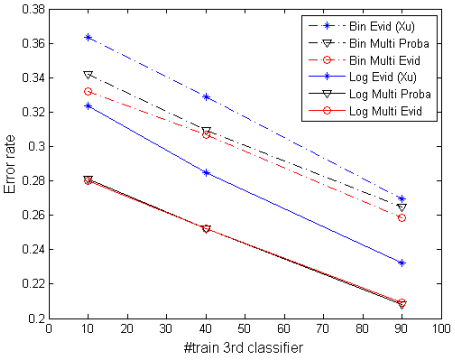
(b) Diabetes.



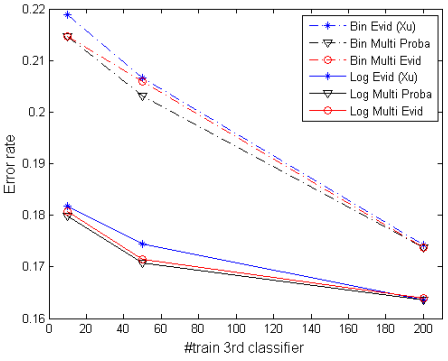
(c) Heart.



(d) Ionosphere.



(e) Sonar.



(f) Simulated data.

Figure 2.9 – Average error rates using binning and logistic regression, with joint (referred to as "multi" in the figures) and disjoint (referred to as "Xu" in the figures) approaches and with both probabilistic and evidential frameworks. The X-axis corresponds to the number of training examples used to train the third classifier.

approach. It can also be noticed that the probabilistic and evidential joint versions nearly give the same results in this experiment. Comparison between probabilistic and evidential versions of calibration based on multiple logistic regression will be performed in Section 2.4.4.

2.4.3 Comparison between evidential joint calibration and evidential trainable combiner on UCI datasets

In the previous experiment, we compared our approach to its probabilistic version and to the so-called disjoint method, which belongs to the non trainable combiner category. In this section, we perform the same experiment but with the aim of comparing our results to those of an approach of the same category, *i.e.*, to an evidential trainable combiner. Indeed, there exist other approaches similar to ours to be compared to, and in particular some methods which can take a score vector as input and return a belief function on the class of a given observed object.

The other evidential trainable combiner that we consider in this experiment relies on the evidential classifier described in [32] and based on the Generalized Bayesian Theorem (GBT) [99].

Let us consider a classification problem with $\Omega = \{w_k\}_{k=1}^{K}$ the finite set of classes. After observing the feature vector $\mathbf{x}$ of an object, the aim is to obtain a belief function about the class label of this object, based on a training set $\mathcal{L} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$ where $\mathbf{x}_i$ represents the feature vector of the i^{th} object, whose true label is y_i. The application of the GBT gives the following MF on Ω about the class of x [32]:

$$m^{\Omega}(A|\mathbf{x}) = \prod_{w_k \in A} Pl[w_k](\mathbf{x}) \prod_{w_k \in \overline{A}} (1 - Pl[w_k](\mathbf{x})), \qquad \forall A \subseteq \Omega, \tag{2.38}$$

where $\overline{A}$ denotes the complement of A, and $Pl[w_k](\mathbf{x})$ represents the plausibility of observing $\mathbf{x}$ under the hypothesis that the true class is ω_k. In particular, Denoeux and Smets have considered in [32] a special case, where

$$Pl[w_k](\mathbf{x}) = \frac{N(\mathbf{x}, k)}{N(k)}, \tag{2.39}$$

with $N(\mathbf{x}, k)$ the number of samples in $\mathcal{L}$ from class w_k contained in a ball S_r of radius r and centered on $\mathbf{x}$, and $N(k)$ the total number of samples from class w_k in $\mathcal{L}$.

We note that it may happen that $m^{\Omega}(\emptyset|\mathbf{x}) > 0$, and in that case the MF $m^{\Omega}(\cdot|\mathbf{x})$ can be transformed into a normalized MF $M^{\Omega}(\cdot|\mathbf{x})$ using the operation defined by

$$M^{\Omega}(A|\mathbf{x}) = \frac{m^{\Omega}(A|\mathbf{x})}{1 - m^{\Omega}(\emptyset|\mathbf{x})}, \qquad \forall A \subseteq \Omega, A \neq \emptyset, \tag{2.40}$$

and $M^{\Omega}(\emptyset|\mathbf{x}) = 0$.

We now apply this classifier to our binary problem, by taking the same inputs as for our approach. In particular, after observing a given object, the feature vector is now the vector of scores $\mathbf{s} = (s_1, ..., s_J)$ obtained by J classifiers, and the training set $\mathcal{L}$ is now $\mathcal{X}'$. Using the definition of the MF given in Eq. (2.38) and the considered particular case of Eq. (2.39), we obtain the MF $m^{\mathbb{Y}}(\cdot|\mathbf{s})$ defined by

$$m^{\mathbb{Y}}(\{0\}|\mathbf{s}) = \frac{N(\mathbf{s},0)}{N(0)} \times (1 - \frac{N(\mathbf{s},1)}{N(1)}), \tag{2.41}$$

$$m^{\mathbb{Y}}(\{1\}|\mathbf{s}) = \frac{N(\mathbf{s},1)}{N(1)} \times (1 - \frac{N(\mathbf{s},0)}{N(0)}), \tag{2.42}$$

$$m^{\mathbb{Y}}(\{0,1\}|\mathbf{s}) = \frac{N(\mathbf{s},1)}{N(1)} \times \frac{N(\mathbf{s},0)}{N(0)}, \tag{2.43}$$

and

$$m^{\mathbb{Y}}(\emptyset|\mathbf{s}) = (1 - \frac{N(\mathbf{s},0)}{N(0)}) \times (1 - \frac{N(\mathbf{s},1)}{N(1)}), \tag{2.44}$$

with $N(\mathbf{s}, k)$ the number of samples in $\mathcal{X}'$ from class k (equal to 0 or 1), contained in a ball S_r of radius r and centered on $\mathbf{s}$. This MF is then normalized similarly as $m^{\Omega}(\cdot|\mathbf{x})$ is normalized using Eq. (2.40).

We may notice that using a ball S_r to build the MFs has some similarities with our multivariable version of binning. Let us illustrate this statement with a simple example, using the dataset Diabetes and with $J = 2$. Figure 2.10 shows the scores returned by two trained classifiers for each sample of a given calibration training set. The X-axis corresponds to the scores given by the first classifier and Y-axis by the second one. A test sample is illustrated by a blue asterisk, and corresponds to $\mathbf{s} = (s_1, s_2)$ the values of the scores returned by the two classifiers. The continuous green lines correspond to the bounds of the joint binning, with $B_M = 3$, and the red circle represents the ball S_r of the GBT-based classifier, with $r = 1$ and centered on $\mathbf{s}$. To build the MF $m^{\mathbb{Y}}(\cdot|\mathbf{s})$, the joint binning uses the training samples belonging to the bin containing $\mathbf{s}$, while the GBT-based classifier uses the ones contained by the ball S_r.

We performed the experiment with $r = 1$, because some preliminary tests showed that the best results were obtained with this value. Figure 2.11 shows the error rates for the GBT-based approach, compared to those obtained with our evidential multivariable versions of binning and logistic regression. As it can be noticed, the results obtained with the GBT-based classifier are better than those obtained with the binning approach. It can be explained by the fact that in the binning approach the bounds of the multi-dimensional bins are fixed, and any test sample belonging to the same multi-dimensional bin has the same associated MF, no matter where the sample is positioned in the bin. By contrast, for the GBT classifier, the ball is centered on the considered test sample, so the neighbourhood of the test sample is taken into account

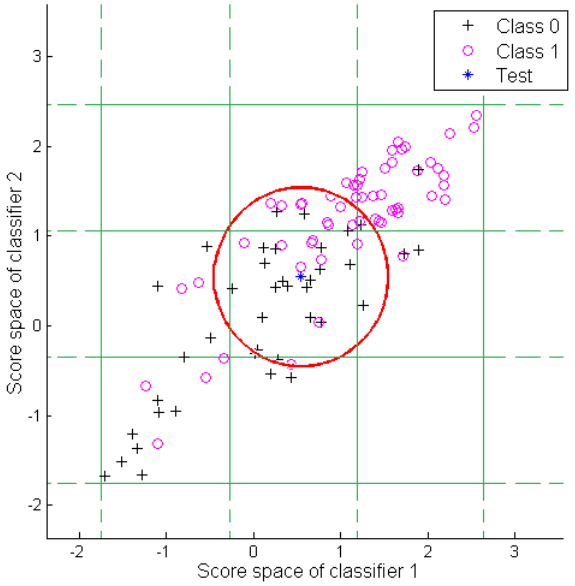


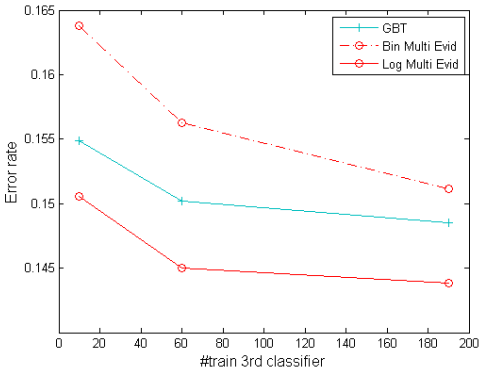
Figure 2.10 – Illustration of the multidimensional bins and the ball S_r, using Diabetes dataset.

in a better way. Furthermore, with other values of r or with other size and number of our multi-dimensional bins, the obtained results may vary significantly, as these approaches highly rely on these parameters. Finally, we can see that the evidential joint calibration using logistic regression is always better than the GBT-based approach in our experiments.

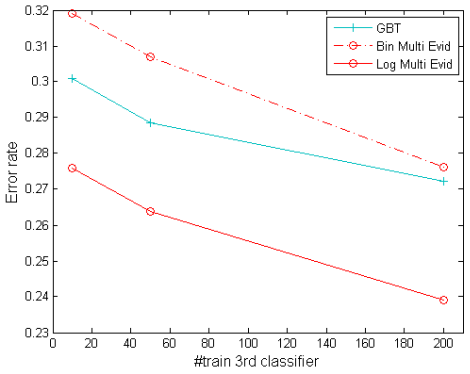
2.4.4 Comparison between evidential and probabilistic versions of joint calibration on UCI datasets

As seen in Sections 2.4.2 and 2.4.3, the evidential joint logistic-based calibration always presents the best results. Yet, we also noted (in Section 2.4.2) that the performance of the probabilistic version of this calibration were nearly the same. Thus, in this section, probabilistic and evidential versions of the calibration based on the multiple logistic regression are further compared. To do that, we introduce the possibility of a third decision for the system given a test sample, by allowing a reject option. Hence, for a given test sample, three possible decisions can be returned: 0, 1, or R. This reject option R expresses doubt and is used for some examples that are difficult to classify. In addition, as recalled in Chapter 1, there are different decision-making criteria in the evidential framework and thus the evidential approach has two possible strategies of decision, either pessimistic or optimistic.

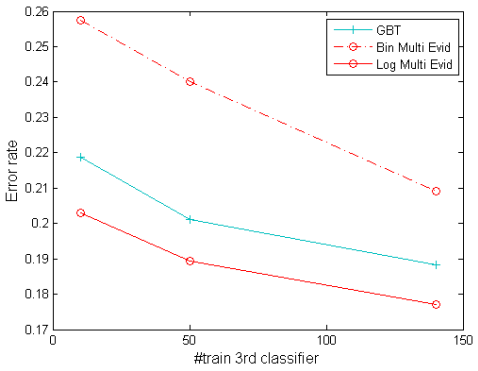
Using the simulated dataset previously defined, 290 training examples were



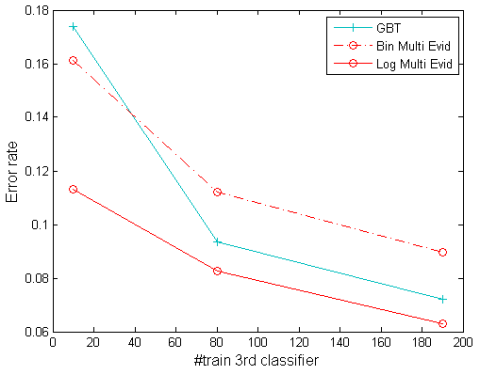
(a) Australian.



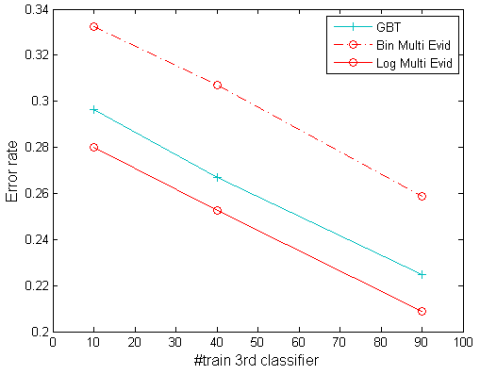
(b) Diabetes.



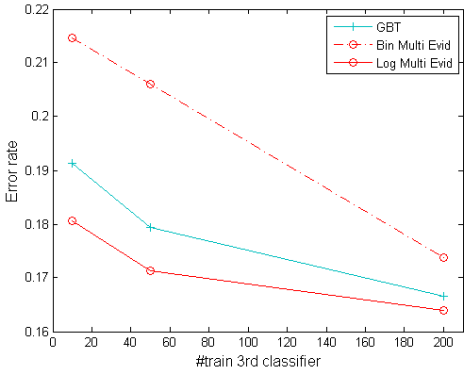
(c) Heart.



(d) Ionosphere.



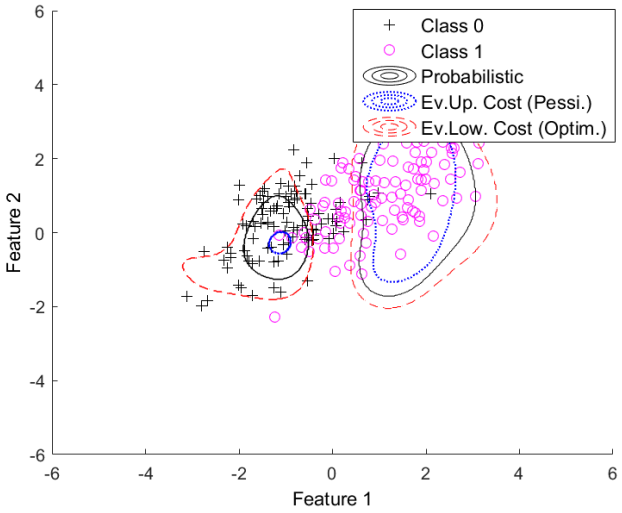
(e) Sonar.



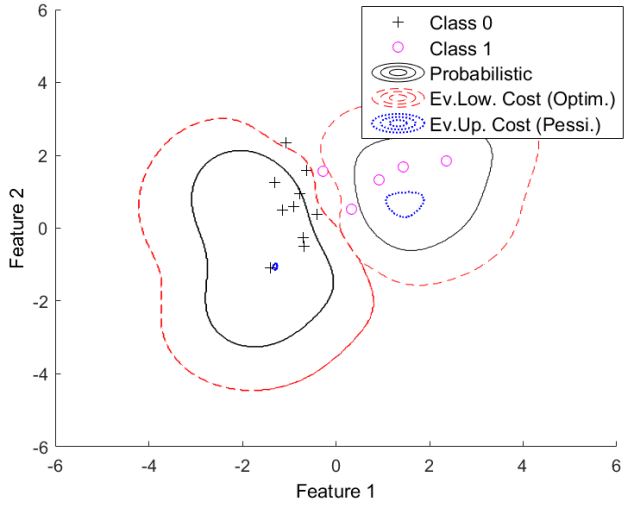
(f) Simulated data.

Figure 2.11 – Average error rates using binning and logistic regression, with evidential joint approaches. The X-axis corresponds to the number of training examples used to train the third classifier.

generated: three SVM classifiers were trained with three non-overlapping subsets of 30 training examples of this set, and the joint calibration using logistic regression was trained with the remaining 200 examples of this set. Then, the same experiment was performed but the joint logistic-based calibration was trained with 15 examples instead of 200. The decision frontiers for both the pessimistic and optimistic strategies and for both cases are illustrated in Figure 2.12 for $R_{rej} = 0.15$.



(a) Joint logistic-based calibration trained with 200 training samples.



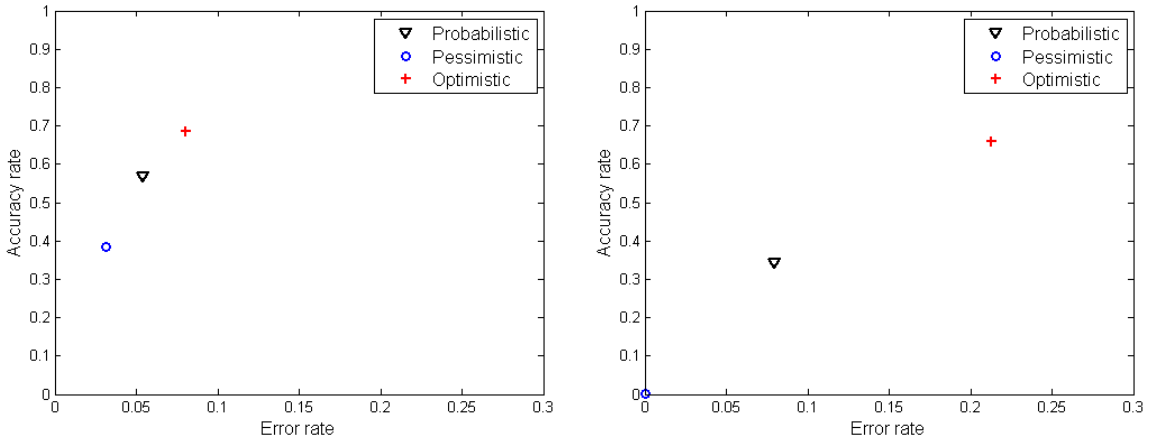
(b) Joint logistic-based calibration trained with 15 training samples.

Figure 2.12 – Decision frontiers in feature space of the probabilistic and evidential joint calibrations based on logistic regression trained with 200 (2.12a) and 15 training examples (2.12b), and with $R_{rej} = 0.15$.

As it can be seen, the evidential joint calibration based on the optimistic

strategy tends to reject less the test samples than the two others. It is the exact opposite for the evidential joint calibration based on the pessimistic strategy, which decide to reject in more cases. The probabilistic approach is between these two. Furthermore, the frontiers associated to the pessimistic and optimistic strategies are a lot more distant from each other in Figure (2.12b) than in Figure (2.12a), *i.e.*, when there are less examples to train the joint calibration and thus more uncertainties. Probabilistic approach is only represented by one frontier so the impact of the uncertainties is not visible. Thus, the evidential approach better reflect the uncertainties than the probabilistic one.

Let us illustrate this point further. The three SVM classifiers were still trained with three non-overlapping subsets of 30 training samples, and the calibration with 200 then 15 samples. We calculated the error rate and accuracy rate for 100 test samples and with $R_{rej} = 0.15$. Accuracy rate represents the number of correctly classified objects over the number of classified objects, *i.e.*, not over the total number of test examples as some of them are rejected. The whole process was repeated for 100 rounds of random partitioning. The obtained average rates are presented in Figure 2.13.



(a) 200 training examples (b) 15 training examples

Figure 2.13 – Obtained error rates for $R_{rej} = 0.15$ and with 200 (2.13a) and 15 (2.13b) training examples.

As it can be seen in Figure 2.13, if there are a lot of examples to train the joint calibration, the obtained error rates are almost equal. Yet, when less training examples are available, the two points obtained for the evidential approach are more distant from each other. This interval reflect the uncertainties, as when it is larger the uncertainties are more important. This information cannot be obtained with the probabilistic calibration, as it is represented by only one point. Thus, the joint calibration based on evidence theory better reflect the uncertainties.

Finally, we performed a similar experiment with R_{rej} varying from 0 to 1, on five datasets (*Australian*, *Diabetes*, *Heart*, *Ionosphere*, *Sonar*) of UCI repository [5] and

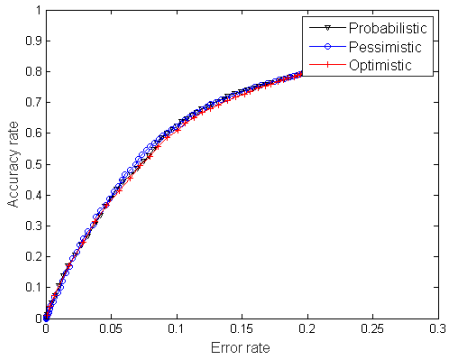
on the simulated dataset. The only difference with the previous experiment is that the multivariable logistic regression was trained with 45 (instead of 200 previously) then 15 samples. Due to the size of *Sonar*, it was tested on 50 sample tests instead of 100 for the other datasets. The whole process was carried out for 100 rounds of random partitioning and Figures 2.14 and 2.15 show the obtained results.

As it can be noticed, for a given error rate, the results obtained with the pessimistic strategy has a higher (or equal) accuracy rate than the probabilistic calibration when few training examples are available (right columns of Figures 2.14 and 2.15). Let us underline that for a fixed error rate, the accuracy rates obtained with the probabilistic calibration and the pessimistic strategy are obtained for different values of R_{rej} (as seen in the previous experiment, the results of which are given in Figure 2.13, a given value of R_{rej} leads in general to different error rates). Furthermore, when the number of training examples is more important (left columns of Figures 2.14 and 2.15), the obtained results become similar for the probabilistic and evidential approaches, as should be.

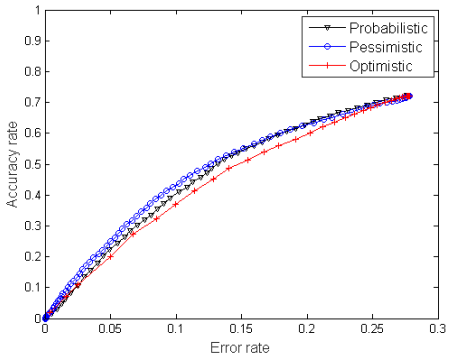
2.5 Conclusion

In this chapter, we have recalled the extension to the evidential framework of the usual calibration techniques proposed by Xu *et al.* in [116]. Given a score returned by a single classifier, these techniques return a belief function. We proposed then to use the multivariable version of the techniques underlying the calibrations and to apply it, in the evidential framework, to the concatenation of the scores returned by multiple classifiers for a given object. Our approach was compared to Xu *et al.*'s disjoint approach, which independently calibrates the scores of SVM classifiers using the evidence theory and combines the obtained mass functions using Dempster's rule of combination. We compared also our proposed method to an approach belonging to the trainable combiner category and based on an evidential classifier. In both cases, the obtained results for our evidential joint calibration based on logistic regression either are better or are comparable to that of the other approaches. Furthermore, by introducing the possibility to reject a test sample, we showed the advantages of the evidential multivariable logistic-based calibration over the probabilistic version: it models more precisely the uncertainties and it exhibits better performances.

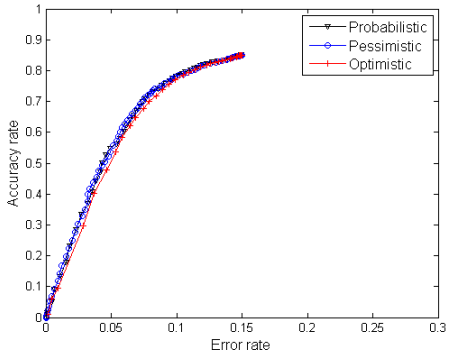
In this first main part of this report, we began by exposing the main concepts of belief function theory, which is a well established formal framework for reasoning with uncertainty. Then, we recalled how this framework has been used to propose evidential extensions of the existing techniques regarding score calibration. We exposed a new approach for combining scores based on this evidential calibration approach but without the use of a rule of combination. The presented approaches were applied in a binary classification problem to the calibration of SVM classifiers, but they may also



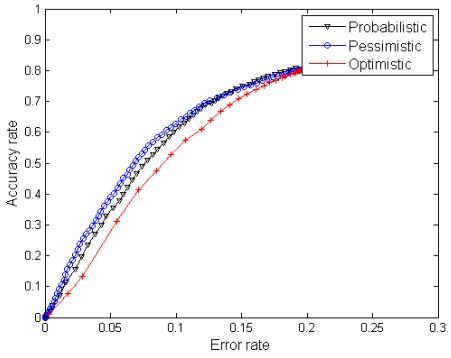
(a) Simulated data – 45 training samples



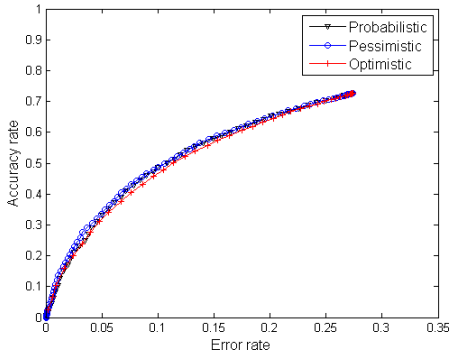
(b) Simulated data – 15 training samples



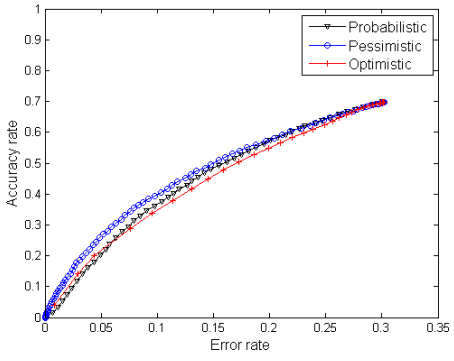
(c) Australian – 45 training samples



(d) Australian – 15 training samples

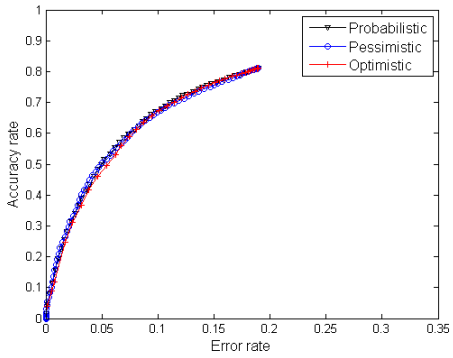


(e) Diabetes – 45 training samples

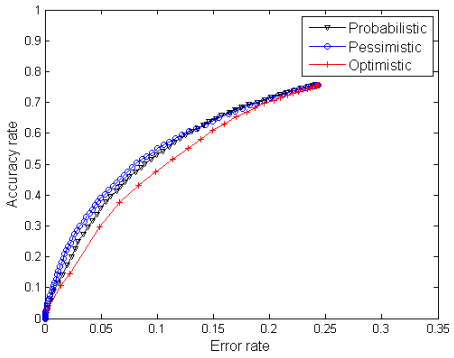


(f) Diabetes – 15 training samples

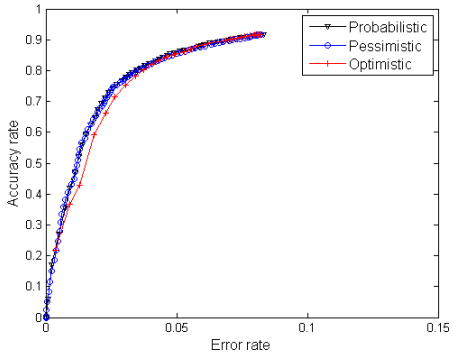
Figure 2.14 – Obtained error rates with 45 training samples (left) and 15 training samples (right) for the simulated dataset, *Australian* and *Diabetes*.



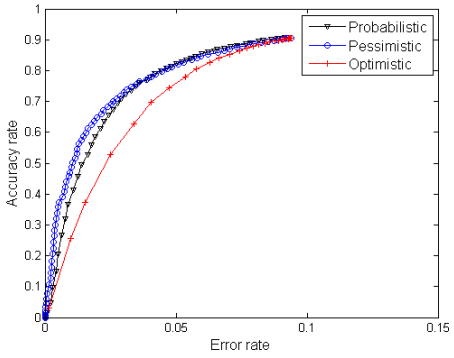
(a) Heart – 45 training samples



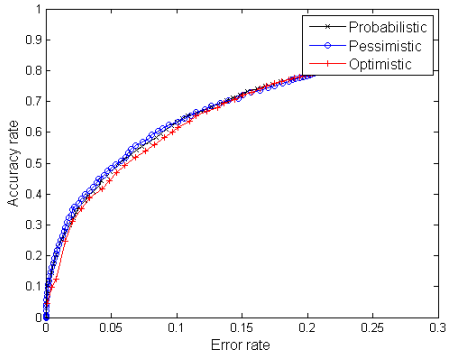
(b) Heart – 15 training samples



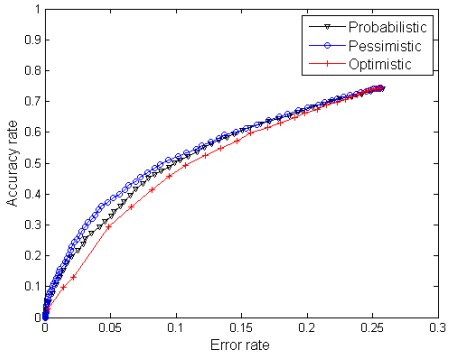
(c) Ionosphere – 45 training samples



(d) Ionosphere – 15 training samples



(e) Sonar – 45 training samples



(f) Sonar – 15 training samples

Figure 2.15 – Obtained error rates with 45 training samples (left) and 15 training samples (right) for *Heart*, *Ionosphere* and *Sonar*.

be applied to any other binary classifiers returning scores. As a matter of fact, in the second part of this report, we will see how these techniques can be applied for the considered issue, *i.e.*, face blurring.

Part II

Application to face blurring

Chapter 3

Pixel-based approach

Contents

3.1 Introduction

Due to legal reasons, faces in a given image may have to be blurred. Yet, it can rapidly become a tedious task if it is done manually, especially if there is a large amount of images to process. A solution may consist in using a face detection system, which aims to automatically find the positions of the faces in a given image.

Since the early 2000s, there has been significant research on face detection and many algorithms have been proposed, in particular based on machine learning techniques, such as the well-known Viola and Jones approach [110] or the neural network-based approach proposed by Rowley *et al.* [90]. Recently, more elaborate algorithms based on deep convolutional neural networks [39, 117, 123] made a major breakthrough in the field. An overview of the state-of-the-art concerning face detection can be found in Appendix A. Yet, another path of research consists in merging information given by multiple sources, whether situated at the pixel level or directly on the faces [1, 41, 73, 104]. Indeed, since sources, such as face detectors, generally provide complementary information, using several of them is a means to improve overall performance.

There are many different ways to perform the fusion of some given information. Among them, in the context of pedestrian detection, Xu *et al.* [114] recently proposed a well-founded and general approach. In this approach, for a given image each used detector provides a set of bounding boxes corresponding to the assumed positions of the pedestrians, as well as a confidence score for each of these boxes. The main idea is then to use score calibration in order to be able to combine these calibrated scores afterwards, and to obtain better detection performance. Of particular interest is that the combination of this approach relies on evidence theory. Hence, by replacing the calibration procedure in [114] by the evidential ones, and by applying to faces the general detection approach introduced in [114], one obtains what may be considered presently as a state-of-the-art face detection system based on multiple detectors. Nonetheless, despite its appeals, we note that such a system suffers from two main limitations inherited from Xu *et al.*'s approach [114]. First, it is designed to handle only detectors providing bounding boxes, *i.e.*, it can not integrate directly sources providing information at the pixel level. Second, this approach relies on a parameter (so-called overlap threshold) necessary in the handling of boxes.

Using a face detection system is a natural means to solve the face blurring problem. However, we may remark that this problem is not exactly equivalent to face detection: face blurring amounts merely to deciding whether a given pixel belongs to a face, whereas face detection amounts to determining whether a given set of pixels corresponds to the same face. This remark opens the path for a different approach to reasoning about blurring, which may then be situated at the pixel-level. Within this scope, we propose in this chapter a face blurring system, which consists essentially in applying at the pixel-level the central idea and contributions of Xu *et al.* [114, 116], *i.e.*,

combining evidentially calibrated information sources. As it will be seen, this pixel-level perspective presents several conceptual advantages over operating at the box-level. In particular, sources providing pixel-level information can be directly integrated and the parameter necessary in the handling of boxes can be avoided.

This chapter is organized as follows. Section 3.2 exposes the system performing face detection using Xu *et al.*'s evidential box-based detection approach [114], improved using Xu *et al.*'s evidential calibration [116], *i.e.*, calibration exposed in Chapter 2. In Section 3.3, our proposed pixel-based face blurring system is detailed and its fundamental differences with respect to blurring using Xu *et al.*'s box-based approach are discussed. The performances of the box-based and pixel-based approaches, given the same input information, are then compared in Section 3.5.2 on two image datasets (one from the literature and one composed of railway platforms images coming from SNCF). The ability of the proposed approach to integrate directly pixel level information is illustrated in Section 3.5.3 on these same two datasets with a classical feature regarding face detection. We detail in Section 3.4 how the evidential joint approach of calibration that we described in Chapter 2 can be applied to the context of face detection. In particular, Section 3.5.4 compares the results of disjoint and the joint pixel-based approaches on the two datasets. Finally, conclusions and perspectives are given in Section 3.6.

3.2 An evidential box-based face detection approach

Face blurring may be achieved using simply the boxes returned by a face detection system. In this section, we present such a system, which may be considered as a state-of-the-art system with respect to face detection based on multiple detectors returning box information and using the evidential framework. In a nutshell, this system is merely Xu *et al.* [114] evidential box-based detection approach, whose calibration step has been replaced by the evidential likelihood-based logistic regression calibration procedure proposed in [116] and recalled in the Chapter 2 of this report. This section provides first an overview of this approach and then details some of its steps.

3.2.1 Overview of the approach

Let us consider a given image and assume that J face detectors are run on this image. Formally, each detector D_j, $j = 1, ..., J$, provides N_j couples $(B_{i,j}, S_{i,j})$, where $B_{i,j}$ denotes the i^{th} box, $i = 1, ..., N_j$, returned by the j^{th} detector and $S_{i,j}$ is the confidence score associated to this box.

Through a calibration procedure using a training set which will be described in Section 3.2.2, score $S_{i,j}$ is transformed into a MF $m^{\mathcal{B}_{i,j}}$ defined over the frame $\mathcal{B}_{i,j} = \{0, 1\}$, where 1 (resp. 0) means that there is a face (resp. no face) in box $B_{i,j}$.

Then, using a clustering procedure detailed in Section 3.2.3, all the boxes $B_{i,j}$ returned by the J detectors for the considered image, are grouped into K clusters C_k, $k = 1, ..., K$, each of these clusters being represented by a single box B_k.

In addition, for each box $B_{i,j} \in C_k$, its associated MF $m^{\mathcal{B}_{i,j}}$ is assumed to represent a piece of evidence regarding the presence of a face in B_k, that is, $m^{\mathcal{B}_{i,j}}$ is converted into a MF $m_{i,j}^{\mathcal{B}_k}$ on $\mathcal{B}_k = \{0, 1\}$ defined by $m_{i,j}^{\mathcal{B}_k}(A) = m^{\mathcal{B}_{i,j}}(A)$, for all $A \subseteq \{0, 1\}$. These pieces of evidence are then combined using Dempster's rule:

$$m^{\mathcal{B}_k} = \bigoplus_{i,j} m_{i,j}^{\mathcal{B}_k}. \tag{3.1}$$

The combination results in a MF $m^{\mathcal{B}_k}$ representing the overall system uncertainty with respect to the presence of a face in B_k. We note that the use of Dempster's rule is appropriate when the sources may be considered to be independent and reliable. More complex combination schemes are also considered in [114]. However, only Dempster's rule, which presents good performance in [114], is considered here.

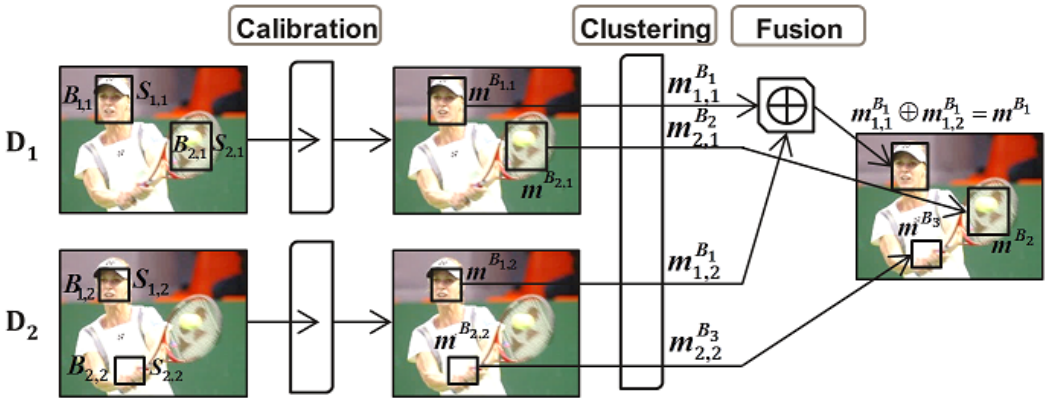


Figure 3.1 – Illustration of the box-based approach

The three main steps of the approach, namely calibration, clustering and fusion, are illustrated in Figure 3.1. For the sake of simplicity only two detectors, each returning two boxes, are considered in this example. $B_{i,j}$ corresponds to the i^{th} box, $i = 1, 2$, returned by the j^{th} detector $j = 1, 2$, and which has $S_{i,j}$ as associated score. In this scenario, the boxes $B_{1,1}$ and $B_{1,2}$ are grouped into the same cluster C_1, represented by the box B_1. Their associated scores, transformed into mass functions, are combined and result in the final mass function $m_{1,1}^{\mathcal{B}_1} \oplus m_{1,2}^{\mathcal{B}_1}$, which is denoted by $m^{\mathcal{B}_1}$. The other boxes $B_{2,1}$ and $B_{2,2}$ form their own clusters, respectively represented

by B_2 and B_3. Finally, for each resulting box with its associated MF, a decision has to be made whether the box has to be blurred or not; it may be done using one of the decision strategy given in Chapter 1 and in particular using Eq. (1.14) for some cost function c.

3.2.2 Box-based score calibration for a detector

In order to transform the score $S_{i,j}$ associated to a box $B_{i,j}$ into a MF $m^{\mathcal{B}_{i,j}}$, detector D_j needs to be calibrated. In particular, the evidential logistic regression calibration procedure recalled in Chapter 2 may be used instead of the cruder procedures used in [114]. This procedure requires a training set, which we denote by $\mathcal{L}_{cal,j}$. We detail below how $\mathcal{L}_{cal,j}$ is built.

Assume that L images are available. Besides, the positions of the faces really present in each of these images are known in the form of bounding boxes. Formally, this means that for a given image ℓ, a set of M^ℓ boxes $G_r^\ell, r = 1, ..., M^\ell$, is available, with G_r^ℓ the r^{th} bounding (ground truth) box on image ℓ.

Furthermore, detector D_j to be calibrated is run on each of these images, yielding N_j^ℓ couples $(B_{t,j}^\ell, S_{t,j}^\ell)$ for each image ℓ, where $B_{t,j}^\ell$ denotes the t^{th} box, $t = 1, ..., N_j^\ell$, returned on image ℓ by detector D_j and $S_{t,j}^\ell$ is the confidence score associated to this box.

From these data, training set $\mathcal{L}_{cal,j}$ is defined as the set of couples $(S_{t,j}^\ell, YB_{t,j}^\ell)$, $\ell = 1, ..., L$, and $t = 1, ..., N_j^\ell$, with $YB_{t,j}^\ell \in \{0, 1\}$ the label obtained by evaluating whether box $B_{t,j}^\ell$ "matches" some face in image ℓ, *i.e.*,

$$YB_{t,j}^\ell = \begin{cases} 1 & \text{if } \exists G_r^\ell, r = 1, ..., M^\ell, \text{ such that } ov(G_r^\ell, B_{t,j}^\ell) \geq \lambda, \\ 0 & \text{otherwise,} \end{cases}$$

where λ is some threshold in $(0, 1)$ and $ov(G_r^\ell, B_{t,j}^\ell)$ is a measure of the overlap between boxes G_r^ℓ and $B_{t,j}^\ell$. It is defined by [37]

$$ov(B_1, B_2) = \frac{area(B_1 \cap B_2)}{area(B_1 \cup B_2)}, \tag{3.2}$$

for any two boxes B_1 and B_2. Informally, $\mathcal{L}_{cal,j}$ stores the scores associated to all the boxes returned by detector D_j on images where the positions of faces are known, and records for each score whether its associated box is a true or false positive. It is then clear that the MF $m^{\mathcal{B}_{i,j}}$ associated to a new score $S_{i,j}$ and obtained from calibration relying on $\mathcal{L}_{cal,j}$, represents uncertainty toward box $B_{i,j}$ containing a face.

3.2.3 Clustering of boxes

As several detectors are used, some boxes may be located in the same area of an image, which means that different boxes assume that there is a face in this particular area. The step of clustering allows one to group those boxes and to retain only one per cluster. A greedy approach is used in [114], based on the work of Dollar *et al.* [34]: the procedure starts by selecting the box $B_{i,j}$ with the highest mass of belief on the face hypothesis, *i.e.,* the box $B_{i,j}$ such that $m^{\mathcal{B}_{i,j}}(\{1\}) > m^{\mathcal{B}_{u,v}}(\{1\}), \forall (u,v) \neq (i,j)$, and this box is considered as the representative of the first cluster. Then, each box $B_{u,v}$, $\forall (u,v) \neq (i,j)$, such that the overlap $ov(B_{i,j}, B_{u,v})$ is above the threshold λ, is grouped into the same cluster as $B_{i,j}$, and is then no longer considered for further associations. Among the remaining boxes, the box $B_{i,j}$ with the highest $m^{\mathcal{B}_{i,j}}(\{1\})$ is selected as representative of the next cluster, and the procedure is repeated until all the boxes are clustered.

3.3 Evidential pixel-based approach

The approach exposed in the previous section is general and well-founded. It is designed for detectors returning boxes, but it does not allow to directly integrate pixel-based information. Besides, as explained in the introduction of this chapter, for the purpose of blurring it seems interesting to work at the pixel level rather than box level. Thus, the idea of the approach proposed in this section is to use elements from the previous system, in particular the evidential calibration and fusion, and to apply them at the pixel level. This section first exposes an overview of the proposed approach. Then, in order to be able to compare subsequently the proposed pixel-based approach to the previous system, we detail how the same input information as in the previous section, *i.e.*, boxes and scores returned by detectors, can be used within our pixel-based approach. Finally, fundamental differences between the two approaches are discussed.

3.3.1 Overview of the approach

To each pixel $p_{x,y}$ in an image, we associate a frame of discernment $\mathcal{P}_{x,y} = \{0,1\}$, where x and y are the coordinates of the pixel in the image and 1 (resp. 0) means that there is a face (resp. no face) in pixel $p_{x,y}$. For the pixel $p_{x,y}$, J mass functions are obtained on $\mathcal{P}_{x,y}$ from J detectors. They are then combined using Dempster's rule of combination, resulting in the MF denoted $m^{\mathcal{P}_{x,y}}$, *i.e*,

$$m^{\mathcal{P}_{x,y}} = \bigoplus_{k=1}^{J} m_k^{\mathcal{P}_{x,y}}, \tag{3.3}$$

with $m_k^{\mathcal{P}_{x,y}}$ the MF representing the uncertainty with respect to the presence of a face in the pixel $p_{x,y}$ for the k^{th} source. Each MF $m_k^{\mathcal{P}_{x,y}}, k = 1, \ldots, J$, is obtained using the calibration method corresponding to the type of the outputs of the k^{th} source. Specifically, if the source gives a score information, the MF is obtained through the evidential logistic regression calibration, using a training set $\mathcal{L}$ composed of couples (X_i, Y_i), with X_i the score associated to the i^{th} object which is now a pixel, and Y_i its true label. Otherwise, if the source does not return a score, we propose to calibrate this information of score absence; this is further explain in the next section.

3.3.2 Face detection as input to our approach

Consider strictly the same input information as in Section 3.2, that is J detectors each returning a set of bounding boxes with associated scores corresponding to the assumed positions of the faces. This section exposes how our approach can be applied in that case.

For a given pixel in an image and a given detector, two exclusive situations occur: either the pixel $p_{x,y}$ is contained by one of the box $B_{i,j}$ returned by the detector, or it is not. If it is contained by a box $B_{i,j}$, the score $S_{i,j}$ of the box is associated ("transferred") to the pixel. If the pixel does not belong to any box, no score is associated to it. As a consequence, the considered pixel either has an associated score, or it does not. These two situations are now detailed.

In the first case, when a score is available for the considered pixel, it is transformed into a MF using the evidential logistic regression and a training set, that we denote $\mathcal{L}_{calP,j}$. Let us describe this set $\mathcal{L}_{calP,j}$ underlying the transformation using calibration of a score $S_{i,j}$ associated to a pixel $p_{x,y}$ by a detector D_j, into a MF $m_{i,j}^{\mathcal{P}_{x,y}}$. For a given image ℓ, each couple $(B_{t,j}^{\ell}, S_{t,j}^{\ell})$ introduced in Section 3.2.2 yields, *via* "transfer", $\left|B_{t,j}^{\ell}\right|$ couples $(p_{d,t,j}^{\ell}, S_{t,j}^{\ell})$, with $d = 1, \ldots, \left|B_{t,j}^{\ell}\right|$, and $\left|B_{t,j}^{\ell}\right|$ the number of pixels in box $B_{t,j}^{\ell}$, and where $p_{d,t,j}^{\ell}$ denotes the pixel in d^{th} position in box $B_{t,j}^{\ell}$. From these data, we define $\mathcal{L}_{calP,j}$ as the set of couples $(S_{t,j}^{\ell}, YP_{d,t,j}^{\ell})$, with $\ell = 1, \ldots, L$, $t = 1, \ldots, N_j^{\ell}$, and $d = 1, \ldots, \left|B_{t,j}^{\ell}\right|$, with $YP_{d,t,j}^{\ell} \in \{0, 1\}$ the label simply obtained by checking whether pixel $p_{d,t,j}^{\ell}$ belongs to some ground truth box G_r^{ℓ} in the image ℓ, *i.e*,

$$YP_{d,t,j}^{\ell} = \begin{cases} 1 & \text{if } \exists\ G_r^{\ell},\ r = 1, \ldots, M^{\ell}, \text{ such that } p_{d,t,j}^{\ell} \in G_r^{\ell}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

$\mathcal{L}_{calP,j}$ may pose a complexity issue as $|\mathcal{L}_{calP,j}| = \sum_{\ell=1}^{L} \sum_{t=1}^{N_j^{\ell}} \left|B_{t,j}^{\ell}\right|$. To avoid this, one may use a smaller set $\mathcal{L}'_{calP,j} \subset \mathcal{L}_{calP,j}$, which represents roughly the same information as $\mathcal{L}_{calP,j}$ and built as follows: for each triple (ℓ, t, j), only 10 couples among the couples

$(S^{\ell}_{t,j}, YP^{\ell}_{d,t,j})$, $d = 1, \ldots, \left|B^{\ell}_{t,j}\right|$, are selected such that the ratio

$$\frac{\left|\{YP^{\ell}_{d,t,j} | d = 1, ..., \left|B^{\ell}_{t,j}\right|, YP^{\ell}_{d,t,j} = 1\}\right|}{\left|\{YP^{\ell}_{d,t,j} | d = 1, ..., \left|B^{\ell}_{t,j}\right|, YP^{\ell}_{d,t,j} = 0\}\right|} \tag{3.5}$$

is preserved. $\mathcal{L}'_{calP,j}$ has then a size of $\left|\mathcal{L}'_{calP,j}\right| = 10 \sum_{\ell=1}^{L} N^{\ell}_{j}$.

Let us now consider the second situation, where a pixel is not contained by any of the boxes and thus does not have an associated score. Since it should be taken into account that detectors do not present the exact same performances (in particular, some may have many more pixels not in boxes than others), it seems interesting to calibrate this kind of outputs from detectors, *i.e.*, we propose to calibrate the information of score absence. Specifically, the training set, denoted $\mathcal{L}_{*,j}$, necessary for this calibration is obtained using L images on which the detector D_j is applied. The number n_j of pixels of these images, which are not contained by any of the boxes returned by the detector D_j, can be obtained. As the ground truth of these L images is known, their associated true label Y_i is available. Using $\mathcal{L}_{*,j}$, it is then possible to obtain a MF, denoted $m^{\mathcal{P}_{x,y}}_{*,j}$, and representing the uncertainty with respect to the presence of a face on pixel $p_{x,y}$ when this pixel is not included in a box of detector D_j. Specifically, if we denote by TN (True Negative) the number of pixels correctly classified on these images as non-face and FN (False Negative) the number of pixels classified as non-face but actually belonging to a face, the MF $m^{\mathcal{P}_{x,y}}_{*,j}$ can be defined by

$$m^{\mathcal{P}_{x,y}}_{*,j}(\{0\}) = \frac{TN}{TN + FN + 1}, \quad m^{\mathcal{P}_{x,y}}_{*,j}(\{1\}) = \frac{FN}{TN + FN + 1}. \tag{3.6}$$

Equation 3.6 may be seen as the binning calibration extended to the evidential framework using the model of Dempster [25].

3.3.3 Comparison of both approaches

The proposed pixel-based approach presents several advantages over the one of Section 3.2. First, as can be seen in Section 3.3.2, the construction of the training set for calibration in case of pixels avoids the use of the parameter λ, whose value needs to be fixed either *a priori* (but then it is arguably arbitrary) or empirically.

Furthermore, our approach avoids the use of the clustering step, which also involves the parameter λ and that may behave non optimally in a multi-object situation, especially when they are close to each other, which may be the case with faces in a crowd.

In addition, it allows us to have an arguably more consistent modelling of box absence than the box-based method. Indeed, in this latter method, for a given

area in an image, there are two different modellings of box absence depending on the situation: either none of the detectors has provided a box, in which case the area is considered as non face, which amounts to considering that the detectors know that there is no face; or only a subset of the detectors has provided a box, in which case the other detectors are ignored, which is equivalent (under Dempster's rule) to considering that these detectors know nothing. By contrast, in the proposed method, the use of calibration enables us to take into account in a consistent manner the information of score absence into the fusion process, as when a detector D_j does not return a box for a given pixel $p_{x,y}$, its associated MF $m_{*,j}^{\mathcal{P}_{x,y}}$ is considered regardless of the outputs of the other detectors for this pixel. Thus, all detectors are involved in each fusion. Figure 3.2 illustrates this point, highlighting the differences with the previous approach.

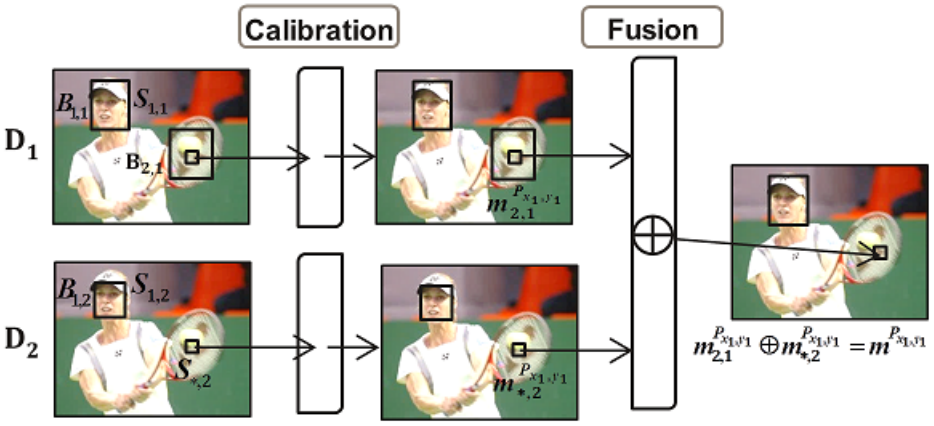


Figure 3.2 – Illustration of the pixel-based approach

For the sake of simplicity only one pixel, at the position (x_1, y_1), is considered here. Pixel p_{x_1,y_1} is contained by the box $B_{2,1}$, with $S_{2,1}$ as associated score, so the corresponding mass function is obtained through the evidential logistic regression. However, there is no box containing p_{x_1,y_1} for the second detector, and thus it does not have an associated score. Yet, the opinion of the second detector is still taken into account *via* the MF $m_{*,2}^{\mathcal{P}_{x_1,y_1}}$ defined in Section 3.3.2.

As explained before, one of the disadvantages of the box-based approach is that the integration of a pixel-based information is not straightforward. In the proposed system however, a source of information which gives pixel-based information can be integrated into the fusion process as easily as a box-based information. It will be illustrated with an experiment in Section 3.5.3.

Nonetheless, let us note that while our approach presents some interests over box-based methods for the problem of face blurring, these latter methods provide more information (specifically, they isolate faces) and are thus relevant for other problems, such as face recognition. Furthermore, we note that locating the approach at the pixel level brings potentially a complexity issue. This will be discussed in Section 3.5.

3.4 Joint evidential pixel-based approach

Both approaches presented in the previous sections, either the box-based one or the proposed pixel-based one, have a common point: they independently calibrate the scores returned by the classifiers before combining them. In the second chapter of this report, we have presented an evidential joint calibration that we tested on UCI datasets and which has presented interesting results. Thus, in this section we propose to apply at a pixel-level this approach with face detection as inputs, *i.e.*, bounding boxes and scores returned by detectors.

3.4.1 Overview of the approach

As for the approach of section 3.3, to each pixel $p_{x,y}$ in an image we associate a frame of discernment $\mathcal{P}_{x,y} = \{0, 1\}$, where x and y are the coordinates of the pixel in the image and 1 (resp. 0) means that there is a face (resp. no face) in pixel $p_{x,y}$. For the pixel $p_{x,y}$, only one mass function $m^{\mathcal{P}_{x,y}}$ is obtained on $\mathcal{P}_{x,y}$ regardless of the number J of independent detectors, with $m^{\mathcal{P}_{x,y}}$ the MF representing the uncertainty with respect to the presence of a face in the pixel $p_{x,y}$. This MF is obtained using the evidential joint logistic regression calibration with as input the concatenation of all the outputs returned by the J detectors. The step of fusion using a predetermined rule is no longer necessary. The set necessary to train the joint calibration is composed by $\mathcal{L}_2 = \{(X_{11}, X_{12}, ...X_{1j}, Y_1), ..., (X_{n1}, X_{n2}, ...X_{nj}, Y_n)\}$, where X_{nj} the output associated to the n^{th} object returned by the j^{th} detector, and Y_n its true label. This training set can be built using the same reasoning than in the disjoint case, *i.e.*, by running the J detectors on L annotated images.

3.4.2 Face detection as input to our approach

We still consider the same input as in Sections 3.2 and 3.3, *i.e.*, J detectors returning bounding boxes with associated scores. For a given pixel in an image and a given detector, there are only two possibilities: either the pixel $p_{x,y}$ is contained by a box $B_{i,j}$ returned by the detector, or it is not. If it is contained by a box $B_{i,j}$, the score $S_{i,j}$ is available, but if the pixel does not belong to any box, no score is associated to it. In that case, we decided to interpret the score absence as a score having a very low value, as a score is necessary for the logistic regression. When a detector returns a score, even the smallest one is strictly superior to zero, thus we arbitrarily chose $S_{*,J} = 0$. The concatenation of the J scores, that are either values or null, is then used as input of the joint calibration, which then gives a corresponding MF for each pixel that is at least contained by one box.

Figure 3.3 illustrates this approach in a very simple case. For the sake of

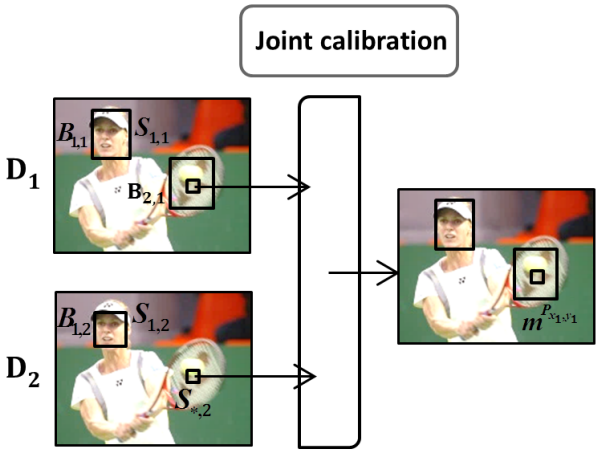


Figure 3.3 – Illustration of the pixel-based approach

simplicity only one pixel, at the position (x_1, y_1), is considered here. Pixel p_{x_1,y_1} is contained by the box $B_{2,1}$, with $S_{2,1}$ as associated score. However, there is no box containing p_{x_1,y_1} for the second detector, and thus it does not have an associated score. As we only kept the positive values of scores of all the outputs of our detectors, we attribute the value zero as score to this detector. Thus, the input for the joint calibration is $(S_{2,1}, 0)$ and the output is the MF $m^{\mathcal{P}_{x,y}}$.

One of the disadvantage of that approach is that it is more difficult to add a classifier as all the joint calibration needs to be re-trained, *i.e.*, a new dataset $\mathcal{L}_2$ needs to be entirely rebuilt while in the disjoint case only the calibration of the considered new classifier has to be trained. Let us note that a common point between these joint and disjoint approaches is that all detectors are involved in each fusion, whether it returns a box for the considered pixel or not. Yet, the particular case of score absence is not taken into account in the same way; in the disjoint case, this information is calibrated, *i.e.*, we learn the MF that represents this information, and then we merged it with the other MFs, while in the joint case we consider that it corresponds to a low score and it is directly integrated in the calibration process. Furthermore, the use of a joint calibration enables us to take into account in a consistent manner the relation between the different detectors. For instance, the joint calibration may learn that a given detector a and detector b do not often give boxes on the same area, but when they do, it is very likely to actually correspond to a face.

3.5 Experimental results

In this section, the results of the proposed pixel-based approach are presented and compared to those of the box-based method, when all available inputs are box-based information. The experiment is performed on a literature dataset as well as on another dataset, composed of images coming from cameras filming railway platforms. The experiment is first described, then the results are discussed. Then, a classical pixel-based information is added to the system. Finally, our proposed disjoint pixel-based approach is compared to the joint one.

3.5.1 Description

An overview of the state-of-the-art regarding face detection is given in Appendix A. We selected four classical face detectors following their popularity and their availability on open source code. The first selected detector is the one proposed by Viola and Jones [110], which is based on the classification algorithm called Gentle Adaboost and that uses Haar feature extraction. The second detector is a variant of the previous one: the same classification algorithm is used but with Local Binary Patterns (LBP) feature extraction [47]. They are both provided by the library OpenCV [13]. Furthermore, an improved HOG+SVM-based algorithm provided by DLIB library [65] was also selected. It is actually the HOG+SVM approach of Dalal and Triggs [22], where the version of HOG features method was replaced by the one exposed in [42]. Finally, the fourth and last selected detector was a deep neural network classifier recently proposed in [59]. It is based on a compact design of a convolutional neural network and a cascade approach and aims to have a reasonable time processing.

We used a literature dataset called Face Detection Data Set and Benchmark (FDDB) [54], which contains the annotations (ground truth) for 5171 faces in a set of 2845 images, in order to train both Adaboost-based detector with the same 2000 images of this dataset. 200 other images were used for the calibration of the four detectors. The performances of the box-based and pixel-based approaches were then evaluated over the remaining 645 images. The third detector, *i.e.*, the HOG+SVM approach provided by DLIB library, was pre-trained by the authors using the dataset called "Labelled Faces in the Wild" [53]. The DNN-based detector was also trained by the authors using the "YouTube Faces Database" [113].

We also created a dataset of 600 images that we extracted from videos provided by the EAS system. These images, which we refer to as SNCF images, contain multiple different conditions such as indoor and outdoor environment, different light settings and low image quality. This is thus a more challenging dataset than FDDB. The true positions of the 1089 faces on these images have been manually annotated. We used the same detectors as for FDDB experiment, *i.e.*, trained with FDDB faces

or other datasets. Let us note that training these algorithms with face images taken from the SNCF dataset would lead in principle to better detection rates, but we did not have enough available annotated faces. Nonetheless, we calibrated these detectors using 100 annotated SNCF images. Performance tests were then conducted over the remaining 500 images.

Figure 3.4 shows an example of bounding boxes and associated scores returned by the four selected detectors on some images extracted from SNCF videos. The Haar+Adaboost detector is represented in red, the LBP+Adaboost detector in yellow, the HOG+SVM in green and finally the DNN in blue. In compliance with the confidentiality requirements, only SNCF employees are present in these images.

As seen in the previous section, the box-based approach returns MFs associated to boxes while our approach gives an MF for each pixel. Whatever the approach, to decide if a given pixel or a given box has to be blurred or not, we use the decision procedure relying on upper expected costs recalled in Chapter 1; in a binary case, they are simply defined by

$$R^*(\{0\}) = m^\Omega(\{1\})c(0,1) + m^\Omega(\{0,1\})c(0,1), \tag{3.7}$$

$$R^*(\{1\}) = m^\Omega(\{0\})c(1,0) + m^\Omega(\{0,1\})c(1,0), \tag{3.8}$$

by considering that the cost is equal to zero when the answer is correct ($c(0,0) = c(1,1) = 0$). As our purpose is to minimize the number of non-blurred faces, it is worse to consider a face as non-face than the opposite. In other words, decisions were made with costs such that $c(1,0) <= c(0,1)$. More specifically, we fixed $c(1,0) = 1$ and gradually increased $c(0,1)$ starting from $c(0,1) = 1$, to obtain different performance points. To quantify performances, we used the recall rate (proportion of pixels correctly blurred among the pixels to be blurred) and the precision rate (proportion of pixels correctly blurred among blurred pixels).

3.5.2 Comparison between box-based and pixel-based approaches on FDDB and SNCF databases

Figure 3.5 compares the results of the four selected detectors taken alone to that of our approach relying on a combination of their outputs, on the FDDB dataset. As it can be seen, the fusion of the four detectors outputs considerably increased the performances, as for example a precision of 80% gives a recall of around 52% for the Haar/Adaboost detector instead of 77% for the combination result. Let us note that the performances of the deep neural network face detector are only represented by a point because all the scores returned by this detector were similar, thus all the boxes have the same associated MF and increasing the cost c(0,1) (the cost of deciding not to blur a pixel while it has to be) does not gradually increase the number of blurred pixels.

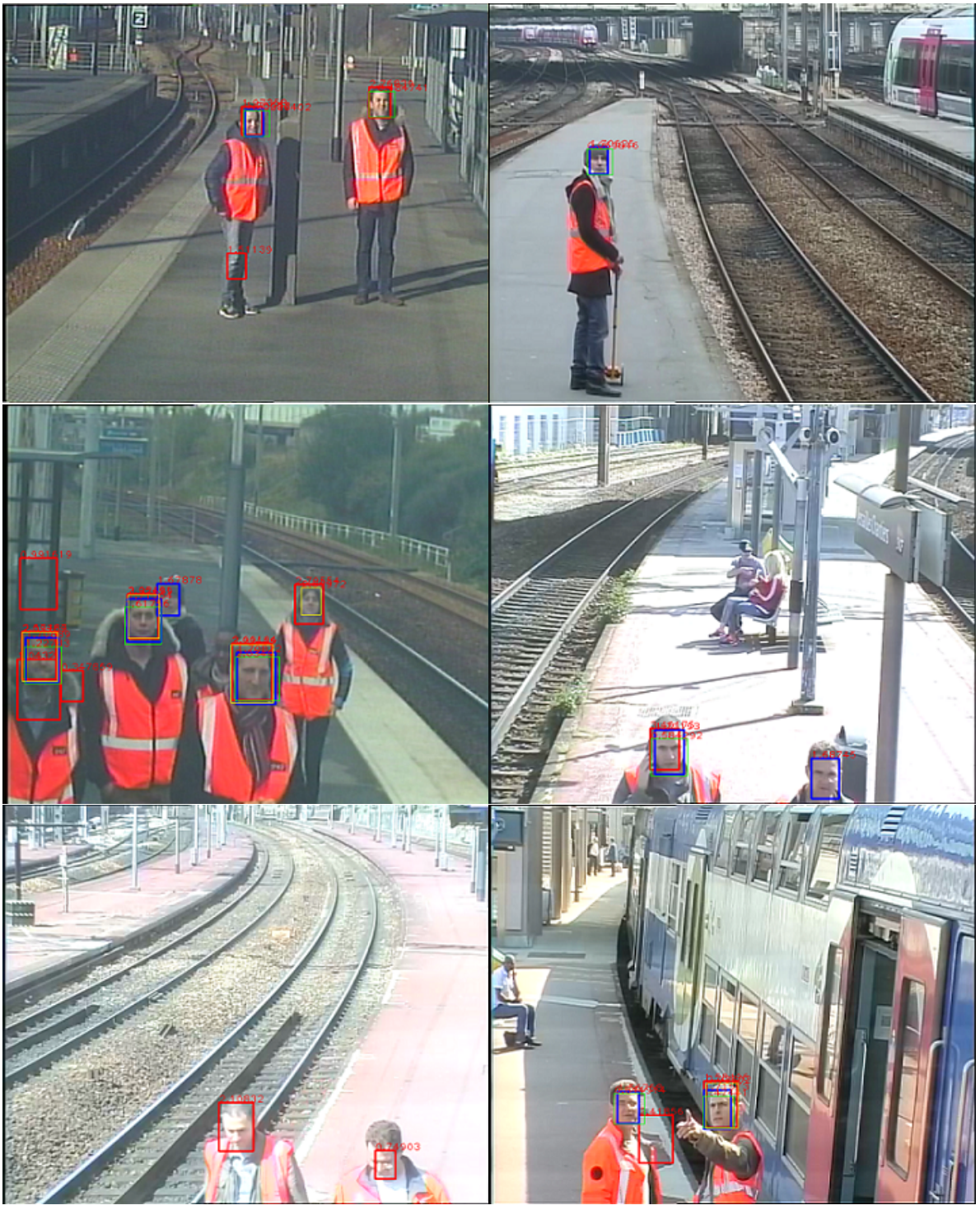


Figure 3.4 – Example of results returned by the four selected detectors. Haar+Adaboost detector is in red, LBP+Adaboost in yellow, HOG+SVM in green and DNN in blue.

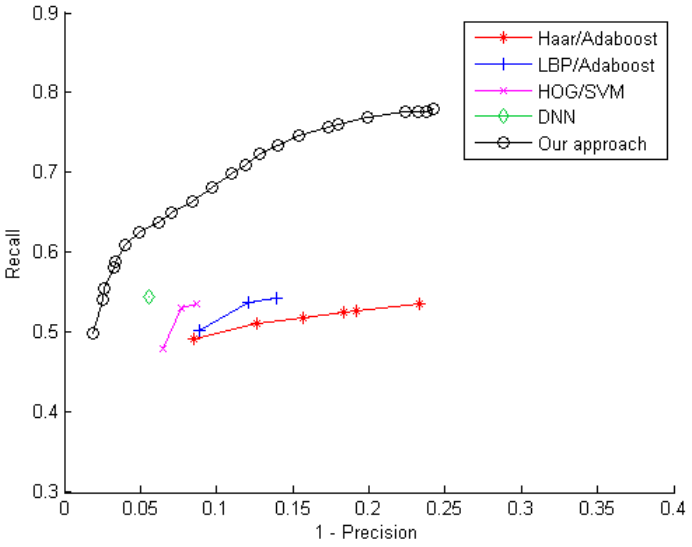


Figure 3.5 – Pixel-based approach vs detectors on FDDB.

Figure 3.6 shows the result for the same experiment but this time on the SNCF dataset. The conclusion is the same as the proposed approach has better performances than the detectors taken alone. Let us remark that their performances could be improved by training them with face and non-face images closer to those encountered in the SNCF dataset.

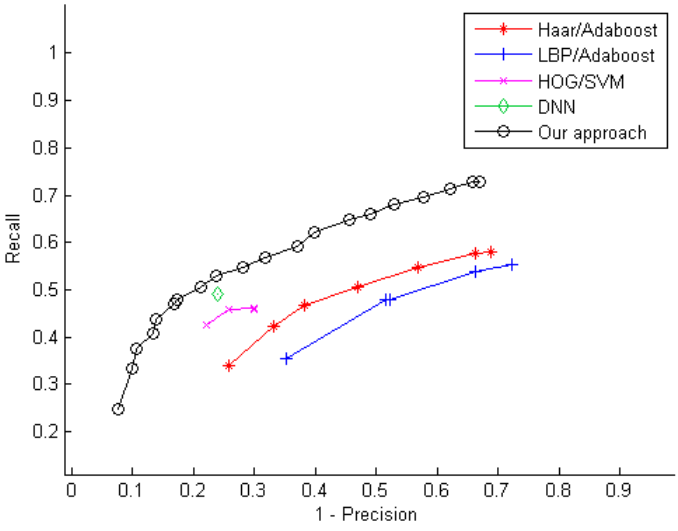


Figure 3.6 – Pixel-based approach vs detectors on SNCF dataset.

Comparison on the FDDB dataset between the box-based approach used with

different values of the overlap threshold λ and our approach is shown in Figure 3.7. As it can be noticed, for a same precision rate, the recall of our approach is always the highest. Figure 3.8 shows the results of this comparison on the SNCF dataset; the conclusions are the same.

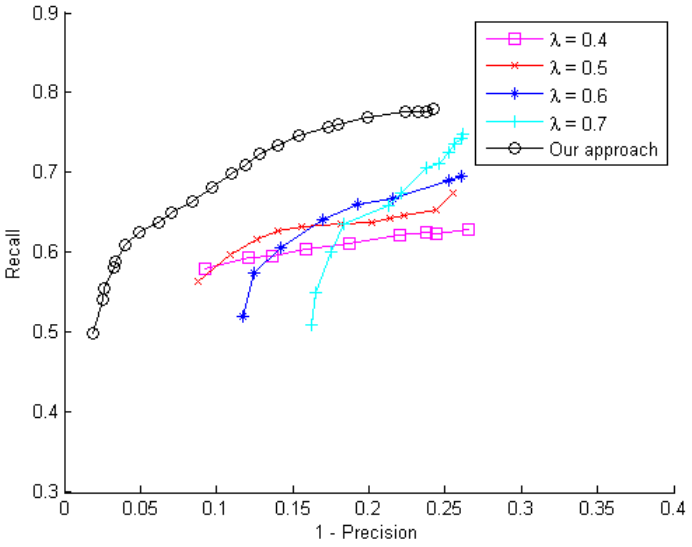


Figure 3.7 – Pixel-based approach vs box-based approach on FDDB.

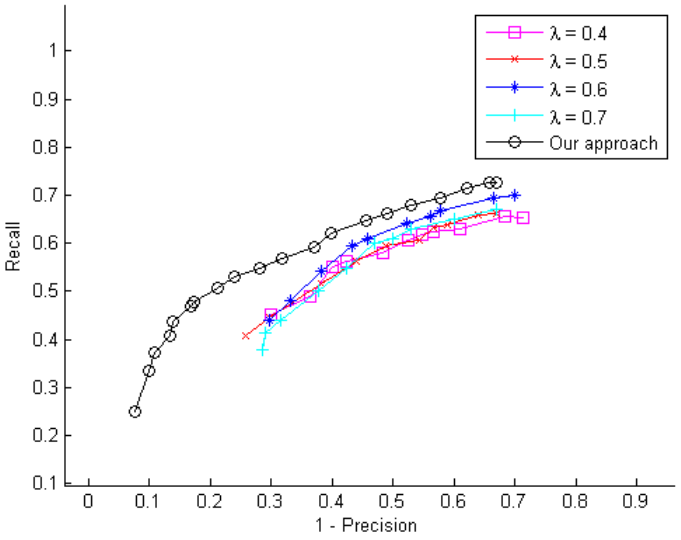


Figure 3.8 – Pixel-based approach vs box-based approach on SNCF dataset.

Let us note that reasoning at the pixel level rather than with boxes as in the box-based approach may involve a complexity issue. Indeed, as the fusion is performed on every pixel instead of on sets of boxes, the proposed approach has *a priori* a higher

complexity. For the pixel approach and for a given image, the number of operations is equal to $J \times a$, where J the number of fusion operations (which is equal to the number of used detectors) and a the number of pixels in the image. By contrast, in the box-based approach, the complexity is $O(b^2)$, with b the total number of boxes returned by J detectors. Indeed, at worst the clustering procedure is $O(b^2)$ [34] and this is the most costly step. Thus, at first glance, it seems that the complexity is much higher for the proposed approach as a is generally significantly higher than b^2. However, any two pixels $p_{x,y}$ and $p_{x',y'}$ that do not belong to any box of D_j have associated MFs with the same definitions, *i.e.*, we have $m_{*,j}^{\mathcal{P}_{x,y}}(A) = m_{*,j}^{\mathcal{P}_{x',y'}}(A)$, for all $A \subseteq \{0,1\}$. Thus, pixels that do not belong to any of the returned boxes by the detectors have the same resulting MF. This latter case happens often in practice, hence this allows us to have a common processing. For instance, in a set of 200 images of FDDB, with the four face detectors considered in our experiment, it corresponds on average at around 80% of the pixels of the image. In terms of time processing, an image takes on average around 120 milliseconds to process (including the time of detection of the four detectors) for the box-based approach and 150 milliseconds for the proposed system; we consider that it is a reasonable difference.

This section showed that given the same information,*i.e.*, detectors returning boxes, the proposed approach gives better results than the box-based approach. Our approach is a little more time-consuming but the difference is reasonable. The following section illustrates another advantage of our approach, which is its ability to integrate directly sources providing pixel-based information.

3.5.3 Addition of pixel-based information on disjoint approaches on FDDB and SNCF databases

Color information can be useful for the face blurring problem as the color of the faces, the skin tone, is very distinct from other colors. It is thus an interesting information that can be used to detect skin, and thus faces, in complex scene images. It is actually a widely studied subject and some surveys can be found in [15, 56, 106]. We used the same detector as in [101], where the RGB values are transformed to Normalized Color Coordinates (NCC), *i.e.*,

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}. \tag{3.9}$$

Pixels with chromaticity (r, g) are then classified as skin pixels or not using a threshold rule given in [101]. We used the same parameters given in [101] for FDDB experiments and recalculate some using a dataset of skin pixel values for SNCF experiment. It gives a detector that returns a binary image, which pixels are either classified as skin or non-skin.

In order to combine this color information with the other detectors, a mass function has to be associated to each pixel of the image. This information can be calibrated in a similar way as score absence is calibrated in Section 3.3.2. When a pixel $p_{x,y}$ is classified as skin by the skin detector, it is possible to obtain a MF, that we denote by $m_{skin}^{\mathcal{P}_{x,y}}$, representing the uncertainty with respect to the presence of a face on pixel $p_{x,y}$. The necessary training set is obtained using L images on which the skin detector is applied; the numbers of pixels which have been classified as skin can be obtained and as the positions of the faces on these L images are available, their true label Y_i is available. Thus, using this training set, the MF representing the uncertainty with respect to the presence of a face on pixel $p_{x,y}$ when this pixel is classified as skin can be calculated. Specifically, if we denote by TP (True Positive) the number of pixels classified as skin and belonging to a face on these images and FP (False Positive) the number of pixels classified as non-skin but actually belonging to a face, the MF can be defined by

$$m_{skin}^{\mathcal{P}_{x,y}}(\{0\}) = \frac{FP}{TP + FP + 1}, \quad m_{skin}^{\mathcal{P}_{x,y}}(\{1\}) = \frac{TP}{TP + FP + 1}. \tag{3.10}$$

In addition, given a pixel classified as non-skin, the whole process can be applied to define a MF representing the uncertainty with respect to the presence of a face on pixel $p_{x,y}$.

The same experiment as in Section 3.5.2 was performed, including the four face detectors, the two different datasets, and the decision strategy. The repartition of the images for the calibration training and the tests was also the same. The fifth source, *i.e.,* the skin detector which gives information on pixels, was simply added to the global system. Figure 3.9 compares the results of the pixel-based approach proposed in Section 3.5.2 and the new system now relying on a combination of the outputs of five detectors instead of four.

We may remark that the skin detection has a lower precision rate than the face detectors. It can partly be explained by the fact that all the other parts of the human body, such as hands or arms, may be correctly classified as skin but are counted as false positives as the ground truth is face positions. Furthermore, the color detector is only represented by one point in Figure 3.9 because all the pixels considered as skin have the same MF, likewise for the pixels indicating non skin. Thus, as for the deep neural network detector, increasing the cost $c(0, 1)$ does not gradually increase the number of blurred pixels. Actually, at some value of cost $c(0, 1)$, which is not represented in Figure 3.9, a second point for the color detector is obtained but it corresponds to a useless point where all the pixels are blurred by the color detector.

As it can be noticed in Figure 3.9, the addition of the skin color information improves the global combination although the performance of skin detection is not that good.

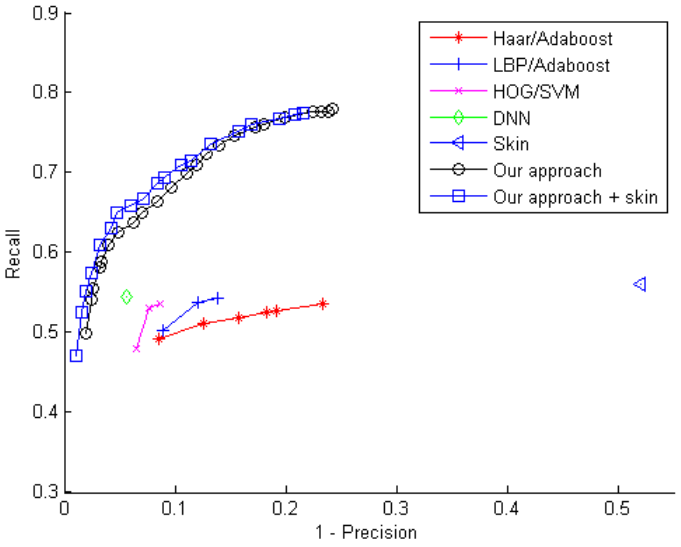


Figure 3.9 – Integration of skin color information to the proposed approach on FDDB.

Finally, we conducted the experiment on the SNCF dataset and the results are shown in Figure 3.10. The conclusion are the same as the integration of skin information also improves the overall performances.

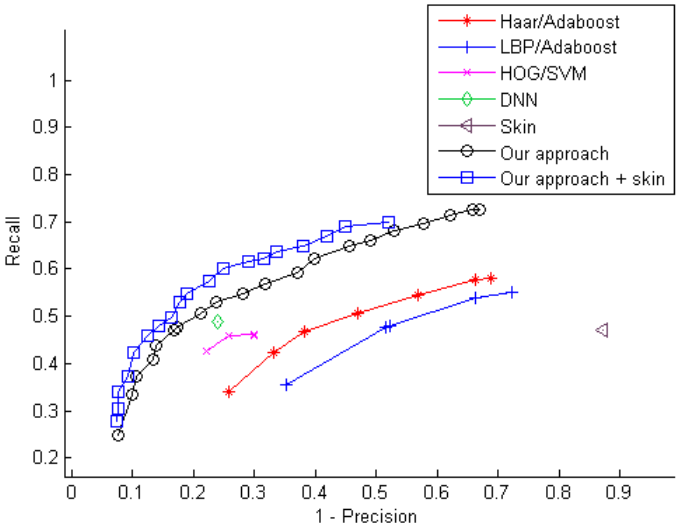


Figure 3.10 – Integration of skin color information to the proposed approach on SNCF dataset.

Figure 3.11 shows some examples of obtained results on SNCF dataset with-

Figure 3.11 – Comparison of the results obtained by our fusion approach without (left) and with (right) the integration of skin color detection.

out (left column) and with (right column) the integration of the skin color detection. The blurred pixels are in red for a better visibility. As it can be seen, adding the information of skin enables to "unblur" some incorrectly blurred pixels, but also to blur some pixels that should be blurred. Yet, it may also unblur some pixels that actually belong to a face.

These experiments concerned our disjoint pixel-based approach, which is an alternative to the box-based approach. A common point between these two approaches is that they calibrate independently each detector and rely on a predetermined rule of combination. We now propose to apply our evidential joint calibration for combining the scores. The results are presented in the next section.

3.5.4 Comparison between disjoint and joint approaches on FDDB and SNCF databases

The joint calibration approach was presented in the second chapter of this report and we have exposed how to apply it to the issue of face detection in Section 3.4 of this chapter. In this section, we present the obtained performance results and compare them to those of our proposed disjoint approach.

The performed experiment was exactly the same as in previous sections. Figure 3.12 shows the comparison between our disjoint approach and the joint one on the FDDB dataset.

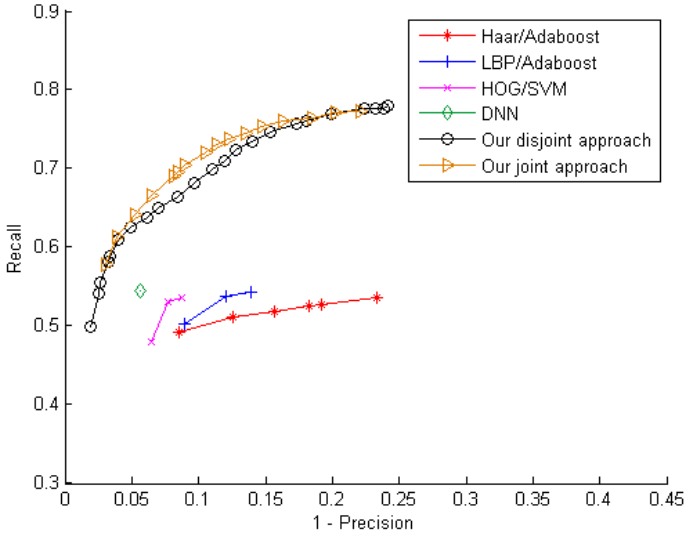


Figure 3.12 – Comparison between disjoint and joint calibration on FDDB.

As it can be noticed, the approach based on the joint calibration has better

global performance than the disjoint one. For instance, for a precision rate of 90%, the disjoint approach gives a recall rate of around 68% while it is equal to 72% for the joint approach. Finally, we conducted the same experiment on the SNCF dataset and the results are shown in Figure 3.13. The conclusion are the same as the joint calibration results are similar or better than the disjoint one.

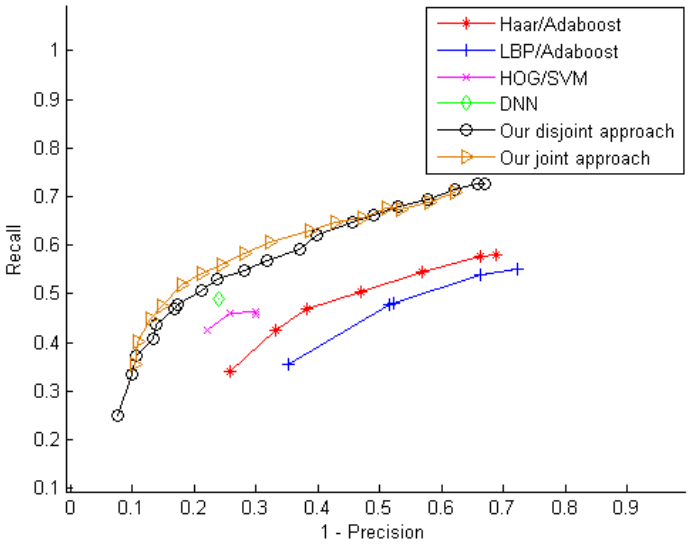


Figure 3.13 – Comparison between disjoint and joint calibration on SNCF dataset.

3.6 Conclusion

In this chapter, a pixel-based face blurring system relying on evidential calibration and fusion of several detector outputs was proposed. This pixel-based approach brings several advantages over a previous box-based proposal. First, an overlap threshold is no longer necessary, as well as a clustering step. Furthermore, it enables to integrate pixel-based or box-based information, and in the considered blurring problem, it allows us to model and to integrate to the fusion process the information of score absence for each detector, *i.e.*, a MF is defined for pixels which are not contained by any of the boxes returned by the detector. The proposed system also shown better performances than the box-based approach, either on a literature dataset or on a more challenging one. We also illustrated the ability of natively integrating a detector giving pixel-based outputs by adding a skin color detector to the global system; this latter addition further improved the overall performances. Furthermore, we applied the joint calibration approach described in Chapter 2 to the problem of face blurring and compared it to our proposed disjoint approach. The system based on joint calibration shows better performances than the proposed disjoint approach on both datasets. A perspective could be to add the skin detection in the joint approach.

All the experiments presented on this chapter concerned tests on still images, yet let us recall that the input of our system are videos. Thus, it seems interesting to exploit temporal information in order to improve the blurring performance. This is the topic of the next chapter.

Chapter 4

Face blurring on videos

Contents

4.1 Introduction

A video contains more information than still images that are not related to one another. In this case, it is interesting to take into account for each image the information contained by the previous frames, more specifically in our case the previous face positions. It can be performed using a tracking algorithm, that allows one to improve the positions of the presumed faces in the considered image based on the previous ones. The main difficulties to track moving objects include the occlusions, a cluttered background, the interaction between the objects, etc. The state-of-the-art in terms of object tracking is abundant [7, 14, 20, 119], and among all the existing approaches, the most known solutions are the Kalman filter [60, 61, 11] and the particle filter [3, 81, 35]. Both algorithms recursively predict an estimate of the state and

updates it given a sequence of observations (measurements), but Kalman filter is easier to apprehend and has much lower computational requirements than particle filters. Within this scope, we propose to integrate a Kalman filter, and more especially multiple Kalman filters, in our global system.

In this chapter, we first expose how a tracking algorithm can be integrated to our blurring system in Section 4.2. Then, in Section 4.3, the general principle of the well-known Kalman tracking algorithm is exposed. The results of the application of this algorithm coupled with our detection system are then compared to a simple detection system in Section 4.4, and the results are discussed.

4.2 Overview of the global system

The overview is first described for a given detector, then extended to the application of J detectors. A common detection-tracking system is composed of four main steps, which are the prediction, detection, association and correction steps, and that we explain below. Figure 4.1 illustrates the relation between these different steps for a system composed of one detector.

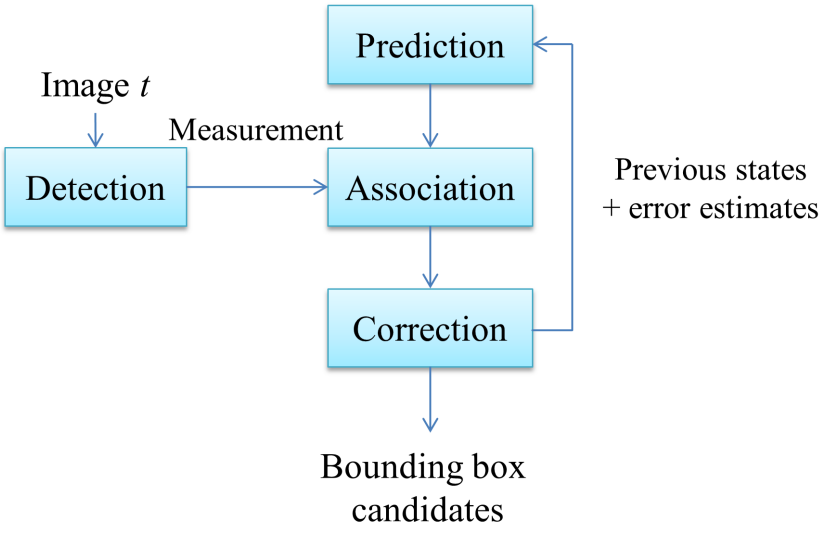


Figure 4.1 – Overview of the system for a given detector.

First, given an image at time t, the detection is performed, *i.e.*, the j^{th} detector is ran on this image and outputs a set of bounding boxes and associated scores. The tracking filter performs target tracking by predicting the future positions of the previous tracked boxes in the frames. For instance, it can be performed using the prediction equations given in Section 4.3.1 for the case of a Kalman filter.

Then, a step called association is necessary as multiple objects are considered in our application. Data association is a specific problem of tracking in a multi-object situation, which consists in finding the true position of the moving targets in presence of different valid candidates [8], *i.e.*, matching the new detected boxes with the tracked ones. The assignment problem is handled using the Hungarian algorithm, also called algorithm of Kuhn-Munkres [67]. The data association assigns one target provided by the detector to a track and manages the track creation, deletion and update. A track goes through these three steps:

- A track is created if there is a detected box which is not assigned to any of the currents tracks. The score associated to this detected box is assigned to this newly tracked box.

- A new detected box is assigned to a track, in which case the detection is used as a measurement for the correction step of the corresponding track. It corresponds to the correction equations given in Section 4.3.2 for the particular case of Kalman filter. Furthermore, the lifetime of the corresponding track is reset, and the score assigned to the tracked box is replaced by the score associated to the new detected box.

- The deletion of a track happens when its lifetime is over a threshold, *i.e.*, when a tracked box has not been detected for p frames, this box is no longer considered. It solves the problem of tracks that have lost their target, for instance if the person has left the filmed scene.

Let us now consider our global system, which is composed of J detectors. Figure 4.2 illustrates the different steps of this global detection-tracking system composed of J detectors. Specifically, it corresponds to J systems of Figure 4.1 and the fusion step.

A tracking filter is defined for each detector, *i.e.*, the boxes returned by a given detector have their own steps of prediction, association and correction. Each tracking filter returned a set of bounding boxes with corrected positions. They are called "candidates" as the fusion step may decide that it does not correspond to a face and thus does not blur the corresponding area. This fusion step corresponds to our joint calibration approach defined in Chapter 2 and applied to the face blurring issue in Chapter 3. Finally, the output of this global system is the input image with the blurred faces.

More specifically, the steps of prediction and correction are performed using a standard Kalman Filter, which general principle is described in next section.

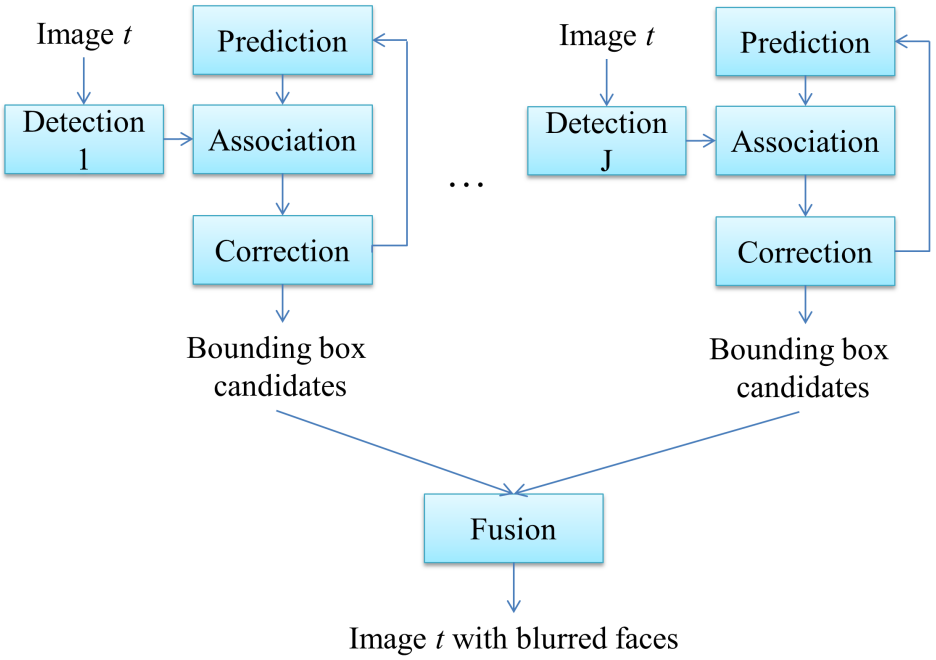


Figure 4.2 – Overview of the global system.

4.3 Kalman filter-based tracking

The Kalman filter is named after Rudolf Emil Kalman, who designed this algorithm more than fifty years ago [60]. Yet, it is still one of the most important and common used tracking algorithm up to this day. It has been successfully used in different prediction applications, such as in satellite navigation device, smoothing the output from laptop trackpads, tracking multiple objects, etc. For instance, the most famous use of the Kalman filter was in the Apollo navigation computer that took N. Armstrong to the moon.

This filter is a recursive estimator as only the estimated state from the previous time step and the current measurement are needed to compute the estimate for the current state. The state of the filter is represented by two variables:

- $\hat{x}_{i|j}$ represents the estimate of x at time i given observations up to and including time $j \leq i$, where x contains the terms of interest for the system, *i.e.*, in our case the position, size and velocity of a face.
- $P_{i|j}$ represents the error covariance matrix at time i (a measure of the estimated accuracy of the state estimate).

The Kalman filter algorithm works in a two-step process: prediction and cor-

rection (also called the update step). The first step uses previous states to predict the current state of the considered objects. The second step uses the current measurement, in our case the outputs of the detection step (position and size), to correct the state.

4.3.1 Prediction step

In this step, the state of the system and its error covariance are transitioned using a defined transition matrix F. The standard Kalman filter equations for the prediction stage are defined by

$$\hat{x}_{t+1|t} = F_t \hat{x}_{t|t} + Bu_t, \tag{4.1}$$

$$P_{t+1|t} = F_t P_{t|t} F_t^T + Q_t, \tag{4.2}$$

where F_t corresponds to the state-transition matrix, which applies the effect of each system state parameter at time $t-1$ on the system state at time t. The matrix B_t is a control-input matrix for each time-step t, which is applied to the control vector u_t (not used in our system). Q_t corresponds to the covariance matrix of a process noise.

4.3.2 Correction step

This second step is also called innovation step or update step. It consists in correcting the states using the measurement and the predictions. The standard Kalman filter equations for the update stage are defined by

$$\hat{x}_{t+1|t+1} = \hat{x}_{t+1|t} + K_{t+1}(y_{t+1} - H_{t+1}\hat{x}_{t+1|t}), \tag{4.3}$$

$$P_{t+1|t+1} = P_{t+1|t} - K_{t+1} H_{t+1} P_{t+1|t}, \tag{4.4}$$

with

$$K_{t+1} = P_{t+1|t} H_{t+1}^T (H_{t+1} P_{t+1|t} H_{t+1}^T + R_{t+1})^{-1}, \tag{4.5}$$

and y_t the measurement vector, H_t the measurement matrix that defines the mapping from the state vector to the measurement vector and R_t the covariance matrix of the measurement noise. The variable K_t is called the Kalman gain.

In order to use the Kalman filter, the matrices F_t, H_t, Q_t and R_t need to be specified according to a motion model. To do that, we used the model of a constant velocity.

4.4 Experimental results

In this section, the results of a frame-by-frame system are compared to the system that integrates a tracking algorithm. These experiments are performed on different annotated SNCF videos, that contain various situations and environment. First, Section 4.4.1 describes the experiment in further details. Then, the obtained results are presented and discussed in Section 4.4.2.

4.4.1 Description

The inputs of the global system are SNCF videos, extracted from EAS system as explained in the Introduction. They have been manually annotated and present various situations: there are more or less persons, different image qualities, walking or running persons, etc. They are between around 45-second long and 2 minutes 30-second long. Their characteristics are given in Table 4.1.

	Number of images	Number of faces
Video 1	2002	386
Video 2	1345	635
Video 3	3631	1526
Video 4	1142	775

Table 4.1 – Particularities of the tested videos.

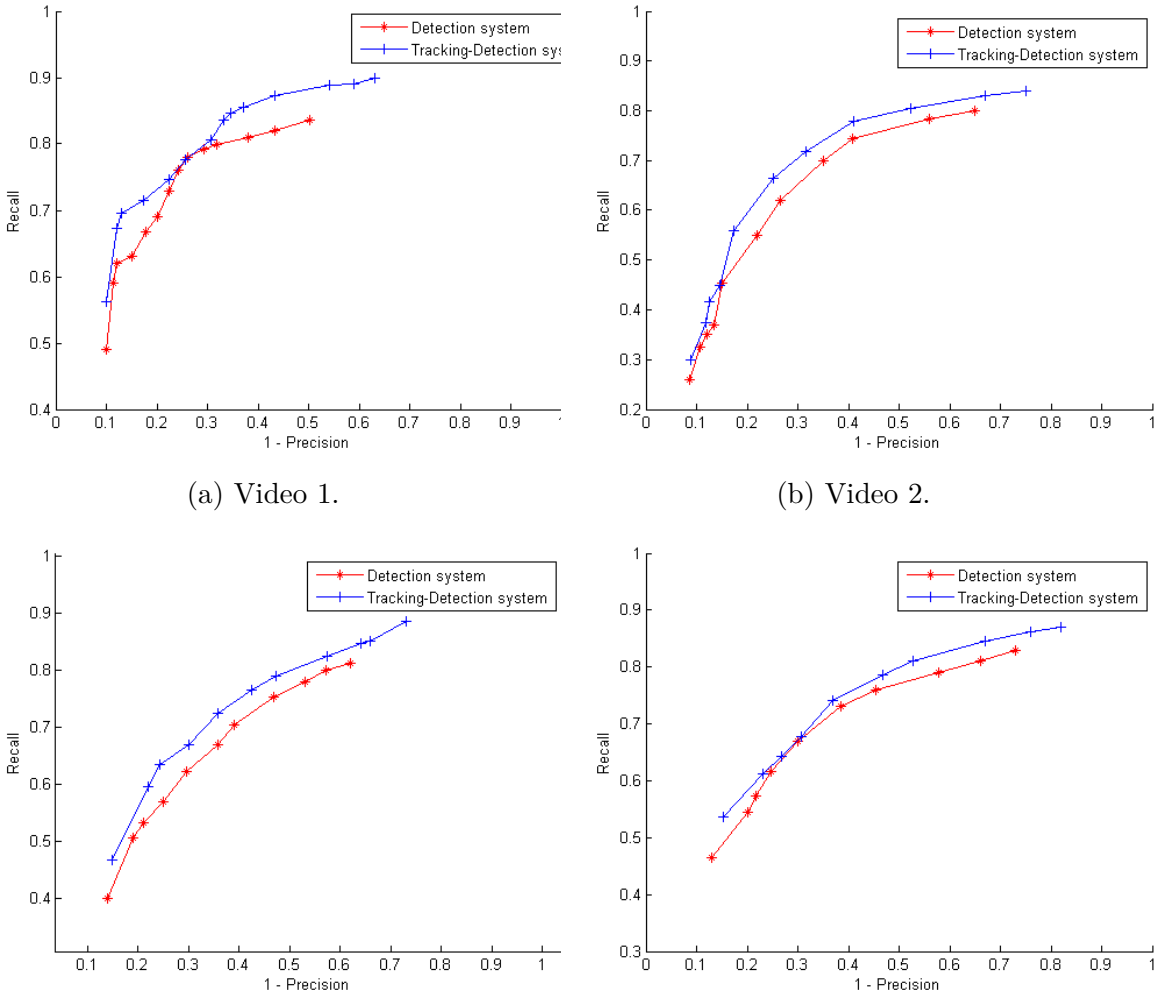
The four selected detectors returning bounding boxes and associated scores are applied in each frame of the video. The first considered system consists simply in applying the joint calibration approach, whose results on still images have been presented in Section 3.5.4 of Chapter 3. The joint calibration approach directly uses the outputs of the detectors. We call this system the *detection* system.

The second system integrates the Kalman-based tracking algorithm as explained in Section 4.2, *i.e.*, for a given image the obtained positions of the boxes are updated using the association and tracking algorithms, then the joint calibration approach is applied using these corrected positions as inputs. We refer to this system as the *detection-tracking* system.

Concerning the performance measure, we use the decision procedure relying on upper expected costs, as in the previous chapter. By gradually increasing $c(0, 1)$, we obtain different points for the recall and precision rate. The obtained results are presented for both systems in the next section.

4.4.2 Comparison of results between the detection and detection-tracking systems

Figure 4.3 shows the results obtained for the two different systems on four different videos, and Figure 4.4 shows example of image extracted from these videos. Recognizable persons that are not SNCF employees are made anonymous for legal reasons.



(a) Video 1.

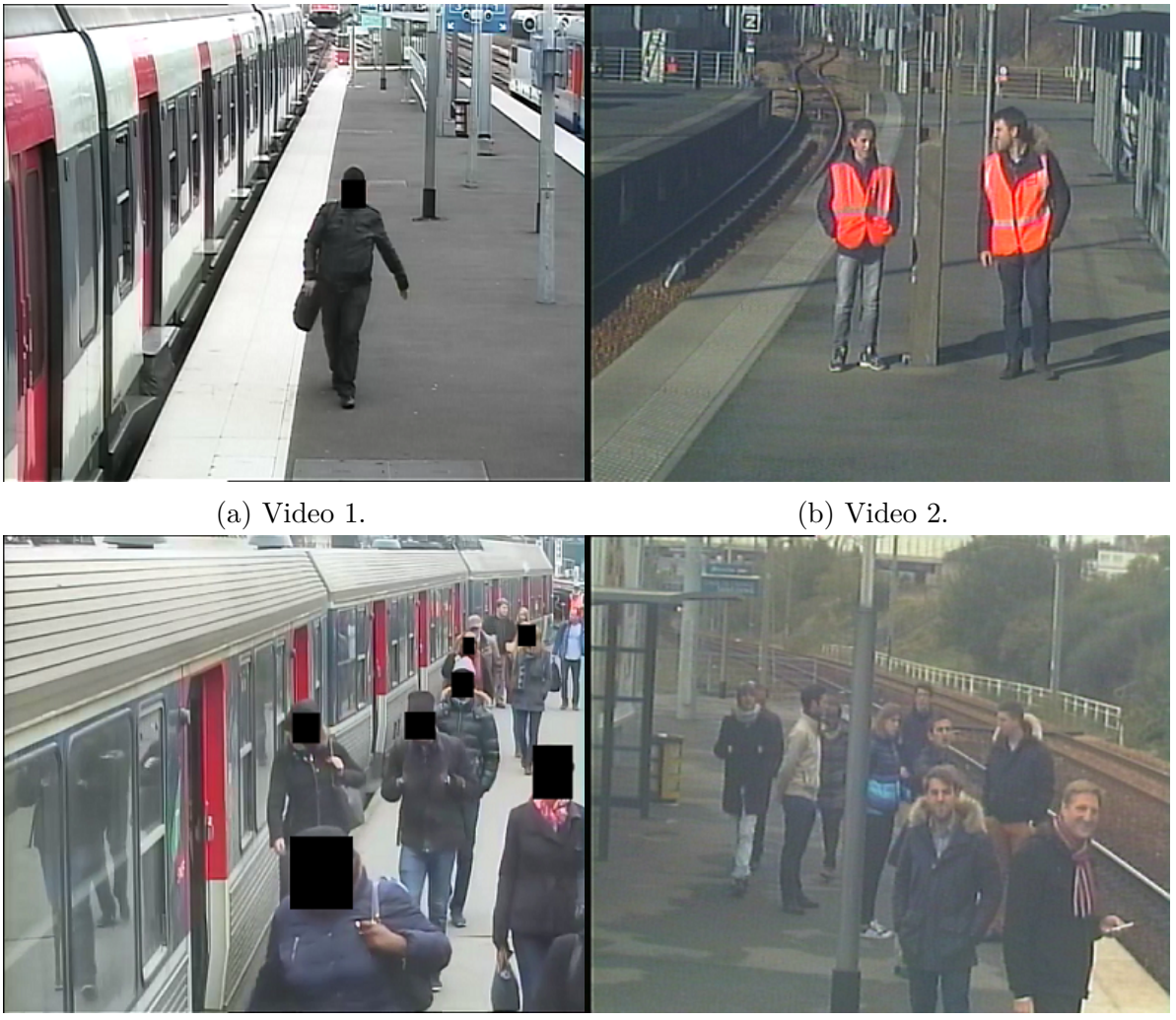
(b) Video 2.

(c) Video 3.

(d) Video 4.

Figure 4.3 – Comparison of performance between the detection and tracking-detection systems on four different videos.

As it can be noticed in Figure 4.3, adding a tracking algorithm enables to improve the overall performance. Indeed, for a given precision rate, the obtained recall rate is always higher for the tracking-detection system than for the detection system.



(a) Video 1. (b) Video 2.

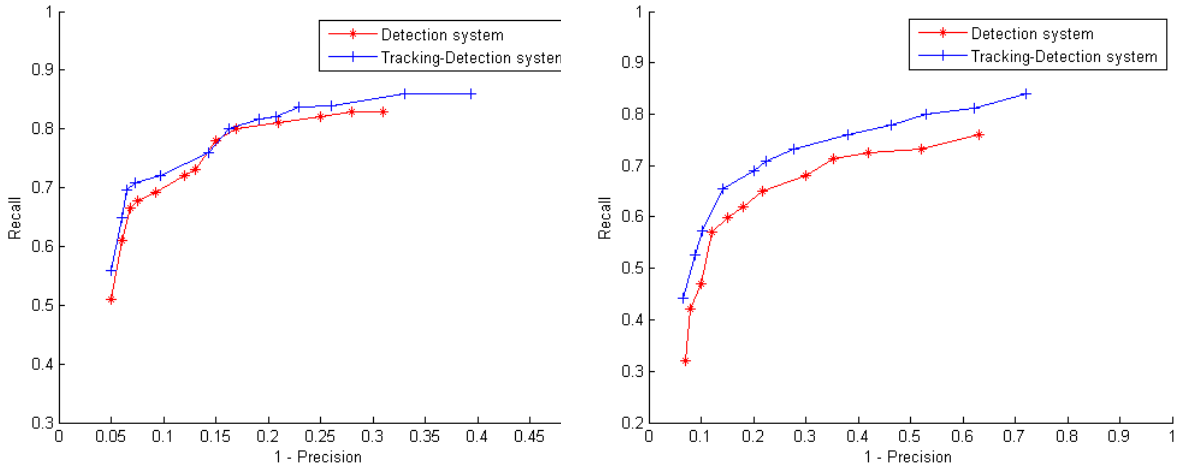
(c) Video 3. (d) Video 4.

Figure 4.4 – Example of image extracted from the four different videos.

We may also remark that the interpretation of the results depends on the value of the cost $c(0,1)$. Let us take the example of Figure 4.3a; when the cost $c(0,1)$ is equal to 1, it corresponds to a precision rate of around 90% for both systems while a recall rate of around 49% for the detection system and 56% for the tracking-detection system. Thus, the tracking algorithm enables to improve the recall rate while keeping the same precision rate. Yet, when the cost $c(0,1)$ is higher, *i.e*, when more candidates pixels are blurred, the recall rate is still improved but the precision rate decreases. For instance, if we consider the last point for both systems in Figure 4.3a, we obtain respectively 50% and 83% of recall and precision rates for the detection system and 38% and 90% for the tracking-detection system. Thus, in that case, the tracking enables to reach a better recall rate but also decreases the precision. The difference between the two considered cases may be explained by the fact that when $c(0,1)$ is low, only the pixels with a high MF are blurred, and they are more likely to belong to a face.

Thus, the tracking enables to blur some pixels that belong to faces that may be not detected in every frame in the detection system. Yet, the higher the cost, the more pixels are blurred, and those pixels may actually not belong to a face. The tracking will thus continue to blur a false positive, and it leads finally to a lower precision rate than in the detection system.

Furthermore, we tested the two systems on two small video extracts corresponding to particular situations of 5 seconds each, *i.e.*, 125 images each. In the first one, which results are given in Figure 4.5a, a person is walking along the railway looking in front of him, *i.e.*, in the direction of the camera. In that case, the detection process works well. The second case corresponds to three persons who look in the direction of the camera, then turn a little, then look again at the camera, etc. In that case, the detection misses some faces. The results of this second case are given in Figure 4.5b.



(a) Case 1. (b) Case 2.

Figure 4.5 – Comparison of performance between the detection and tracking-detection systems for two different cases.

As it can be noticed, in both extracts the detection-tracking system gives better performance than the detection-system, but the difference of results is more striking in the second case (Figure 4.5b) than in the first case (Figure 4.5a).

4.5 Conclusion

In this chapter, we have introduced a tracking algorithm in order to take advantage of the previous information, as the inputs of the considered system are videos. It allows to obtain better estimated positions of the bounding boxes than

a system only based on detection, and we showed that it enables to improve the performance, especially in certain particular situations.

Let us note that we chose a classical algorithm of the literature that may be replaced afterwards by a more efficient one; for instance, a tracking algorithm which is able to work with non-linear systems, such that for instance a particle filter or the extensions of the general Kalman filter that have been developed in that sense, such that the Extended Kalman filter or the Unscented Kalman filter [57]. Furthermore, a smoothing algorithm could be considered in order to obtain better position estimations, as a smoothing algorithm also takes into account the next frames of the videos in addition to the previous frames. The association step could also be improved with a better approach [33].

Conclusion

Summary of contributions

This thesis aimed to study and develop an efficient fusion system composed of different classifiers in order to blur people faces on SNCF videos. Two main contributions were exposed in this thesis.

The first one is the evidential joint calibration approach proposed in order to handle the scores returned by multiple classifiers. This approach belongs to the category of trainable combiners as it takes a score vector as input and does not need a predetermined rule of combination. Of particular interest is that we used evidence theory to handle better the uncertainties associated with calibration techniques. Our approach was compared to Xu *et al.*'s disjoint approach, which independently calibrates the scores of classifiers using the evidence theory and combines the obtained mass functions using Dempster's rule of combination. We compared also our proposed method to an approach belonging to the trainable combiner category and based on an evidential classifier. In both cases, the obtained results for our evidential joint calibration based on logistic regression either are better or are comparable to that of the other approaches. Furthermore, by introducing the possibility to reject a test sample, we showed the advantages of the evidential multivariable logistic-based calibration over the probabilistic version: it models more precisely the uncertainties and it exhibits better performances. Yet, we may notice that it is more convenient to have a disjoint approach if the considered application aims to add more and more classifiers, as the joint calibration needs to entirely rebuilt in this case. Furthermore, the complexity is higher in the joint case than in the disjoint one, but using existing techniques that aim to decrease the computation time of the gradient ascent enables to obtain a reasonable time of processing even in the joint case.

Our second main contribution consisted in studying the issue of face blurring on an image. A well-founded box-based approach of detection was applied at a pixel-level and it has been shown that this approach brings several advantages over the box-based proposal as well as better performances. In the considered blurring problem, it allows us to model and to integrate to the fusion process the information of score ab-

sence for each detector. We also illustrated the ability of integrating a detector giving pixel-based outputs by adding a skin color detector to the global system; this latter addition further improved the overall performances. The experiments were performed on a literature dataset as well as on a more challenging one. This latter dataset corresponds to SNCF images, in which the final goal is to blur the faces. We also exposed how to apply the joint calibration approach to this face blurring issue and compared it with the disjoint proposed approach; it showed similar or better performance.

To sum up, following the works of Xu *et al.* on evidential calibration, we proposed and studied approaches based on evidential joint calibrations that enable to obtain a belief function for a given object without the need of a rule of combination. We published this work in [77]. Furthermore, we developed an approach for face blurring based on evidential calibration and applied at a pixel-level; it leads to several advantages compared to a state-of-the-art box-based approach. This work has been published in [76, 78]. The first contribution was also tested on that application.

Finally, as the input of the application are videos, a tracking algorithm has been integrated to the system in order to account for the temporal information. Indeed, one could use the fact that a pixel is more likely to be blurred if it has been blurred on a previous image. We used a general basic algorithm as a first approach, and we showed that it enables to improve overall performance, and in particular to increase the recall rate, *i.e.*, to miss less faces that have to be blurred.

In parallel of this work, we developed a human-machine interface that allows the user to manually interact and that integrates different functionalities in order to have the most efficient tool to blur faces on videos. For instance, the possibility of going back on the videos has been made possible so that if the system missed a face, the user can return to the previous frames and manually blur it. Furthermore, a permanent blurring may be settled during all the video. It is useful if there is an area of the video that we constantly want to be blurred. On the contrary, the permanent un-blurred may be performed, *i.e.*, an area of the image that will never be blurred.

Future works

The work presented in this thesis may be continued in many directions, and may concern the application, *i.e.*, the face blurring issue, but also the proposed joint calibration approach. In the following paragraphs, we sketch a few of them. More specifically, some prospects concerning the joint calibration are first exposed followed by the ones concerning the application.

First, our evidential joint calibration may be applied with different inputs; indeed, we tested it with scores returned by classifiers, but it could be applied directly on the features of the examples given by UCI datasets, that is it may be used as an

evidential classifier.

Furthermore, we only applied our approach to some binary classification problems. The extension of the proposed evidential joint calibration to the multi-class problem may be tackled in future works, following the same reasoning of [115], which addressed this extension in the single classifier case.

Finally, it seems interesting to develop an evidential joint calibration approach relying on a generalization of logistic regression, known as choquistic regression [105], in order to have a more flexible modeling of the interactions between the classifiers.

Concerning the application, the proposed pixel-based approach can be applied with other detectors, which can be based on boxes or pixels. One perspective consists in replacing one of the face detectors by a more efficient one or to add one to the global system. Indeed, new face detectors are proposed every year, especially since the breakthrough triggered by the deep neural networks, and the purpose of this work was not to develop a new face detector but to investigate on how combining the outputs of several classical detectors. Furthermore, the same remark could be made concerning the tracking algorithm, which could be replaced for instance by a method accounting also the next frames.

Another perspective is to make use of the spatio-temporal context of a given pixel directly in the joint calibration. Indeed, as recalled, we used as input to our joint calibration approach the scores provided by some binary classifiers and it gives us a face blurring system in still images. Then, we added a tracking algorithm in order to account for temporal information. A perspective could thus be to directly integrate all the available information in a global joint calibration, *i.e.*, the scores given by the different classifiers but also the temporal information. Similarly, it is reasonable to consider that a pixel is more likely to be blurred if its neighbours have been blurred, and this information could also be taken into account, perhaps in the joint calibration. It could be an inspiration to extend our approach and make it more general. Yet, it would increase the number of parameters in the calibration and thus a complexity issue may appear.

Publications

International journals

- P. Minary, F. Pichon, D. Mercier, E. Lefèvre, B. Droit. A pixel-based face blurring approach using evidential calibration and fusion. *International Journal of Approximate Reasoning*, pp 202-215, Vol. 91, December 2017.

International conferences

- P. Minary, F. Pichon, D. Mercier, E. Lefèvre, B. Droit. Evidential joint calibration of binary SVM classifiers using logistic regression. In *Proceedings of the 11th International Conference on Scalable Uncertainty Management, Granada, Spain, October 4-6,* pages 405-411, Springer, 2017.

- P. Minary, F. Pichon, D. Mercier, E. Lefèvre, B. Droit. An Evidential Pixel-Based Face Blurring Approach. In *J. Vejnarová and V. Kratochvil, editors, Belief Functions: Theory and Applications, Proceedings of the Fourth International Conference on Belief Functions, Prague, Czech Republic, September 21-23, 2016,* volume 9861 of Lecture Notes in Computer Science, pages 222-230, Springer, 2016. (Best student paper award).

- P. Minary, B. Droit, F. Pichon, D. Mercier, E. Lefevre. A fusion method for blurring faces on platforms using belief functions. In *11th World Congress on Railway Research, Milan, Italy, May 29-June 2*, 2016.

National conferences

- P. Minary, F. Pichon, D. Mercier, E. Lefèvre, B. Droit. Calibration évidentielle conjointe de classifieurs SVM binaires fondée sur la régression logistique. In *Les Rencontres Francophones sur la Logique Floue et ses Applications (LFA), Amiens, France, 19-20 Octobre 2017.* (Best student paper award).

Appendix A

Main approaches for face detection

Contents

The field of computer vision and image processing has been subject to numerous researches in recent years. Regarding the object detection category, and more specifically face detection, the applications are multiple and can lead to systems with countless opportunities, such as face recognition [124], facial expression recognition [84], or face tracking [41, 64]. For instance, the famous social network Facebook uses face detection and recognition algorithms for the purpose of automatic tagging people on images. As detection is the cornerstone of these systems, it is thus essential to have an efficient algorithm, presenting the best possible performance. Given an image, the aim of face detection is to determine whether it contains any faces and, if present, return the position(s) and size(s) of the detected face(s) in the image. This output is often represented by a rectangular box or an ellipse bounding each face. This detection problem is considered as a complex issue because faces are deformable objects: appearance (shape, color) and orientation are some examples of variability. Furthermore, they are directly affected by the sensor quality (camera), the environment (uncontrolled light source), or by partial occlusions (objects hiding part of the face).

Before the 2000's, the main algorithms aiming to detect a face were based on features such as geometrical, color-based or texture-based methods. They were quite simple but did not work well in case of complicated environment. In fact, they mainly concerned face localization problem, which aim to determine the position of a single face in an image; this is a simplified detection problem with the assumption that an input image contains only one face. Face detection has made significant progress since, especially as the significant breakthrough achieved with the system based on boosting proposed by Viola and Jones [109, 110]. Furthermore, the development of machine learning techniques, helped by the rapid growth in processing power and storage capacity of the modern computers, significantly improved the detection rate. Also, these techniques are based on image dataset as it will be explained in Section A.1 and the community created more and more growing databases. For instance, recently a new database called WIDER FACE and composed of more than 30 000 images with almost 400 000 face annotations was made available [74].

The state-of-the-art regarding the methods aiming to detect a face is abundant, especially since the past two decades. Over the years different categorizations of the approaches of face detection have been proposed in the literature, evolving over time. In the early 2000's, these techniques were usually divided into four categories [118]:

- Knowledge-based methods, which use pre-defined geometric rules based on human face knowledge.
- Feature invariant methods, which find structural features such as colour, contours or texture.
- Template matching methods, which use pre-stored face templates.
- Appearance-based methods, which learn the appearance of the face using a dataset.

Another type of categorization was also commonly used to group the different methods of face detection [72, 51]; it distinguished methods based on the local features of the face and the so-called holistic methods. The first one regroups the geometrical methods, based on face geometrical configuration, the texture-based methods and most importantly the color-based methods. Skin detection is a very important element and has been the subject of numerous researches [56, 106, 101, 58, 108]. The holistic approaches take the face as a whole and are based on learning techniques (this latter category refers thus to appearance-based methods of the first categorization).

Yet, in the light of recent technical and scientific developments, a third categorization has appeared. Indeed, the modern face detectors are mostly appearance-based methods, *i.e.*, they need a dataset composed of face images to be trained. Thus, the

above categorizations hardly applies on recent approaches. A recent survey on face detectors can be found in [122], where the authors organized the algorithms in the following two major categories:

- Methods based on rigid-templates, such as boosting and neural networks.
- Methods that learn and apply a Deformable Parts-based Model (DPM) [42] to model a potential deformation between facial parts.

Furthermore, in [38] the authors regroup the approaches in three categories: cascade-based, DPM-based and neural network-based.

Following these recent categorizations, and as DPM approach uses the classical HOG+SVM object detection algorithm [82], we propose to expose three classical approaches to detect a face that are the Viola & Jones approach (based on boosting and cascade), HOG+SVM, and neural network.

A.1 Overall description of modern face detectors

The modern face detectors, *i.e.*, detectors developed in the past two decades, mostly follow a similar process: a training step, a technique of sliding window and a grouping of redundant detections.

First, a training step is performed aiming to teach the classifier to recognize a face, using a dataset of images. This training is said supervised when each object of the training set is labeled, *i.e.*, the class of each object is known. More formally, for face detection, the training set is defined by $\mathcal{S} = \{(x_1, y_1), ...(x_p, y_p)\}$ where x_i corresponds to the i^{th} image of the dataset and y_i its associated label, *i.e.*, $y_i = 1$ for a face image and $y_i = -1$ for a non-face image. All the images must have the size corresponding to the size of the sliding window. The examples contained in the training set must be as representative as possible of the considered application and in significant quantity. Indeed, this principle is, in a certain sense, similar to the functioning of the human brain: we end by recognizing particular objects thanks to the characteristic elements that make up these, but not only; it is also essential to have seen several times during his life. For each example of $\mathcal{S}$, a vector of features is extracted and enables the classifier to discriminate a face of a non-face. The goal of training is to find a function able to make a decision about the class membership of the input examples based on the feature vector, while having the lowest possible classification error. We give some example of training algorithm model (AdaBoost, SVM) and features (Haar, LBP, HOG) in the following sections. This training step is performed upstream of the test process as it can take time, depending on the type of algorithm and the size of the training set $\mathcal{S}$.

Then, for a given image test, a sliding window technique is used to scan each area of the image. At each position of this window, a feature vector similar to those

extracted during training is computed. By comparing it to what it has learned during training, the classifier must make a decision and decide whether or not that portion of the image represents a face. Furthermore, it often provides as well a confidence score associated to this portion, which gives an idea on how much the classifier is confident that the considered portion is a face.

Finally, a step is necessary to group multiple detections, *i.e.*, detections that belong to the same face. Indeed, firstly because the window moves a few pixels on the image, the faces can be detected several times. In addition, this sliding window process is applied to different scales of the image, and thus it may also cause multiple detections. It is therefore essential to carry out a grouping of these multiple detections. The technique generally used is to set a minimum number of neighbours for detection to be maintained; the detections are considered neighbours if their overlap exceeds a certain threshold, generally fixed at 50%. All the rectangles in a group are then replaced by the average rectangle of that group. The larger the parameter, the lower the number of false detections but the more faces are missed, and vice versa.

This general architecture, represented in Figure A.1, applies to most of modern approaches. In particular, it is the case for Viola & Jones and SVM approaches, that we respectively describe in Section A.2 and A.3. Yet, as it will be explained in Section A.4, neural networks are not based on specific features but learn the features by themselves.

A.2 Viola & Jones

The first boosting procedure was initially proposed by Schapire [93]. Years later, Freund [43] proposed a new boosting algorithm based on the ideas presented in [93]. After these initial separate works on boosting algorithms, Freund and Schapire proposed the adaptive boosting (AdaBoost) algorithm [45, 44]. This Adaboost algorithm was in particular successfully applied to the issue of face detection by Paul Viola & Michael Jones in the early 2000's [109]. Indeed, their proposed algorithm triggered a revolution in the field of face detection and became one of the most famous face detector for many years and still remains widely used in the community. This work contains three main ideas:

- The use of a so-called integral image to compute faster the features;
- The training of the classifier with a boosting-based approach;
- The cascade architecture to reduce computational time.

Section A.2.1 defines common features used with the Viola and Jones algorithm, which are the Haar features and Local Binary Patterns (LBP). The integral

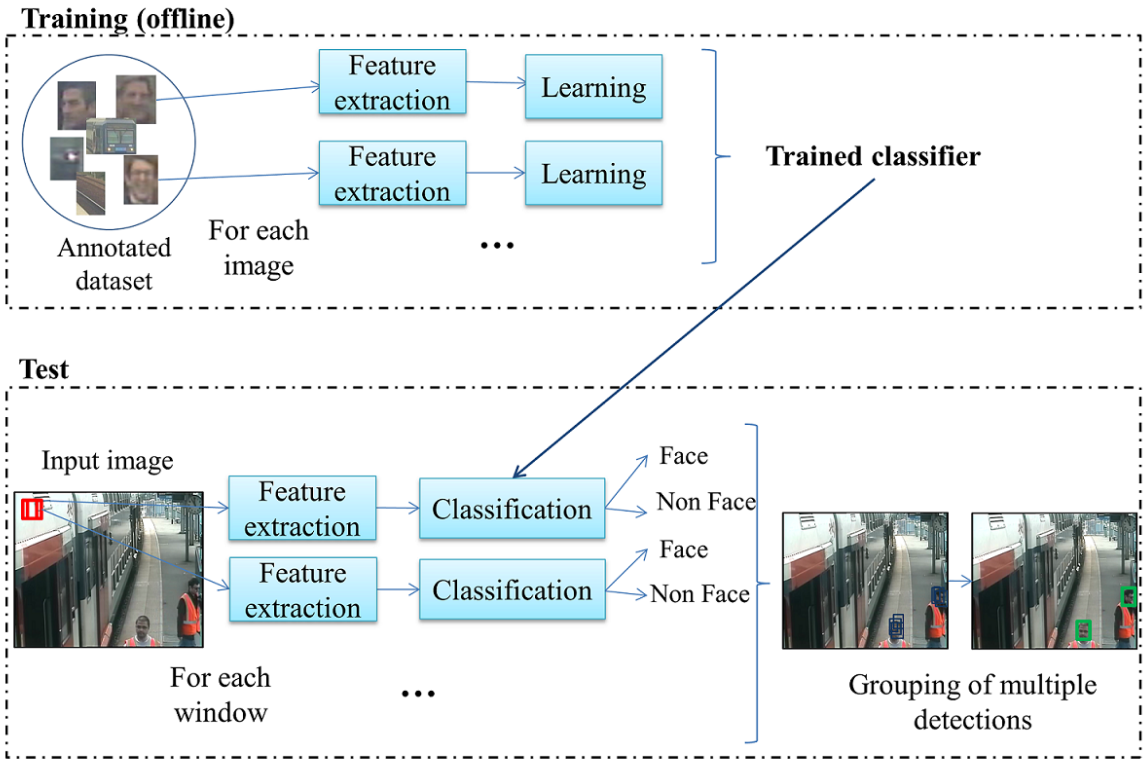


Figure A.1 – Illustration of the general principle of modern face detectors.

image principle is also described in this section. The training model AdaBoost applied for face detection is described in Section A.2.2, and the cascade architecture is given in Section A.2.3.

A.2.1 Haar and Local Binary Pattern features

The Haar features are obtained by calculating the difference between the sum of the pixels of adjacent rectangles. Examples of these adjacent regions are represented in Figure A.2; the Haar features compute the intensity difference between the black rectangular regions and the whites one. These area can be defined with various sizes and orientations. Two examples are given in Figure A.2.

In order to compute faster these features, Viola and Jones proposed to use integral image. The pixel at position (x, y) of the integral image is defined as the sum of all the pixels at the top and left of this pixel, which means that the integral image II of image I is constructed as follows:

$$II(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y'), \tag{A.1}$$

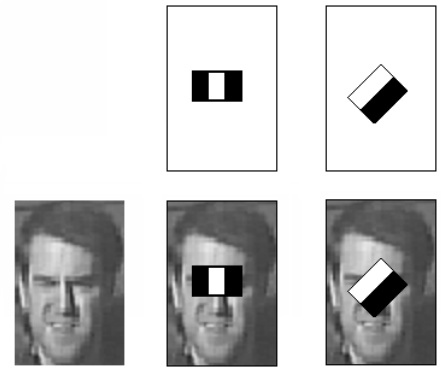


Figure A.2 – Haar-like features.

where $II(x, y)$ is the pixel value of the integral image at pixel location (x, y) and $I(x', y')$ the pixel value of the original image at pixel location (x', y'). Thus, the sum of pixels inside the rectangle region ABCD of Figure A.3 can be computed using $II(D) + II(A) - II(B) - II(C)$.

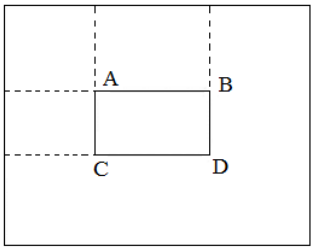


Figure A.3 – Illustration of an integral image.

Thus, to compute a feature composed of two (resp. three) rectangles for instance, using the integral image allows to do that with only six (resp. eight) operations.

Another important feature used with Viola and Jones algorithm is the Local Binary Patterns (LBP) [80]. Local Binary Patterns belong to the category of attributes based on spatial modelling of textures. The purpose of the LBP operator is to assign to each pixel of the image a value characterizing the local pattern of the neighbourhood 3×3 of the pixel. A threshold is carried out over the whole neighbourhood of the considered pixel: if the value of the neighbouring pixel is less than that of the central pixel, the result is 0, otherwise 1. A number in binary code can then be deduced and is associated to the considered pixel. An example is given in Figure A.4. A 256-bin histogram can be computed over an image and each bin corresponds to a feature.

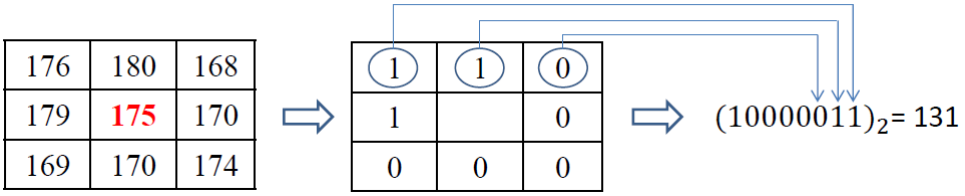


Figure A.4 – Example of LBP feature computation.

A.2.2 AdaBoost

Viola and Jones were the first to introduce a real-time face detection method based on the boosting learning system called Adaboost (Adaptive Boosting) [45]. Its general principle consists in constructing a *strong* classifier from a weighted combination of *weak* classifiers. A weak classifier simply corresponds to the computation of a feature (such as Haar feature) on which a threshold is applied, and thus returns -1 or 1.

Initially, all weights are initialized equally. At each iteration t, the best weak classifier is found, *i.e.*, the classifier h_t which minimizes the classification error ϵ_t. A parameter λ_t measuring the importance that this weak classifier will have in the final combination is calculated. The weights are then updated: more importance is given to the samples which have been poorly classified by this weak classifier, *i.e.*, their weight is increased, and similarly the well classified ones will have a weaker weight. The final (strong) classifier corresponds to the linear combination of the weak classifiers obtained at each iteration, weighted by their coefficient.

The procedure of the Discrete Adaboost is described on the Algorithm 3. Over the years, variants of this Adaboost algorithm have been proposed in order to improve the performance, such as the Real Adaboost [92] or Gentle Adaboost [46]. It has been shown that this latter variant outperforms the other variants in many cases.

A.2.3 Cascade structure

Viola and Jones introduced the notion of cascade in order to find a compromise between computation time and performance [109]. This structure is motivated by the fact that on average, only 0.01% of all windows of an image correspond to a face, the rest being no interest background. Thus, the algorithm must spend most time only on potentially positive windows.

Stages in the cascade are constructed by training classifiers using AdaBoost. The principle of the cascade of classifiers is similar to the decision trees: on each stage, either the strong classifier rejects the sample and the process stops, *i.e.*, the window is classified as non-face, or it is accepted and is transmitted to the next stage, and

Algorithm 3 Adaboost algorithm

Require: training dataset $\{x_1, y_1\}, ..., \{x_p, y_p\}$, maximal number of iteration T, weak classifiers $h : x \to [-1, 1]$.

Weights initialization : $w_1^{(i)} = \frac{1}{p}, i = 1, ..., p$.

for $t = 1$ to T **do**

- Find the weak classifier having the smallest error classification:

$$\epsilon_t = \sum_{\substack{i=1 \\ y_i \neq h_t(x_i)}}^{p} w_t^{(i)}$$

.

- Compute the coefficient $\lambda_t = \frac{1}{2} log(\frac{1-\epsilon_t}{\epsilon_t})$.
- Update the weights

$$\begin{aligned} w_{t+1}^{(i)} &= \frac{w_t^{(i)}}{Z_t} \times \begin{cases} e^{-\lambda_t} \text{ if } h_t(x_i) = y_i, \\ e^{\lambda_t} \text{ if } h_t(x_i) \neq y_i, \end{cases} \\ &= \frac{w_t^{(i)}}{Z_t} e^{(-\lambda_t y_i h_t(x_i))}, \end{aligned}$$

with Z_t a normalization factor.

end for

Final strong classifier is defined by:
$H(x) = sign(\sum_{t=1}^{T} \lambda_t h_t(x))$.

so on. The first stage aims to eliminate many of the easiest cases, thus a majority of windows without faces is quickly eliminated. The further down one goes on the cascade, the more the number of features used is big, and therefore the more the classifier is discriminating. This idea is schematized in Figure A.5. This idea of cascade has been

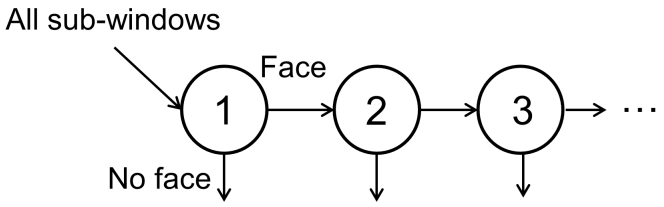


Figure A.5 – Illustration of the cascade architecture.

used in many other algorithms to reduce time processing such as with neural networks [59] and with SVM [117, 127].

A complete algorithmic description of the Viola-Jones algorithm can be found in [112]. This algorithm has provoked a lot of enthusiasm within the computer vision community, and still remains today a famous and widely used method for detecting a face in an image. Extensions and improvements have been proposed over the years, such as for instance for the time processing of the algorithm [50] or its performance [71].

A.3 HOG+SVM

Another widely applied method for object detection is the approach combining the Support Vector Machine (SVM) as training classifier and Histogram of Oriented Gradient (HOG) as feature. This HOG+SVM method was first introduced by Dalal and Triggs in [22], in the field of human detection. In fact, this approach becomes very popular to detect pedestrians [102, 48, 111, 83], but is also used for face detection [82, 18, 117]. Furthermore, a recent and popular object detection method, called Deformable Part Model (DPM) [42], uses this algorithm.

The principle of the feature called Histogram of Oriented Gradients is first detailed in Section A.3.1, followed by the training algorithm called SVM in Section A.3.2.

A.3.1 Histogram of Oriented Gradients

The descriptors called Histogram of Oriented Gradients (HOG) are often used for object detection [102, 111, 127]. In particular, Dalal and Triggs showed that these

features performed better than other features for pedestrian detection [22]. They are also used to detect faces [18, 89].

The first step to compute this feature in a given image x_i of the training set $\mathcal{S}$ is to calculate the gradient values. To do this, a derivative mask is generally applied in one or two directions (horizontal and vertical), *i.e.*, the $[-1, 0, 1]$ and $[-1, 0, 1]^T$ median filters are applied to the image. Thus, for each pixel, the gradient direction (phase) and gradient magnitude (norm) are obtained. If a RGB color image is used, the filtering is performed on each of the three components and, for each pixel, the gradient with the highest norm is kept.

Then, a grid of the image is defined. The second step consists in creating an histogram for each cell of this grid: each pixel of a cell votes for a class of the histogram according to the orientation of the gradient, and its vote corresponds to its magnitude. For instance, Dalal and Triggs used cells composed of 8x8 pixels and 9-bin histograms ranging from 0 to 180 degrees.

To account for changes in illumination and contrast, a step of gradient normalization is performed. For this purpose, the authors group together several cells into a same block, and normalization is performed on these blocks. As the blocks overlap, a same cell participates several times in the final descriptor, as a member of different blocks. This cell and blocks repartition is illustrated in Figure A.6.

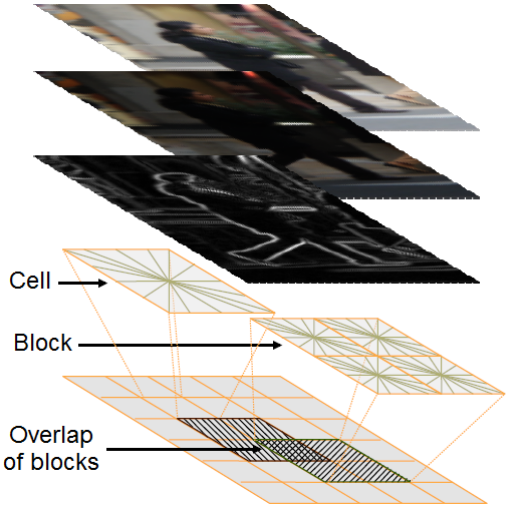


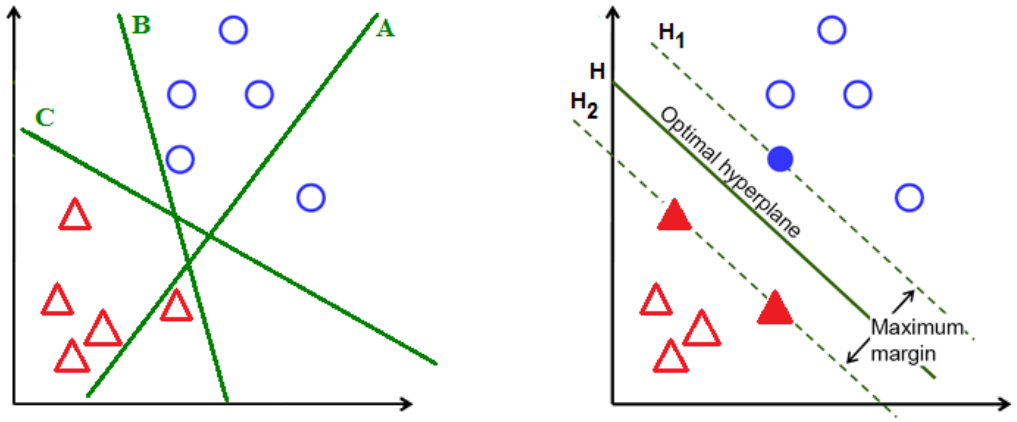
Figure A.6 – Illustration of the grid to compute HOG [22].

Finally, a HOG descriptor, which consists of all the cell histograms for each block in the image, is assigned to each image of the training set. This set of descriptors is then used as input to train a SVM, detailed in the next section.

A.3.2 Support Vector Machine

Support Vector Machines (SVM) were initially introduced by Vapnik *et al.* [21, 95], and first applied to the problem of face detection by Osuna *et al.* [82]. The purpose of SVM is the same as boosting methods and as any learning algorithms: to train a classifier that offers the best possible classification performance.

The principle of SVM consists in finding a hyperplane that separates the object class from the non-object class while maximizing the margin between these two classes. Let us first consider the case where the training data are linearly separable; in that case, there exists a hyperplane which separates the positive from the negative examples. Figure A.7a illustrates examples of three hyperplanes for some given training data. As it can be seen, the hyperplane A does not separate the two classes, contrary to hyperplanes B and C. Finding the best hyperplane between B and C consists in choosing the one that maximizes more the distances between him and the nearest data point (either class). Thus, C is better than B.



(a) Different hyperplanes. (b) Optimal hyperplane.

Figure A.7 – Finding the optimal hyperplane.

Figure A.7b shows the illustration of the optimum hyperplane, which maximizes the margin between the two classes. In particular, H_1 and H_2 are the two parallel hyperplanes that separate the two classes of data so that the distance between them (called the margin) is as large as possible. The desired maximum-margin hyperplane is the hyperplane that lies halfway between them. We may note that the points belonging to these two hyperplanes, represented by the filled triangles and circles in Figure A.7b, are called the Support Vectors. They are the elements of the training dataset that would change the position of the hyperplane H if they were changed or removed. These two particular hyperplanes can be described by

$$H_1 : wx_i - b = 1, \quad H_2 : wx_i - b = -1, \tag{A.2}$$

where w is a weight vector and b the bias. Using the distance between a point and a

hyperplane, we obtain that the distance between these two hyperplanes is $2 \times 1/||w|| = 2/||w||$. The solution consists in finding the maximum margin by minimizing $\|w||$, subject to the constraint $y_i(x_i w + b) \geq 1$ for $i = 1, ..., p$. This optimization problem can be performed using Lagrangian method, which gives the optimal parameters $\hat{w}$ and $\hat{b}$ of the optimal hyperplane. Then, given a test feature x, its class is given by the sign of $\hat{w}x + \hat{b}$.

Yet, we may notice that this linear SVM cannot solve classification problems where the data are non-linearly separable. An extension to the non-lineraly case was proposed by mapping data to higher dimensions (changing the feature representation) [12].

A.4 Artificial Neural Networks

Artificial Neural Networks (ANN) refers to a family of machine learning algorithms inspired by the way nervous system, such as the brain, processes the information. It is composed of a large number of interconnected elements, corresponding to the neurones of a brain. They were first proposed in 1943 by W. S. McCulloch, a neuroscientist, and W. Pitts, a logician. They described the concept of a neuron: a single cell, which takes part of a network of multiple cells, that receives and processes inputs and generates an output. Applications of neural networks to vision problems are not recent; in particular, they always have been a popular approach for face detection [90, 88]. Yet, it was not until recently that they emerged as highly successful on many applications in the form of *deep* neural networks, and in particular in the classification field [96, 66, 98, 91, 103]. They are thus some current major approaches for face detection [123, 38, 59, 55]. An historical survey compactly summarizes relevant works on neural networks since the 1940s up until now in [94].

The main difference between neural network-based approaches and other object detector algorithms is that the traditional approaches are based on extraction of features such as Haar or HOG (features designed by human engineers), while the neural networks do not make any assumptions about the features to extract. Neural network models are capable of learning to focus on the right features by themselves.

Neural networks are typically organized in layers. There is first an input layer, which communicates to one or more other layers called hidden layers, linked finally to an output layer. An architecture of a typical feed-forward neural network composed of two hidden layers is illustrated in Figure A.8. The term feed-forward indicates that, except during training, the links extend in only one direction (from the input layer to the output layer).

Each layer is composed of a number of nodes, which correspond to "brain neurons". A neuron is a computational unit that produces an output called activation

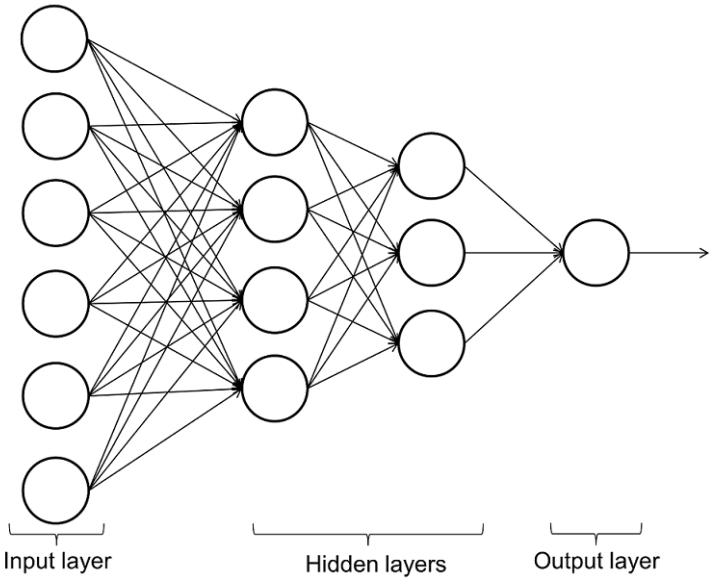


Figure A.8 – Illustration of a neural network architecture.

based on a set of inputs and associated weights, and a bias term. It is performed using a function called activation function. The final layer's activations are the predictions that the network actually makes. Multiple activation functions have been defined over the years such as for instance the sigmoid function, that we exposed in Chapter 2, or the hyperbolic tangent (tanh) function.

Training a neural network consists in finding the right set of weights for all of the connections, to make the right decisions. The most classic way to train a neural network is to use back-propagation algorithm [70]. First, the weights of the network are randomly initialized. At each iteration, *i.e.*, for each training data, the data is presented to network and the unit outputs are calculated. For all layers, starting with the output layer back to the input layer (hence the term *back*-propagation), the network output is compared to the correct output through an error function and the weights are updated according to the result.

Deep Neural Networks (DNN) is the name used for networks composed of several (more than one) hidden layers. The processing power of the computers have considerably increased these last years, and larger and larger datasets have been recently made available. For instance, Facebook can use all the faces tagged on the photos posted by the billion users it currently has. These factors enable to train deep neural networks with more and more hidden layers, and the more hidden layers are, the more complex the features the nodes can recognize. For instance, ConvNet architecture may have 10 to 20 layers, with millions of connections between units [66] and more recently, the authors of [49] proposed a network with a depth of up to 152 layers.

Despite their outstanding performance, deep neural networks present some

disadvantages: they require a large amount of data, they are extremely computationally expensive to train, and more importantly, the learning process is considered as a black box, we do not know the features that are learned, as there is no proper defined mathematical model.

References

[1] P. Aarabi, J. C. L. Lam, and A. Keshavarz. Face detection using information fusion. In *Proceedings of the 10th International Conference on Information Fusion*, pages 1–8, Quebec, Canada, July 2007.

[2] C. C. Aggarwal and S. Y. Philip. A survey of uncertain data algorithms and applications. *Transactions on Knowledge and Data Engineering*, 21(5):609–623, 2009.

[3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2001.

[4] M. Ayer, H. D. Brunk, G. M. Ewing, W. T. Reid, and E. Silverman. An empirical distribution function for sampling with incomplete information. *Annals of mathematical statistics*, 26(4):641–647, 1955.

[5] K. Bache and M. Lichman. UCI machine learning repository. University of California, School of Information and Computer Sciences, 2013. http://archive.ics.uci.edu/ml.

[6] S. C. Bagley, H. White, and B. A. Golomb. Logistic regression in the medical literature: Standards for use and reporting, with particular attention to one medical domain. *Journal of clinical epidemiology*, 54(10):979–985, 2001.

[7] Y. Bar-Shalom. Multitarget-multisensor tracking: advanced applications. *Norwood, MA, Artech House, 1990, 391 p.*, 1990.

[8] Y. Bar-Shalom, T. E. Fortmann, and P. G. Cable. Tracking and data association. *The Journal of the Acoustical Society of America*, 87(2):918–919, 1990.

[9] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA journal of numerical analysis*, 8(1):141–148, 1988.

[10] Y. Bi, J. Guan, and D. Bell. The combination of multiple classifiers using an evidential reasoning approach. *Artificial Intelligence*, 172(15):1731–1751, 2008.

[11] G. Bishop and G. Welch. An introduction to the Kalman filter. *Proc of SIGGRAPH*, 8(27599-23175):41, 2001.

[12] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, Pittsburgh, Pennsylvania, USA, July 1992. ACM.

[13] G. Bradski. The OpenCV library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000.

[14] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 1998.

[15] J. Brand and J. S. Mason. A comparative assessment of three approaches to pixel-level human skin-detection. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 1, pages 1056–1059, Barcelona, Spain, Sept. 2000.

[16] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[17] A. S. Capelle, C. Fernandez-Maloigne, and O. Colot. Segmentation of brain tumors by evidence theory: on the use of the conflict information. In *International Conference on Information Fusion*, pages 264–271, Stockholm, Sweden, June 2004.

[18] L. R. Cerna, G. Cámara-Chávez, and D. Menotti. Face detection: Histogram of oriented gradients and bag of feature method. In *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, page 1, Las Vegas, Nevada, USA, July 2013.

[19] C-C. Chang and C-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[20] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 25(5):564–577, 2003.

[21] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[22] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893, San Diego, California, June 2005.

[23] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38:325–339, 1967.

[24] A. P. Dempster. Upper and lower probabilities generated by a random closed interval. *Annals of Mathematical Statistics*, 39(3):957–966, 1968.

[25] A.P. Dempster. New methods for reasoning towards posterior distributions based on sample data. *Annals of Mathematical Statistics*, 37(2):355–374, 1966.

[26] T. Denœux. Analysis of evidence-theoretic decision rules for pattern classification. *Pattern Recognition*, 30(7):1095–1107, 1997.

[27] T. Denœux. A neural network classifier based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 30(2):131–150, 2000.

[28] T. Denœux. The cautious rule of combination for belief functions and some extensions. In *Proceedings of the 9th International Conference on Information Fusion*, pages 1–8, Florence, Italy, July 2006.

[29] T. Denœux. Likelihood-based belief function: justification and some extensions to low-quality data. *International Journal of Approximate Reasoning*, 55(7):1535–1547, 2014.

[30] T. Denœux, O. Kanjanatarakul, and S. Sriboonchitta. Ek-nnclus: a clustering procedure based on the evidential k-nearest neighbor rule. *Knowledge-Based Systems*, 88:57–69, 2015.

[31] T. Denœux and M-H Masson. Evclus: evidential clustering of proximity data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):95–109, 2004.

[32] T. Denœux and Ph. Smets. Classification using belief functions: relationship between case-based and model-based approaches. *IEEE Transactions on Systems, Man and Cybernetics B*, 36(6):1395–1406, 2006.

[33] T. Denœux, N. El Zoghby, V. Cherfaoui, and A. Jouglet. Optimal object association in the Dempster–Shafer framework. *IEEE transactions on cybernetics*, 44(12):2521–2531, 2014.

[34] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *Proceedings of the British Machine Vision Conference*, pages 91.1 – 91.11, 2009.

[35] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

[36] R. P. W. Duin. The combining classifier: to train or not to train? In *Proceedings of the 16th International Conference on Pattern Recognition, Quebec City, Quebec, Canada, August, 2002*, volume 2, pages 765–770, 2002.

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

[38] S. S. Farfade and M. J. Saberian andL J. Li. Multi-view face detection using deep convolutional neural networks. In *Proceedings of the 5th International Conference on Multimedia Retrieval (ICMR)*, pages 643–650, Shanghai, China, June 2015.

[39] S. S. Farfade, M. Saberian, and L.-J. Li. Multi-view face detection using deep convolutional neural networks. In *Proceedings of the International Conference on Multimedia Retrieval*, Shanghai, China, June 2015.

[40] F. Faux and F. Luthon. Robust face tracking using colour Dempster-Shafer fusion and particle filter. In *9th International Conference on Information Fusion*, pages 1–7, Florence, Italy, July 2006. IEEE.

[41] F. Faux and F. Luthon. Theory of evidence for face detection and tracking. *International Journal of Approximate Reasoning*, 53(5):728–746, 2012.

[42] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[43] Y. Freund. Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285, 1995.

[44] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, volume 96, pages 148–156, Bari, Italy, 1996.

[45] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[46] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 28(2):337–407, 2000.

[47] A. Hadid, M. Pietikäinen, and T. Ahonen. A discriminative feature space for detecting and recognizing faces. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–797, Washington, DC, June 2004.

[48] F. Han, Y. Shan, R. Cekander, H. S. Sawhney, and R. Kumar. A two-stage approach to people and vehicle detection with HOG-based SVM. In *Performance Metrics for Intelligent Systems (PerMIS) Workshop*, pages 133–140, Gaithersburg, Maryland, USA, Aug. 2006.

[49] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, Las Vegas, Nevada, USA, June 2016.

[50] D. Hefenbrock, J. Oberg, N. T. N. Thanh, R. Kastner, and S. B. Baden. Accelerating Viola-Jones face detection to FPGA-level using GPUs. In *18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pages 11–18, Charlotte, North Carolina, USA, May 2010. IEEE.

[51] E. Hjelmås and B. K. Low. Face detection: A survey. *Computer vision and image understanding*, 83(3):236–274, 2001.

[52] D.W. Hosmer, S. Lemeshow, and R. X. Sturdivant. *Applied logistic regression*, volume 398. Wiley & Sons, 2013.

[53] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[54] V. Jain and E. Learned-Miller. FDDB: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, Univ. Massachusetts, 2010.

[55] H. Jiang and E. Learned-Miller. Face detection with the faster R-CNN. In *12th IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, pages 650–657, Washington, DC, USA, May 2017. IEEE.

[56] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.

[57] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[58] P. Kakumanu, S. Makrogiannis, and N. Bourbakis. A survey of skin-color modeling and detection methods. *Pattern recognition*, 40(3):1106–1122, 2007.

[59] I. A. Kalinovskii and V. G. Spitsyn. Compact convolutional neural network cascade for face detection. In *Proceedings of the 10th Annual International Scientific Conference on Parallel Computing Technologies (PCT)*, volume 1576, pages 375–387, Arkhangelsk, Russia, March 2016.

[60] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[61] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Journal of basic engineering*, 83(1):95–108, 1961.

[62] O. Kanjanatarakul, T. Denœux, and S. Sriboonchitta. Prediction of future observations using belief functions: A likelihood-based approach. *International Journal of Approximate Reasoning*, 72:71–94, 2016.

[63] O. Kanjanatarakul, S. Sriboonchitta, and T. Denœux. Forecasting using belief functions: an application to marketing econometrics. *International Journal of Approximate Reasoning*, 55(5):1113–1128, 2014.

[64] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[65] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[66] A. Krizhevsky and I. Sutskeverand G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[67] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(1-2):83–97, 1955.

[68] L. I. Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2004.

[69] B. Lelandais, S. Ruan, T. Denœux, P. Vera, and I. Gardin. Fusion of multi-tracer pet images for dose painting. *Medical image analysis*, 18(7):1247–1259, 2014.

[70] H. Leung and S. Haykin. The complex backpropagation algorithm. *IEEE Transactions on Signal Processing*, 39(9):2101–2104, 1991.

[71] Q. Li, U. Niaz, and B. Merialdo. An improved algorithm on Viola-Jones object detector. In *10th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 1–6. IEEE, 2012.

[72] Y. Lu, J. Zhou, and S. Yu. A survey of face detection, extraction and recognition. *Computing and informatics*, 22(2):163–195, 2012.

[73] G. C. Luh. Face detection using combination of skin color pixel detection and Viola-Jones face detector. In *Proceedings of the International Conference on Machine Learning and Cybernetics*, volume 1, pages 364–370, Lanzhou, China, July 2014.

[74] S. Yangand P. Luo, C-C. Loy, and X. Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5525–5533, Las Vegas, Nevada, USA, July 2016.

[75] D. Mercier, G. Cron, T. Denœux, and M-H. Masson. Decision fusion for postal address recognition using belief functions. *Expert Systems with Applications*, 36(3):5643–5653, 2009.

[76] P. Minary, F. Pichon, D. Mercier, E. Lefevre, and B. Droit. An evidential pixel-based face blurring approach. In *Proceedings of the 4th International Conference on Belief Functions*, Lecture Notes in Computer Science, pages 222–230, Prague, Czech Republic, Sept. 2016. Springer.

[77] P. Minary, F. Pichon, D. Mercier, E. Lefevre, and B. Droit. Evidential joint calibration of binary svm classifiers using logistic regression. In *Proceedings of the 11th International Conference on Scalable Uncertainty Management*, Lecture Notes in Artificial Intelligence, pages 405–411, Granada, Spain, October 2017. Springer.

[78] P. Minary, F. Pichon, D. Mercier, E. Lefevre, and B. Droit. Face pixel detection using evidential calibration and fusion. *International Journal of Approximate Reasoning*, 91:202–215, December 2017.

[79] H. T. Nguyen. *An Introduction to Random Sets*. Chapman and Hall/CRC press, 2006.

[80] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

[81] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *European conference on computer vision*, pages 28–39, Prague, Czech Republic, May 2004. Springer.

[82] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 130–136, San Juan, Puerto Rico, June 1997.

[83] Y. Pang, Y. Yuan, X. Li, and J. Pan. Efficient HOG human detection. *Signal Processing*, 91(4):773 – 781, 2011.

[84] M. Pantic and L. J. M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on pattern analysis and machine intelligence*, 22(12):1424–1445, 2000.

[85] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[86] B. Quost, M-H. Masson, and T. Denœux. Classifier fusion in the Dempster–Shafer framework using optimized t-norm based combination rules. *International Journal of Approximate Reasoning*, 52(3):353–374, 2011.

[87] E. Ramasso, C. Panagiotakis, D. Pellerin, and M. Rombaut. Human action recognition in videos based on the Transferable Belief Model. *Pattern analysis and Applications*, 11(1):1–19, 2008.

[88] F. Raphael, J. B. Olivier, and J-E. Viallet. A fast and accurate face detector based on neural networks. *IEEE Transactions on pattern analysis and machine intelligence*, 23(1):42–53, 2001.

[89] N. Rekha and M. Z. Kurian. Face detection in real time based on HOG. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 3:1345–1352, 2014.

[90] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 20(1):23–38, 1998.

[91] T. N. Sainath, A-R. Mohamed, B. Kingsbury, and B. Ramabhadran. Deep convolutional neural networks for lvcsr. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8614–8618, 2013.

[92] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine learning*, 37(3):297–336, 1999.

[93] R.E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[94] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[95] B. Schölkopf, P. Simard, V. Vapnik, and A.J. Smola. Improving the accuracy and speed of support vector machines. *Advances in neural information processing systems*, 9:375–381, 1997.

[96] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *Proceedings of the International Conference on Learning Representations*, 2014.

[97] G. Shafer. *A mathematical theory of evidence*, volume 1. Princeton University Press, 1976.

[98] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[99] Ph. Smets. Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem. *International Journal of approximate reasoning*, 9(1):1–35, 1993.

[100] Ph. Smets and R. Kennes. The Transferable Belief Model. *Artificial Intelligence*, 66:191–243, 1994.

[101] M. Soriano, B. Martinkauppi, S. Huovinen, and M. Laaksonen. Using the skin locus to cope with changing illumination conditions in color-based face tracking. In *Proceedings of the Nordic Signal Processing Symposium*, volume 38, pages 383–386, Kolmarden, Sweden, June 2000.

[102] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi. Pedestrian detection using infrared images and histograms of oriented gradients. In *Intelligent Vehicles Symposium*, pages 206–212. IEEE, 2006.

[103] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *Proceedings of the 27th Conference on Neural Information Processing Systems (NIPS)*, pages 2553–2561, Lake Tahoe, Nevada, USA, Dec. 2013.

[104] Z. S. Tabatabaie, R. W. Rahmat, N. I. B. Udzir, and E. Kheirkhah. A hybrid face detection system using combination of appearance-based and feature-based methods. *International Journal of Computer Science and Network Security*, 9(5):181–185, 2009.

[105] A. Fallah Tehrani, W. Cheng, K. Dembczyński, and E. Hüllermeier. Learning monotone nonlinear models using the Choquet integral. *Machine Learning*, 89(1):183–211, Oct 2012.

[106] J-C. Terrillon, M. N. Shirazi, H. Fukamachi, and S. Akamatsu. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In *Proceedings of the fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 54–61, Grenoble, France, March 2000.

[107] S. Tulyakov, S. Jaeger, V. Govindaraju, and D. Doermann. Review of classifier combination methods. In S. Marinai and H. Fujisawa, editors, *Machine Learning in Document Analysis and Recognition*, pages 361–386. Springer, 2008.

[108] V. Vezhnevets, V. Sazonov, and A. Andreeva. A survey on pixel-based skin color detection techniques. In *Proceedings of the Graphicon Conference*, volume 3, pages 85–92. Moscow, Russia, 2003.

[109] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 511–518, Kauai, Hawaii, Dec. 2001.

[110] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[111] X. Wang, T. X. Han, and S. Yan. An HOG-LBP human detector with partial occlusion handling. In *12th International Conference on Computer Vision (ICCV)*, pages 32–39, Kyoto, Japan, Sept. 2009. IEEE.

[112] Y-Q. Wang. An analysis of the Viola-Jones face detection algorithm. *Image Processing On Line*, 4:128–148, 2014.

[113] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 529–534, Colorado Springs, Colorado, USA, June 2011. IEEE.

[114] P. Xu, F. Davoine, and T. Denœux. Evidential combination of pedestrian detectors. In *Proceedings of the 25th British Machine Vision Conference*. BMVA Press, 2014.

[115] P. Xu, F. Davoine, and T. Denœux. Evidential multinomial logistic regression for multiclass classifier calibration. In *Proceedings of the 18th International Conference on Information Fusion*, pages 1106–1112, Washington, DC, USA, July 2015.

[116] P. Xu, F. Davoine, H. Zha, and T. Denœux. Evidential calibration of binary SVM classifiers. *International Journal of Approximate Reasoning*, 72:55–70, 2016.

[117] H-C. Yang and X. A. Wang. Cascade face detection based on histograms of oriented gradients and support vector machine. In *10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, pages 766–770, Krakow, Poland, Nov. 2015. IEEE.

[118] M-H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

[119] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys (CSUR)*, 38(4):13, 2006.

[120] B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Proceedings of the International Conference on Machine Learning*, pages 609–616, Williamstown, Massachusetts, June 2001.

[121] B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth International Conference on Knowledge Discovery and Data Mining*, pages 694–699, Edmonton, AB, Canada, July 2002.

[122] S. Zafeiriou, C. Zhang, and Z. Zhang. A survey on face detection in the wild: past, present and future. *Computer Vision and Image Understanding*, 138:1–24, 2015.

[123] C. Zhang and Z. Zhang. Improving multiview face detection with multi-task deep convolutional neural networks. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 1036–1041, Colorado, USA, March 2014.

[124] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys (CSUR)*, 35(4):399–458, 2003.

[125] W. Zhong and J. T. Kwok. Accurate probability calibration for multiple classifiers. In *Proceedings of the Twenty-Third international Joint Conference on Artificial Intelligence*, pages 1939–1945, Beijing, China, August 2013.

[126] Z-H. Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[127] Q. Zhu, M-C. Yeh, K-T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1491–1498, New York, NY, USA, June 2006.