Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Fusion of supervised classifiers using theory of evidence

F. Rhéaume
A.-L. Jousselme
É . Bossé
DRDC Valcartier

Defence R&D Canada – Valcartier
Technical Report
DRDC Valcartier TR 2003-318
September 2006

Canada

Fusion of supervised classifiers using theory of evidence

F. Rhéaume
A.-L. Jousselme
É. Bossé
Defence R&D Canada - Valcartier

Defence R&D Canada - Valcartier

Technical Report

DRDC Valcartier TR 2003-318

September 2006

Principal Author

François Rhéaume

Approved by

Éloi Bossé
Head/A Section

Approved for release by

G. Bérubé
Chief Scientist

Abstract

In the field of pattern recognition, more specifically in the area of supervised and feature-vector-based classifications, various classification methods exist, but none of them is flawless given any data sources. Each classifier behaves differently, with its own strengths and weaknesses. Some are more efficient than others in particular situations. Performance of these individual classifiers can be improved by combining them into one multiple classifier. In order to make more realistic decisions, a multiple classifier may use some measures generated by each classifier and use some *a priori* knowledge, such as probability distributions, reliability rates and confusion matrices. Individual classifiers studied in this project are the Bayes, the k-nearest neighbors and the neural network classifiers. They are combined using Dempster-Shafer's theory. The problem simplifies in finding weights that best represent the evidences of individual classifier. We suggest basic probability assignments (BPAs) based on some measures which precede the decision step.

Following the study of some classical multiple classifiers in the literacy, we compare them with our approach that separates in two distinct multiple classifiers. Tests are made on three different databases that are infrared images of ships, handwritten digits and satellite images of terrains. One of the suggested multiple classifier gives better results than all other classical multiple classifiers tested in this work.

Résumé

Dans le domaine de la classification supervisée à base de vecteurs d'attributs, plusieurs méthodologies existent, certaines étant plus efficaces que d'autres selon le contexte. La performance peut être accrue en combinant les classificateurs individuels, formant ainsi un classificateur multiple. Bien que la méthode de combinaison des classificateurs puisse être généralisée, nous présentons dans ce travail un classificateur multiple issu de la combinaison des trois classificateurs simples suivants : un classificateur bayesien, un classificateur des k plus proches voisins et un réseau de neurones (MLP). Afin de prendre en compte l'incertitude laissée par un classificateur simple sur la classe d'appartenance d'un objet, nous utilisons ici une approche basée sur la théorie de l'évidence de Dempster-Shafer. Ainsi, nous proposons des fonctions de masse initiales (BPA) basées sur les réponses de chaque classificateur simple avant décision.

Après avoir présenté quelques classificateurs multiples classiques de la littérature, nous les comparons à notre approche ayant donné naissance à deux classificateurs multiples. Des tests comparatifs sont effectués pour les applications à la reconnaissance d'images infrarouges de bateaux, de chiffres manuscrits et d'images satellites de terrains. À la lumière de ces résultats, un des deux classificateurs proposés offre les meilleurs résultats parmi tous ceux présentés dans ce rapport.

This page intentionally left blank.

Executive summary

Fusion of supervised classifiers using theory of evidence

F. Rhéaume, A.-L. Jousselme, É. Bossé ; DRDC Valcartier TR 2003-318; Defence R&D Canada - Valcartier; September 2006.

Over the years, the field of pattern classification has seen more and more systems evolved. As computer power increases, solutions for more demanding tasks like automatic object recognition or situation recognition have become realizable. Classification systems may have to analyze reports from many sources of information, or sensors, with the aim to recognize a situation or an object among a number of *a priori* known hypothesis. This refers to the supervised classification area, where the task is to determine the class of an element based on its features.

Classification algorithms relying on different theories already exist. Classifiers can react differently facing a variety of circumstances. One can be highly effective in some specific circumstances but vulnerable to some others. The circumstances may include the nature of the features, the number of features, the nature of the classes and the number of classes. Thus, flexibility and robustness are required in a classification system. Considering the latter and because the sets of objects misclassified by each classifier do not necessarily intersect, combining individual classifiers can be regarded as a way to improve global results in an application.

A multiple classifier is made of a combination algorithm that fuses the information coming from the individual classifiers. There are many combination algorithms existing and associated with different decision rules. Four of the most popular multiple classifiers are presented in this report: BKS, Lu and Xu's multiple classifiers, as well as Appriou's method for making basic probability assignments.

The main difficulty with multiple classifiers is to make each individual classifier contribute its best according to its capability. This relates to the issue of determining the information each classifier must provide in the combination. The issue can be simplified in choosing between using the measures of a classifier or using its decision made on the measures. There is also the possibility of using or not *a priori* knowledge.

Our suggested solutions aim to meet as best as possible the performance and the flexibility requirements, that are to recognize objects the most successfully in any context. The first one relies only on measures (Measure Level Based (MLB)), while the second one relies on the class decision as more as on some *a priori* knowledge (Abstract Level Based (ALB)). The information is represented and combined using Dempster-Shafer's theory. We also propose a multiple classifier that uses Appriou's model of initial mass assignment.

Tests that combine three individual classifiers, that are Bayes, k Nearest Neighbors (k-NN) and Neural Network (Multi-Layer Perceptron), were performed on all multiple classifiers studied in this report. Three different applications were tested: IR ship images classification, handwritten digit classification and Landsat satellite image classification. The Measure Level Based multiple classifier stands out as the best one in all of the three applications.

Sommaire

Fusion of supervised classifiers using theory of evidence

F. Rhéaume, A.-L. Jousselme, É. Bossé ; DRDC Valcartier TR 2003-318; R&d pour la défense Canada - Valcartier; septembre 2006.

Au cours des années, le champ de la classification de données a vu de plus en plus de systèmes développés. Avec la puissance toujours plus grandissante des ordinateurs, des solutions à des tâches plus exigeantes sont devenues réalisables, comme la reconnaissance automatique d'objets ou la reconnaissance de situation. De tels systèmes analysent les rapports de plusieurs sources d'information pour arriver à reconnaître une situation parmi un certain nombre d'hypothèses connues *a priori*. Ceci a trait à la classification supervisée où le but est de déterminer la classe d'un élément d'après ses attributs représentés dans un vecteur.

Des algorithmes de classification basés sur différentes théories existent déjà. Les classificateurs peuvent réagir différemment face à des situations changeantes. Certains peuvent être très efficaces dans une situation particulière, mais médiocres dans d'autres. Les changements peuvent être dans la nature des attributs, dans le nombre d'attributs, dans la nature des classes ou dans leur nombre. Ainsi, la flexibilité et la robustesse sont des aspects dont il faut tenir compte en classification de données. Cela pris en compte et parce que les ensembles des objets mal classifiés pour chaque classificateur ne se superposent pas nécessairement, la combinaison des classificateurs simples peut être envisagée afin d'améliorer la performance globale pour une application.

Un classificateur multiple est défini comme un algorithme qui fusionne l'information de classificateurs individuels. Plusieures techniques de combinaison existent et peuvent être associées à différentes règles de décision. Quatre des classificateurs multiples les plus notoires de la littérature sont présentés dans ce rapport : BKS, Lu et Xu ainsi que la méthode d'Appriou pour produire les fonctions de masses initiales.

L'enjeu majeur dans la combinaison de classificateurs est d'arriver à faire contribuer chaque classificateur au mieux de sa capacité. Cela soulève la question sur l'information que chaque classificateur doit fournir pour la combinaison. De manière générale, cette question se reduit à choisir entre utiliser les mesures d'un classificateur ou utiliser la décision qui en ressort. Aussi s'ajoute la possibilité d'utiliser ou non des connaissances *a priori*.

Les solutions suggérées visent à répondre le mieux possible aux exigences de performance et de flexibilité, c'est-à-dire reconnaître tout objet le plus souvent possible dans quelque situation qu'il soit. La première solution se sert seulement des mesures, tandis que la deuxième utilise la décision et des connaissances *a priori*. L'information est représentée et combinée à l'aide de la théorie de Dempster-Shafer. Un classificateur multiple qui utilise le modèle d'Appriou pour la création des fonctions de masses initiales est proposé.

Des tests combinant trois classificateurs individuels, soit celui de Bayes, des k plus proches voisins et du réseau de neurones (perceptron multicouche), ont été exécutés sur tous les clas-

sificateurs multiples étudiés dans ce rapport. Trois applications différentes ont été retenues : la classification d'images infrarouges de bateaux, la classification de chiffres manuscrits et la classification d'images satellite Landsat. Le classificateur multiple basé sur les mesures s'est révélé le meilleur pour ces trois applications.

Table of contents

List of figures

List of tables

1 Introduction

The increase of computation power over the years has led to new more sophisticated applications. Artificial intelligence is an area that benefits from that in many ways. In particular, pattern classification has seen more and more systems evolved, being able to respond to more demanding tasks, like automatic object recognition or situation recognition. Such systems analyze reports from many sources of information, or sensors, with the aim to recognize a situation among a number of *a priori* known hypothesis.

Among the several applications likely to use pattern classification systems, some exist in the military area like, for example, the automatic recognition of targets using infrared imagery [1], in the security area like audiovisual person recognition [2, 3], in medicine, in handwritten character recognition [4, 5, 6, 7, 8, 9] as well as in botany with the classification of plants [10]. All of these require the use of supervised classification methods, where feature-vector based classifiers are used. With feature-vector based classifiers, the features used may vary with each application and they are the inputs of the classifiers. The task of a classifier is to determine the class of an element according to its feature vector. When the number of different classes for a particular problem is known, the classifier is a supervised one and, at the opposite, it is said to be unsupervised when the number of classes is unknown.

Classification algorithms relying on different theories already exist, as do the Bayes classifier, the k nearest neighbors classifier and the neural network classifier. Because they use various methodologies and different data characteristics, some classifiers can react differently when they are faced to a given situation in a particular application. Moreover, although some classifiers may succeed better than others in some cases, none is perfect and none can classify any data without mistakes. Each classifier has its own strengths and weaknesses. One can be highly effective in a determinate situation but vulnerable to some others. Moreover, the results of the classifiers must be good not only on a restricted set of data but also on the entire range of possible inputs, no matter what they are. Thus, flexibility and robustness are required in a classification system. Taking that into account and considering that the sets of objects misclassified by each classifier do not necessarily intersect, combining individual classifiers can be regarded as a way to improve global results in an application.

The combination of individual classifiers produces a multiple classifier. A multiple classifier has a combination algorithm that fuses the information coming from the individual classifiers. There are many combination algorithms existing, that are associated to different decision rules. Classifiers using weighted majority vote [11], majority vote [9, 12], Bayes rule [13], knowledge spaces [8] and Dempster-Shafer's theory [12, 14, 15, 16] have been developed. Each of these methods is supported with a particular theoretical basis and therefore each method behaves distinctively according to its theoretical basis. Comparative tests have been performed in most of these methods in order to figure those that do best in specific applications [9, 12].

The combination algorithm integrates the results provided by the individual classifiers. The main difficulty related to this aggregation is to make each classifier contribute its best according to its capability. This implies valuating the contribution of each one in order to take advantage of their strengths and to lessen their weaknesses. That in mind, the information each classifier provides must first be determined. Usually, the information available out of a classification algorithm divides into two dependent levels. The bottom level involves measures related to the classification algorithm. The top level is an abstraction of the bottom one and constitutes the final answer. This final answer represent the membership class of the element under identification.

Besides choosing the right information for each classifier to be combined, the information must also be represented properly and it must also be represented with a related uncertainty measure or precision degree, which is useful in the combination of classifiers. The representation of the information depends in part of the nature of the combination algorithm chosen.

The choice of the information provided by a classifier as well as its representation is a problem for which multiple solutions exist. This report is a comparison of some different multiple classifiers and a new approach is suggested. The new approach intends to meet the performance and flexibility requirements as best as possible, that is to recognize as most objects as possible in any context. In Chapter 2, some classical individual classifiers are first studied in order to understand how they can be combined. These classifiers are the Bayes classifiers, the k nearest neighbor classifier and a neural network (multi-layer perceptron). Chapter 3 presents some existing multiple classifiers among the most cited in literature : the Behavior Knowledge Space classifier [8] and two others that use Dempster-Shafer's theory, which are Lu's one [7] and Xu's one [6]. Another multiple classifier using Appriou's model of initial mass assignment [16] is also presented. Chapter 4 proposes a new combination method based on the evidential theory and emphasizes on selection and interpretation of the information in the creation of basic probability assignments. The following three chapters present some tests that compare the classifiers in different applications that are: infrared ship image recognition in Chapter 5, handwritten digits recognition in Chapter 6 and satellite image recognition in Chapter 7. Finally, concluding remarks are made in Chapter 8.

2 Supervised Classification

2.1 Introduction

Classification is a process that assigns a class to an observed object according to some characteristics. An object is defined to be anything that can be represented by some measurable features. For example, a plane can be an object belonging to one of the different classes of planes according to features like length, maximum speed, maximum altitude, etc. Let us suppose a another simple example of an application in which a given species of shark must be distinguished from a particular species of whales. After making observations in the sea, the task is to classify the species observed in either the shark class or the whale class. Figure 1 shows some examples of unclassified samples, represented by two parameters that are the length (x_1) and the weight (x_2) of the animal. Figure 2 shows the classification results for these samples, where Θ is the set of classes of all the species under consideration. Such a classification case is a supervised one because the number of classes (two) is known. Otherwise, classifications are called unsupervised when the number of classes is unknown. The process for a supervised classification is shown in Figure 3, where features are extracted from an object to be used at the input of the classifier. The classifier, following specific algorithms, outputs a class label among the set of known classes.

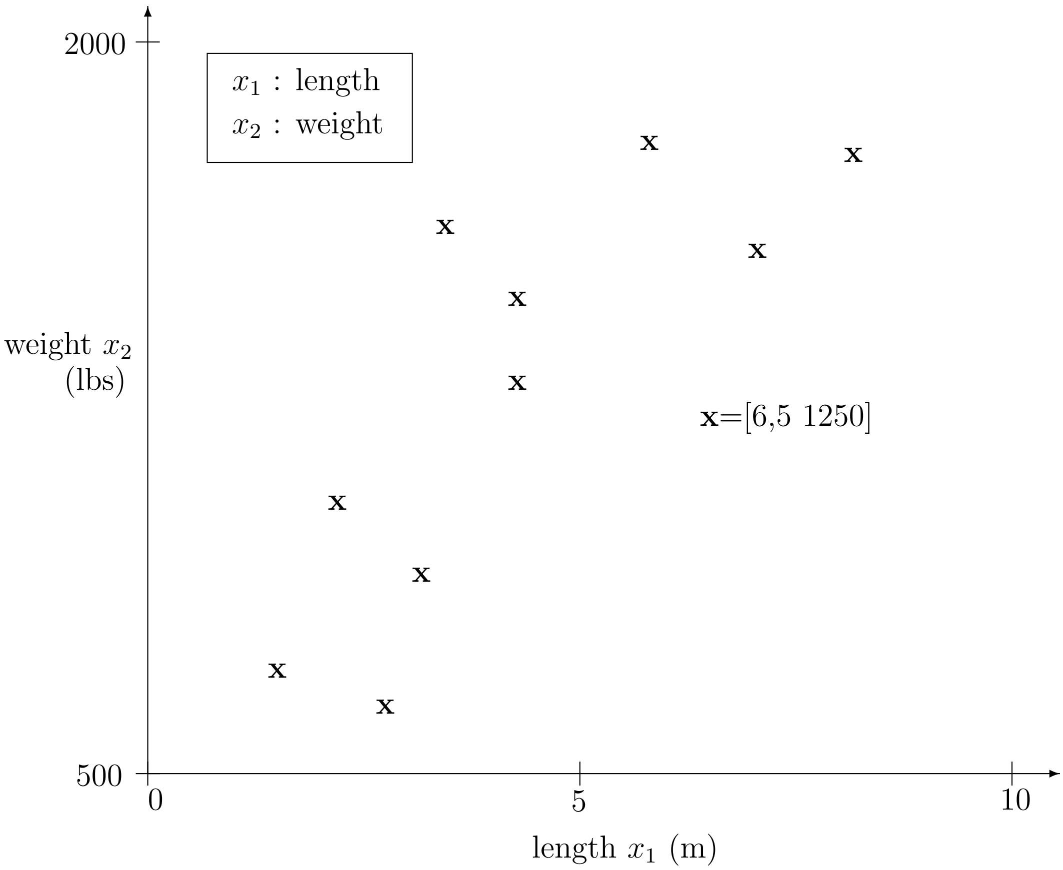


Figure 1: *Unclassified samples characterized by length (x_1) and weight (x_2)*

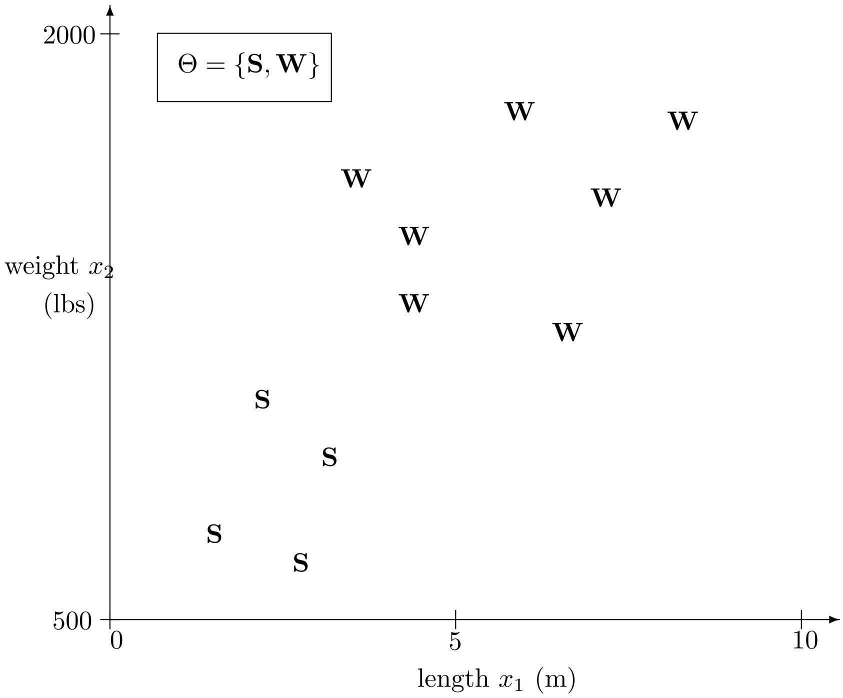


Figure 2: *Samples classified into one of the classes in* Θ, *where* Θ *comprises the shark class* ($\mathbf{S}$) *and the whale class* ($\mathbf{W}$)

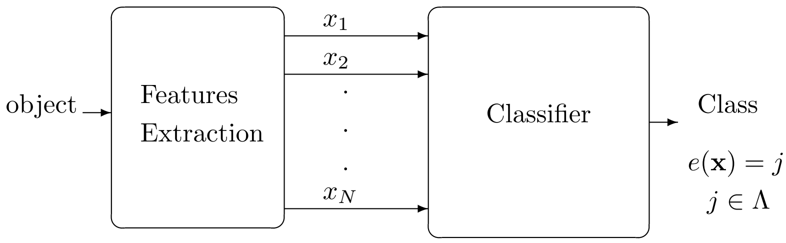


Figure 3: *Classification process*

2.2 Classical Supervised Classification Methods

2.2.1 Notations

The classifiers receive feature vectors at the input [17]. They take in a fixed length vector $\mathbf{x} = [x_1 \ x_2 \ ... \ x_N]^T$ of N parameters and they produce at the output a class labeled i from the set $\Lambda = \{1, 2, ..., M\}$, corresponding to class $C_i \in \Theta$, where $\Theta = \{C_1, C_2, ..., C_M\}$. Another class C_{M+1} is sometimes added to Θ in order to identify the rejected objects or the ignorance. In this work, the classifiers examined are the Bayes classifier, the k Nearest Neighbors classifier and the Neural Network classifier.

2.2.2 Bayes Approach

The method with Bayes classifier is about determining the probability of having a certain class given a feature value. Let I be a random variable representing one of the M classes in Λ and X_n be the random variable associated to feature n. The conditional probability of I given X_n follows Bayes' law :

$$P(I|X_n) = \frac{P(X_n|I)P(I)}{\sum_{I\in\Lambda} P(X_n|I)P(I)} \tag{1}$$

In order to get $P(I|X_n)$, *a priori* conditional probabilities $P(X_n|I)$ and *a priori* probabilities $P(I)$ must be known. The *a priori* probabilities must be obtained in a way that they are as close as possible to reality. This may be difficult to achieve depending of the application. Consequently, in practice the *a posteriori* probabilities $P(I|X_n)$ are imperfect.

Probabilities $P(X_n|I)$ can be estimated with one of the two proposed distributions: Gaussian distributions or grouped frequency distributions. In the particular case where data distribution would be similar to a normal distribution or if the available data would make it hard, for some reasons, to have a grouped frequency distribution, then a Gaussian distribution could be considered. For each class $I \in \Lambda$, Gaussian distributions for a given feature n are defined according to its average μ_n and its variance ${\sigma^2}_n$:

$$P(X_n|I) = \frac{1}{\sigma_n\sqrt{2\pi}} e^{-\frac{(X_n-\mu_n)^2}{2\sigma_n^2}} \tag{2}$$

When grouped frequency distributions do not follow a normal law, Gaussian distributions are inappropriate. In that case, grouped frequency distributions are used, where samples are cumulated over intervals. That way, for each attribute, the full range of possible values is divided into Q intervals, yielding the conditional probabilities $P(X_n = x_n^q|I = i)$, $n = 1, ..., N, q = 1, ..., Q$ and $i \in \Lambda$. In a training phase, each sample is cumulated in the interval containing the feature value. When an unknown object $\mathbf{x}$ must be identified, the M *a posteriori* probabilities $P(I = I_m|X_n = x_n^q)$ are calculated following (1), given the value x_n^q for feature n. That means $N \times M$ conditional probabilities $P(I_m|X_n)$, $n = 1, ..., N, m \in \Lambda$ must be computed .

2.2.2.1 Decision

A decision is made by computing the summed probability for each class:

$$P_s(I) = \sum_{n=1}^{N} P(I|X_n) \tag{3}$$

The class with the highest summed probability is the chosen one:

$$e_B(\mathbf{x}) = \text{Arg}\{\max_{i\in\Lambda}[P_s(I = i)]\} \tag{4}$$

Let us suppose a classification example in which an object $\mathbf{x}$ has to be classified, with $\Lambda = [1, 2, 3, 4]$ and with two features (x_1 and x_2) representing the observed object. Then there are $2 \times 4 = 8$ conditional probabilities and four summed probabilities. These are presented in the following table (by supposing $P(I = 1)$ and $P(I = 2)$ are equal to 1) :

$$\begin{array}{lll}
P(I=1|X_1)=0.1 & P(I=1|X_2)=0.2 & ,P_s(I=1)=0.3 \\
P(I=2|X_1)=0.35 & P(I=2|X_2)=0.3 & ,P_s(I=2)=0.65 \\
P(I=3|X_1)=0.4 & P(I=3|X_2)=0.15 & ,P_s(I=3)=0.55 \\
P(I=4|X_1)=0.15 & P(I=4|X_2)=0.35 & ,P_s(I=4)=0.5
\end{array}$$

Class 2 has the highest summed probability, with 0.65, so that $e_B(\mathbf{x}) = 2$.

Another method can be considered, although it seems to give lower performance according to what was done in [18]. It consists in obtaining, for each feature, the class having the highest *a posteriori* probability (5), and then performing a majority vote among the N results (6).

$$\begin{array}{ccccc}
P(I=1|X_1) & P(I=1|X_2) & ... & P(I=1|X_N) & \longrightarrow & Y[1] \\
P(I=2|X_1) & P(I=2|X_2) & ... & P(I=2|X_N) & \longrightarrow & Y[2] \\
. & & & . & & . \\
. & & & . & & . \\
. & & & . & & . \\
P(I=M|X_1) & P(I=M|X_2) & ... & P(I=M|X_N) & \longrightarrow & Y[N]
\end{array}$$

where

$$Y[j] = \text{Arg}\left\{\max_{I\in\Lambda} P(I|X_j)\right\} \tag{5}$$

Here, $Y[j]$ refers to the class with highest *a posteriori* probability for feature j. Let n_i be the number of times where $Y[j] = i$, $i = 1, ..., M$ and $j = 1, ..., N$, then the class of $\mathbf{x}$ is :

$$e_{B'}(\mathbf{x}) = \text{Arg}\{\max_{i\in\Lambda}[n_i]\} \tag{6}$$

In this previous example, the answer would include both classes 3 and 4 because there is no majority according to the two features involved.

2.2.3 k-Nearest Neighbors (k-NN)

This technique consists in selecting the k closest objects to the unknown object. Because this method implies a distance notion, we consider the problem as a geometrical one, and work in a N-dimensional space called $\mathcal{C}_N$. An object is then represented by N coordinates.

Let $\mathbf{x}$ be an element of this space, *i.e.* an object to be classified. Suppose that from a previous learning, we know a set of B vectors $\mathbf{x_1}, \ldots, \mathbf{x_B}$ with their corresponding classes. We will first select the k nearest neighbors of $\mathbf{x}$ among the B known vectors, according to a defined distance. In the general case, the distance between two elements $\mathbf{x_1}$ and $\mathbf{x_2}$ can be given by:

$$d^2(\mathbf{x_1}, \mathbf{x_2}) = (\mathbf{x_1} - \mathbf{x_2})^T \mathbf{D} (\mathbf{x_1} - \mathbf{x_2}) \tag{7}$$

where D is an $N \times N$ positively defined matrix. Although many distances can be used, we obtained the best results with a distance weighted by the interlaced covariance matrix (also called Mahalanobis distance):

$$d_\Gamma^2(\mathbf{x_1}, \mathbf{x_2}) = (\mathbf{x_1} - \mathbf{x_2})^T \Gamma^{-1} (\mathbf{x_1} - \mathbf{x_2}) \tag{8}$$

where Γ is the interclass covariance matrix defined by

$$\Gamma = \frac{\sum_{i=1}^{B} (\mathbf{x_i} - \overline{\mathbf{x}})^T (\mathbf{x_i} - \overline{\mathbf{x}})}{B - 1} \tag{9}$$

with $\overline{\mathbf{x}} = \frac{\sum_{i=1}^{B} \mathbf{x_i}}{B}$

2.2.3.1 Decision

Once the k nearest neighbors of $\mathbf{x}$ have been selected according to Equation (8), the class of $\mathbf{x}$ is determined by the majority vote, that means the most represented class among the k selected. Let n_i be the number of objects with class C_i among the k ones. Then, the class of $\mathbf{x}$ is:

$$e_{kNN}(\mathbf{x}) = \mathrm{Arg}\{\max_{i \in \Lambda} [n_i]\} \tag{10}$$

Figure 4 shows an example of classifying an unknown object $\mathbf{x}$ with k=4, two features and a set of four classes. Thus, among the four nearest neighbors, three belong to class 2 and one belongs to class 3. Class 2 has majority among the four classes and therefore $\mathbf{x}$ is assigned class 2. Furthermore, a probability $P(C_i)$ can be related to each class :

$$P(C_i) = \frac{n_i}{k} \tag{11}$$

2.2.4 Neural Network Classifier (Multi-Layer Perceptron)

A neural network classifier is an algorithm that has attribute vectors as inputs and that processes them through neurons up to a predefined number of outputs. The network used is a Multi-Layer Perceptron (MLP). In the learning phase, feature vectors of which the membership classes are known are placed at the input of the network. The training process enables internal weights $w_i, i = 1, \ldots, n$ to be adjusted. An error retro-propagation method is used to train the network, where, for each training vector $\mathbf{x}$, the sum ER_S (12) of the mean square errors er^2 (13) related to the output neurons is minimized. The network internal weights are then adjusted to minimize ER_S.

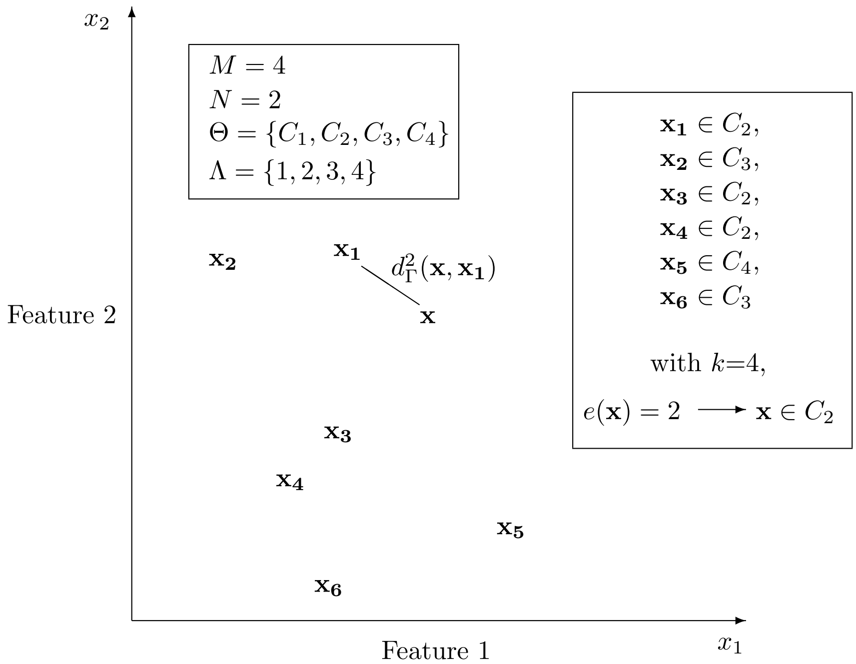


Figure 4: *k Nearest Neighbors classification example*

$$ER_S(\mathbf{x}) = \frac{1}{2} \sum_{i \in \Lambda} er_i^2(\mathbf{x}) \tag{12}$$

$$er_i(\mathbf{x}) = ys_i(\mathbf{x}) - y_i(\mathbf{x}), i \in \Lambda \tag{13}$$

where ys_i corresponds to the desired output value for output neuron i and y_i corresponds to the observed output for output neuron i of the multi-layer perceptron.

A multi-layer perceptron model is presented in Figure 5. Figure 6 illustrates a three-layer perceptron example, with three neurons on the first layer, two on the second and two on the final one. The output of a neuron is represented by y and determined following the threshold function φ. The threshold function used here is the logistic function, where $y \in [0, 1]$:

$$y = \varphi(v) = \frac{1}{1 + e^{-v}} \tag{14}$$

where $v = \sum_{i=1}^{n} w_i x_i - \zeta$, with ζ a threshold.

In the recognition step, feature vectors of unknown objects are placed at the input. With the previously computed weights, the neural network is then able to assign correspondence measures $O_1, O_2, ..., O_M$. The outputs $O_1, O_2, ..., O_M$ are normalized so that their sum is 1. The probability related to each class is then :

$$P(C_i) = O_i \tag{15}$$

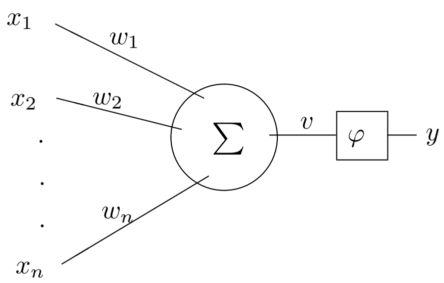


Figure 5: *Multi-Layer Perceptron model*

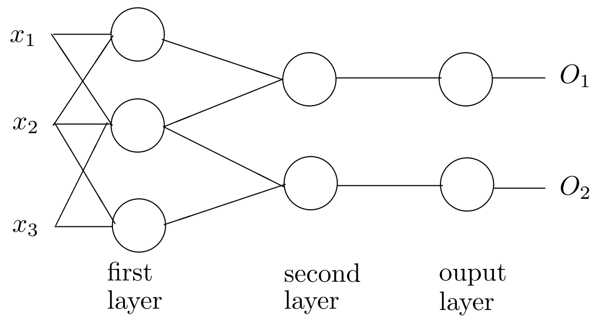


Figure 6: *Example of a neural network*

2.2.4.1 Decision

The selected class is then:

$$e_{NN}(\mathbf{x}) = \text{Arg}\{\max_{i \in \Lambda}[O_i]\} \tag{16}$$

Suppose the example of classifying an unknown object $\mathbf{x}$ by using a neural network, having two features and $\Lambda = [1, 2, 3, 4]$. The neural network will output measures O_1, O_2, O_3 and O_4. If we have O_1=0.2 , O_2=0.45 , O_3=0.2 and O_4=0.15, then class 2 would be assigned to $\mathbf{x}$.

2.3 Performance Ratings

2.3.1 Useful Rates

Four different rates allow measuring the performance of a classifier: the recognition rate (ϵ_r), the substitution rate (ϵ_s), the rejection rate (ϵ_{rej}) and the reliability rate (ϵ_f). ϵ_f and ϵ_{rej} are defined in terms of ϵ_r and ϵ_s :

$$\epsilon_f = \frac{\epsilon_r}{\epsilon_r + \epsilon_s} \tag{17}$$

$$\epsilon_{rej} = 1 - \epsilon_r - \epsilon_s \tag{18}$$

2.3.1.1 Recognition Rate

The recognition rate (ϵ_r) is the ratio of the number of successfully classified objects over the total number of classified objects.

2.3.1.2 Substitution Rate

The substitution rate (ϵ_s) is the ratio of the number of misclassified objects over the total number of classified objects.

2.3.1.3 Rejection Rate

The rejection rate (ϵ_{rej}) is the ratio of the number of rejected objects over the total number of classified objects.

2.3.1.4 Reliability Rate

The reliability rate (ϵ_f) is the ratio of the number of successfully classified objects over the total number of non-rejected classified objects.

2.3.2 Confusion Matrix

The confusion matrix illustrates how a classifier behaves according to the class of an object (Table 1). Tests must be performed to obtain a confusion matrix, where for each training vectors the class returned by the classifier is recorded in relation with the membership class of the training vector. Each line i in the matrix corresponds to the class i of objects, and each column j corresponds to a class j returned by a classifier. The element n_{ij} is the number of objects belonging to C_i and having been classified in C_j by the classifier. Objects classified successfully are represented on the diagonal of the matrix.

Table 1: *Confusion matrix*

Input \ Output	C_1	C_2	...	C_j	...	C_M
C_1	n_{11}	n_{12}	...	n_{1j}	...	n_{1M}
C_2	n_{21}	n_{22}	...	n_{2j}	...	n_{2M}
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
C_i	n_{i1}	n_{i2}	...	n_{ij}	...	n_{iM}
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
C_M	n_{M1}	n_{M2}	...	n_{Mj}	...	n_{MM}

According to that, a perfect classifier would have a confusion matrix like the one shown in Table2.

Table 2: *Ideal confusion matrix*

Intput \ Output	C_1	C_2	...	C_M
C_1	1	0	...	0
C_2	0	1	...	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$
C_M	0	0	...	1

2.3.3 Credibility Rate

Credibility rates are computed from the confusion matrix [6]. They indicate to some extent the uncertainty related to each class :

$$Cr(i) = P(X \in C_i | e(X) = j) = \frac{n_{ij}}{\sum_{i=1}^{M} n_{ij}}, i = 1, ..., M. \tag{19}$$

2.4 Error and Uncertainty Sources

In practice, there is no classifier being able to classify any kind of data perfectly in any situation. Real applications in which classifiers like those of Subsections 2.2.2, 2.2.3 and 2.2.4 have perfect (100 %) or almost perfect recognition rate are very rare. Indeed, some uncertainty is related to each classifier. In a given application, the uncertainty related to a classifier can be estimated globally using the recognition rate ϵ_r and the substitution rate ϵ_s. These rates are obtained by testing the classifiers on known objects. The lowest ϵ_r is and the highest ϵ_s is, the more the results of a classifier are uncertain, and therefore the more probable errors are.

The main error sources of a classifier are class ambiguity, imperfect modeling of the decision limits between classes and poor sampling that does not represent well the problem, maybe because of a wrong choice of features or simply because the number of samples is to small [19].

Class ambiguity becomes obvious when, for example, two samples belonging to different classes have equal feature values. Besides errors or inaccuracies when collecting data, the ambiguity problem can be related to an inappropriate choice of features. Moreover, the modeling of the decision limits between classes is particular to each classifier. Modeling capability of each may vary from an application to another, but generally it will be higher when the representation of the features in the space $\mathcal{C}_N$ forms disjoints sets for each classes. Furthermore, a small set of samples will make the modeling process easier but the resulting representation will be less accurate and divergent from the truth.

There is uncertainty caused by the imperfect object representation in features, in addition to imprecision in feature recording as well as the uncertainty coming from the inability of the classifiers to model perfectly the decision limits between classes. Note that the uncertainty is different for each classification method, implying that different classifiers do not necessary get errors on the same data set for a given problem.

2.5 Conclusion

The simple classifiers to be combined, which are the Bayesian, the k nearest neighbors and the neural network ones, all share the same input and output formats. They differ on the reasoning methods and on the related knowledge. There is uncertainty related to the input features, which is the same for every individual classifier. However, because the individual classifiers have different reasoning methods and different associated *a priori* knowledge, the uncertainty and the imprecision related to the output of each classifier may be different. This explains why the answers of each classifier may be different. Considering that the sources of uncertainty may vary and may depend on the classification algorithm, combining individual classifier into a multiple classifier can be considered as a promising solution to have better performance.

3 Multiple Classifier

3.1 Introduction

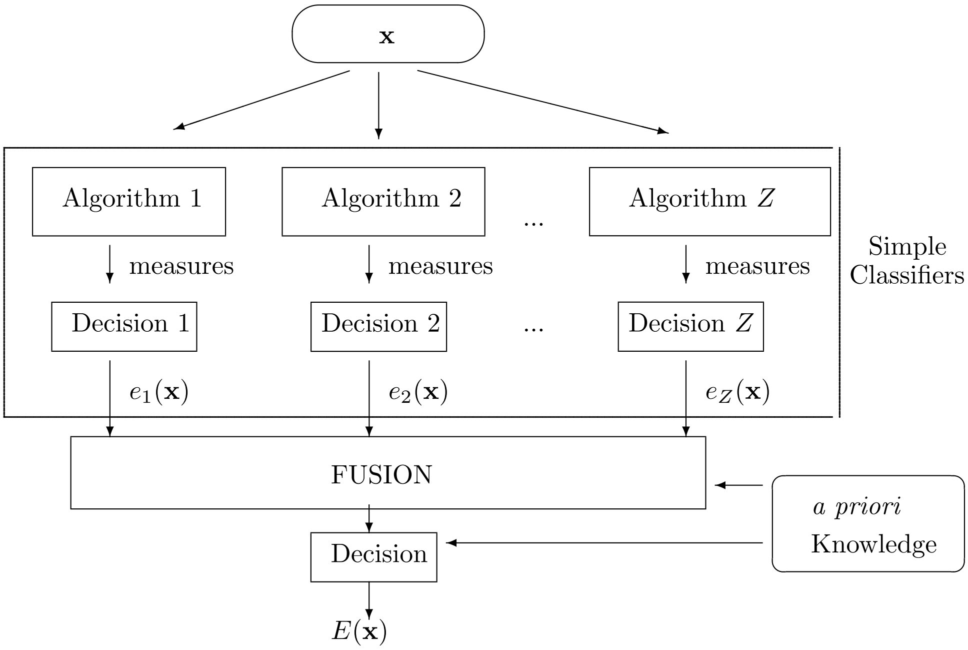


Figure 7: *Structure of a classical multiple classifier*

A multiple classifier is made by the combination of two or more individual classifiers. It fuses the information of the individual classifiers and takes a decision on the new combined information. Figure 7 shows the structure of a classical multiple classifier that fuses the final answers of individual classifiers. The fusion and the decision are most of the time supported with *a priori* knowledge.

A multiple classifier is a technique to improve the performance of individual classifiers. It has two major benefits:

1. Consideration of uncertainty at the output of the individual classifiers and reduction of it by combining their information.

2. Unification of the strengths of each individual classifiers.

Still, it is necessary to know what information of the individual classifiers is relevant and what information is not. In this context, the problem lies in choosing the information to combine, finding a way of representing it and pondering it, determining a combination algorithm and finally establishing an appropriate decision rule. For some reasons related to the consideration of uncertainty (see Section 2.4), the most important step of all is likely to be the representation of the information, that includes the choice of the information to combine and its pondering. That issue will be developed in more details in this work.

Many multiple classifiers have already been proposed. In Subsection 3.3.1, the ‘Behavior Knowledge Space’ [8] is discussed. Subsections 3.3.2 and 3.3.3 present two multiple classifiers that use Dempster-Shafer’s theory, that are Lu’s classifier [7] and Xu’s classifier [6]. Finally, Section 3.3.4 presents a new innovation in this work, where one of the two models proposed by Appriou [16] is applied to the construction of a multiple classifier.

3.2 Using Evidential Theory

The evidential theory presents two major advantages in the design of a multiple classifier:

- It is a theory of the uncertain reasoning and by that takes into account our aptitude to take a good decision at a given time.
- A combination of two distinct believes is well defined and various methods can be implemented to represent the evidence provided by the individual classifiers.

So, it appears as a good tool for multiple classifiers.

3.2.1 Review of Dempster-Shafer’s Theory

Here is a brief introduction to the evidential theory developed by Arthur Dempster [20], then by Glenn Shafer [21].

Let Θ be the set of all the possible outputs of an experiment. Θ is called the frame of discernment. For our example of classification, Θ contains the M (or $M+1$) possible classes to be identified. The power set of Θ, noted 2^Θ, is the set of the 2^M subsets of Θ. A Basic Probability Assignment (or BPA) is a function m defined from 2^Θ to $[0,1]$, respecting the following two conditions:

$$m(\emptyset) = 0 \quad \text{and} \quad \sum_{A \subseteq \Theta} m(A) = 1 \tag{20}$$

where $\emptyset$ is the empty set. $m(A)$ represents the belief that the searched element is in A.

3.2.1.1 Combination

Let m_1 and m_2 be two BPAs representing the believes coming from two different sources. The resulting belief is then given by the Dempster’s rule of combination [20] :

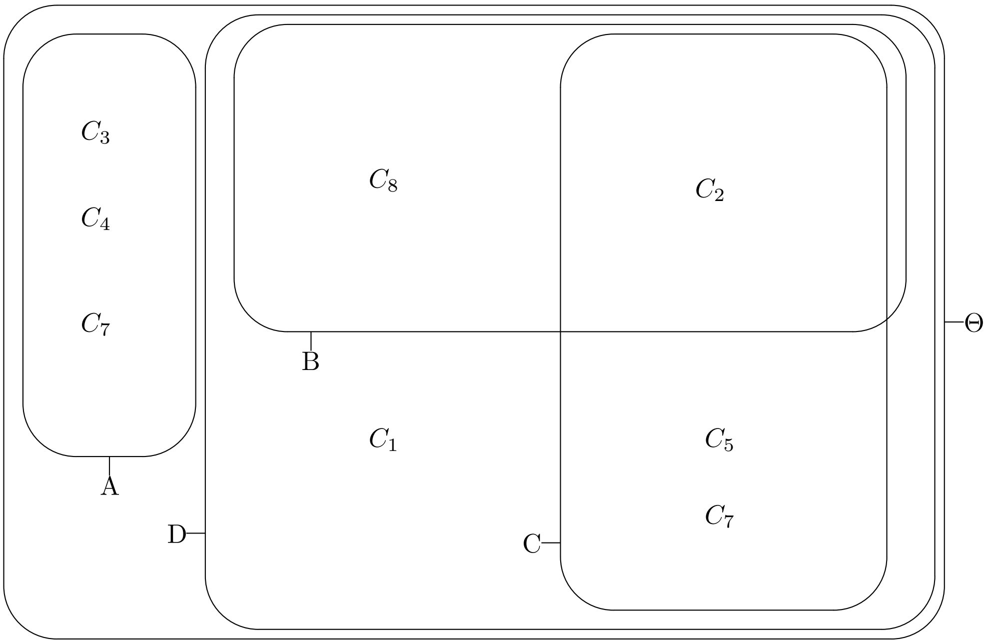
$$m(A) = (m_1 \oplus m_2)(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - K} \tag{21}$$

where K is the conflict factor, $K \in [0,1]$:

$$K = \sum_{B \cap C = \emptyset} m_1(B) m_2(C) \tag{22}$$

The higher K is, the more the combined BPAs are in contradiction, thus K measures the conflict between both evidences represented in BPAs m_1 and m_2. When $K = 0$, m_1 and m_2 are the same, while when $K = 1$ there is total conflict and the combination of m_1 and m_2 results in an empty BPA m.

Figures 8 and 9 illustrate a combination example between two BPAs m_1 and m_2. m_1 and m_2 are defined in Figure 8 and the masses associated to their propositions are shown in the grid of Figure 9. On that grid, the intersection is made between all propositions of each BPA. The resulting mass of the intersection between two propositions is equal to the multiplication of the masses of the propositions. For example, the grid in Figure 9 shows $m_1(B) = 0.5$ and $m_2(D) = 0.4$, where $B \cap D = B$ and with a mass equal to $0.5 \times 0.4 = 0.2$. $m(B)$ value is got by first adding the masses of all the intersections that resulted in set B, that is $0.5 \times 0.4 + 0.5 \times 0.4 + 0.5 \times 0.2 + 0.4 \times 0.1 = 0.54$, and then dividing by $1 - K$. On the grid, K is the sum of the masses of the empty sets, so that $K = 0.2 \times 0.4 + 0.2 \times 0.4 = 0.16$. Finally, $m(B) = 0.54/(1 - 0.16) \approx 0.64$.



$$m_1 : m_1(A) + m_1(B) + m_1(C) + m_1(\Theta) = 1$$

$$m_2 : m_2(B) + m_2(D) + m_2(\Theta) = 1$$

Figure 8: *Example of two BPAs, m_1 and m_2, to be combined*

3.2.1.2 Decision Rule

Some functions enable to describe in similar ways the evidence for a particular proposition, such as the belief function Bel and the plausibility function Pl, given respectively by :

$$\mathrm{Bel}(A) = \sum_{X \subseteq A} m(X) \tag{23}$$

		m_1	0,2 A	0,5 B	0,2 C	0,1 Θ
	0,4	B	ø	B	C_2	B
m_2	0,4	D	ø	B	C	D
	0,2	Θ	A	B	C	Θ

Figure 9: *Combination grid for two BPAs m_1 and m_2*

$$\mathrm{Pl}(A) = \sum_{X \cap A \neq \emptyset} m(X) \tag{24}$$

Note that (24) can also be written $\mathrm{Pl}(A) = 1 - \mathrm{Bel}(\bar{A})$. $\mathrm{Bel}(A)$ measures the total belief according to the fact that the searched element be included in A and $\mathrm{Pl}(A)$ measures the belief that would be allowed to A. Another function that can be used as a decision rule is the pignistic probability function [22]. It is a probability distribution and it takes into account the cardinal of the sets :

$$\mathrm{p(A)} = \sum_{A \in X} \frac{m(X)}{|X|} \tag{25}$$

Thus, a multiple classifier can be created by combining the results of individual classifiers, represented in BPAs, to then get a new BPA on which a decision can be made. That way, uncertainty is considered, conflict between classifiers can be managed, and the performance rates should be improved with a decision that is based on information coming from several classifiers.

3.3 Some Multiple Classifiers

3.3.1 Behavior-Knowledge Space Classifier

The Behavior-Knowledge Space (BKS) method was developed to enhance performance in the recognition of handwritten numerals [8], though it can be applied to other areas of supervised classification. According to Huang and Suen [8], this multiple classifier performs best when there is a high dependency between the answers of the different individual classifiers.

In order to combine K individual classifiers, a K-dimensional space is used, called a Behavior-Knowledge Space (BKS). Each dimension in the space corresponds to the decision of one classifier. A training phase must take place, where for each training sample the decisions returned by the individual classifiers and the membership class are recorded

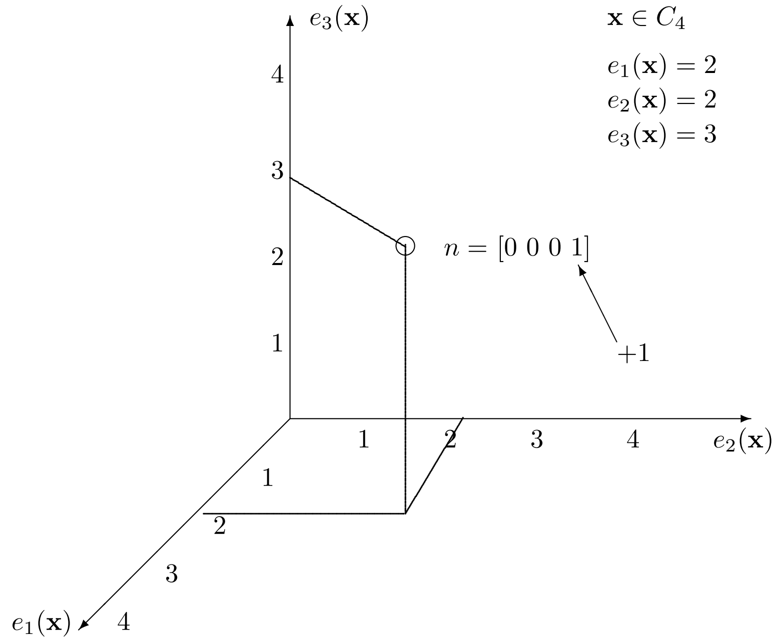


Figure 10: *Training example with Behavior-Knowledge Space multiple classifier*

in the BKS. A unit of the BKS is represented by the decisions of individuals classifiers. At the end of the training, each unit of the BKS contains the best representative class $R_{e(1)\ldots e(K)}$. The best representative class m is the one that has the highest number of samples $n_{e(1)\ldots e(K)}(m)$ among the $T_{e(1)\ldots e(K)}$ samples cumulated in the unit. The decision rule either approve or reject $R_{e(1)\ldots e(K)}$ following some rules described in (26), where λ is a threshold between 0 and 1 (a method to evaluate λ is described in [8]).

$$E_{BKS}(\mathbf{x}) = \begin{cases} R_{e(1)\ldots e(K)}, & \text{when } T_{e(1)\ldots e(K)} > 0 \text{ and } \frac{n_{e(1)\ldots e(K)}(R_{e(1)\ldots e(K)})}{T_{e(1)\ldots e(K)}} \geq \lambda \\ M+1, & \text{otherwise} \end{cases} \tag{26}$$

where $M+1$ represents ignorance.

Figure 10 shows a training process example, with three simple classifiers e_1, e_2 and e_3 and where $\Lambda = \{1, 2, 3, 4\}$. Having a training vector $\mathbf{x}$ belonging to class 4, for which the classifiers return the answers $e_1(\mathbf{x}) = 2$, $e_2(\mathbf{x}) = 2$ and $e_3(\mathbf{x}) = 3$, then $n_{223}(4)$ will be incremented by 1 (Figure 10).

Figure 11 presents a test situation with a vector $\mathbf{x}$, where $e_1(\mathbf{x}) = 2$, $e_2(\mathbf{x}) = 3$ and $e_3(\mathbf{x}) = 4$. In the associated unit of BKS ($BKS(2, 3, 4)$), n is [3 20 76 14]. The best representative class is class 3 because it has the highest number of samples accumulated at unit $BKS(2, 3, 4)$. Thus, $E_{BKS}(\mathbf{x}) = 3$.

The Behavior-Knowledge Space multiple classifier uses only the final decisions of simple classifiers as the information to combine. Also it does not require the independence assumption

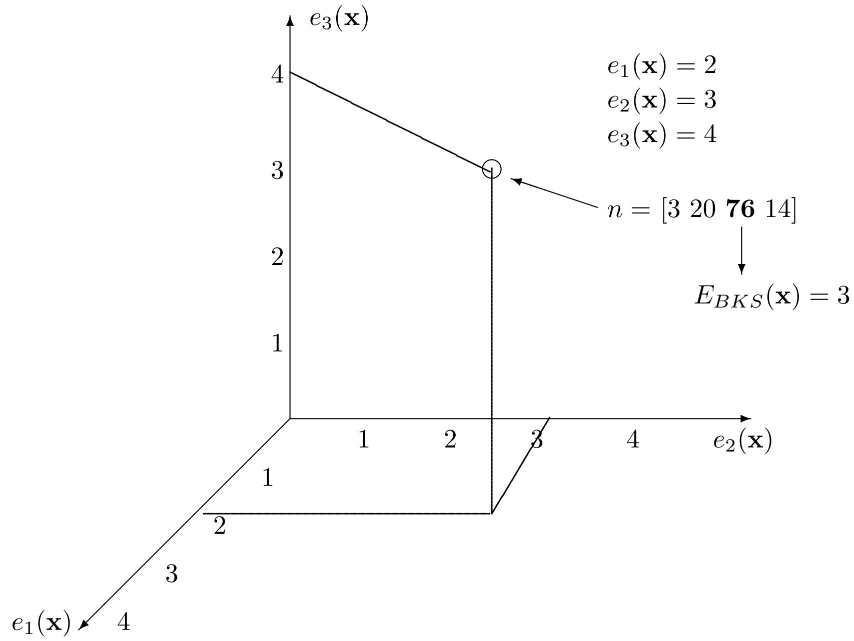


Figure 11: *Test example with Behavior-Knowledge Space multiple classifier*

between individual classifiers output. One of its strength is that it concurrently records the decisions of all classifiers, but it does not take any other information into account. All of its reasoning is based only on the final answers of the individual classifiers.

3.3.2 Lu's Classifier

Before the step of identifying an object, that is to say before the decision step, a classifier has a list with confidence values related to each class. The list is made prior to some algorithms specific to the classifier. Lu's classifier structure is shown in Figure 12. That method receives as only inputs the list of confidence values related to each class [7] and creates basic probability assignments that are combined with Dempster's rule (21). The mass assignment method resolves the problem where confidence values from all classifiers do not have the same range and it also resolves the problem where the same confidence values are equivalent from one classifier to another. Thus, instead of giving a mass to a class using the confidence value reported, a mass is rather given using a function of the confidence value.

With an individual classifier z, the mass related to a class $i \in \Lambda$ with a confidence value $Cf_i \in [0, 1]$ is equal to the probability $P_i^z(Cf_i^z)$ that an object belonging to that class be correctly classified when a confidence value Cf_i^z as been given to the class. This implies a training step where the reliability rate $R_i^z(Cf_i^z)$ is calculated for each class $i \in \Lambda$ and for each of the possible confidence values Cf_i^z for the class. $R_i^z(Cf_i^z)$ expresses the probability that an object $\mathbf{x}$ belonging to C_i be recognized as a member of C_i by classifier z, i.e. $P(e_z(\mathbf{x}) = C_i | \mathbf{x} \in C_i)$ when the confidence value Cf_i^z has been associated to answer j_z. $R_i^z(Cf_i^z)$ can be seen as a density probability function for Cf. In [7], Lu does not specify how to get $R_i^z(Cf_i^z)$ in practice, but it is implied that a frequency histogram for Cf is used. In Subsection 5.3.2, we propose an histogram interpolation method to create a continuous $R_i^z(Cf_i^z)$ function. $P_i^z(Cf_i^z)$ is the ratio between $R_i^z(Cf_i^z)$ and the number N_i^z of objects tested with a non-zero confidence value related to C_i :

$$P_i^z(Cf_i^z) = \frac{R_i^z(Cf_i^z)}{N_i^z} \tag{27}$$

In practice, this ratio is calculated in separated intervals for Cf_i^z. Moreover, a threshold $\lambda_{UC_i^z}$ associated to Cf_i^z is defined. $\lambda_{UC_i^z}$ is determined is such a way that for all $Cf_i^z \geq \lambda_{UC_i^z}$, $R_i^z(Cf_i^z)$ can be approximated by a monotonic curve. It plays a role in the creation of the BPAs. Thus, with classifier z, we have :

$$\begin{aligned} m_z(i) &= \begin{cases} \frac{P_i^z(Cf_i^z)}{\alpha^z} & \text{if } \alpha^z \geq 1, Cf_i^z \geq \lambda_{UC_i^z} \\ P_i^z(Cf_i^z) & \text{if } \alpha^z < 1, Cf_i^z \geq \lambda_{UC_i^z} \\ 0 & \text{if } Cf_i^z < \lambda_{UC_i^z} \end{cases} , \forall i \in \Lambda \\ m_z(\Lambda) &= 1 - \textstyle\sum_{i \in \Lambda} m_z(i) \end{aligned} \tag{28}$$

where $\alpha^z = \sum_{i=0}^{M} P_i^z(Cf_i^z)$.

The combination of Z classifiers is made according to Dempster's rule (21) :

$$m = \bigoplus_{z=1}^{Z} m_z \tag{29}$$

The final decision is defined by the maximum belief :

$$E_{LU}(\mathbf{x}) = \text{Arg}\{\max_{i \in \Lambda} \text{Bel}(i)\} \tag{30}$$

However, if more than one class is returned in (30), than the class with the highest plausibility (Pl) is chosen :

$$A = \{j \in \Lambda | j = \text{Arg}\{\max_{i \in \Lambda} \text{Bel}(i)\}\} \tag{31}$$

$$E_{LU}(\mathbf{x}) = \text{Arg}\{\max_{i \in A} \text{Pl}(i)\} \tag{32}$$

However, this step seems to be useless by the fact that the belief function with a singleton equals its plausibility. Lu's rule states that if the equality still persist (that should always be the case with singletons), then the answer is the set of classes that share the maximum values regarding Bel and Pl.

3.3.3 Xu's Classifier

For the sake of creating a multiple classifier using Dempster-Shafer's theory, Xu proposes a simple method of creating the basic probability assignments (BPAs) [6]. Answer j_z

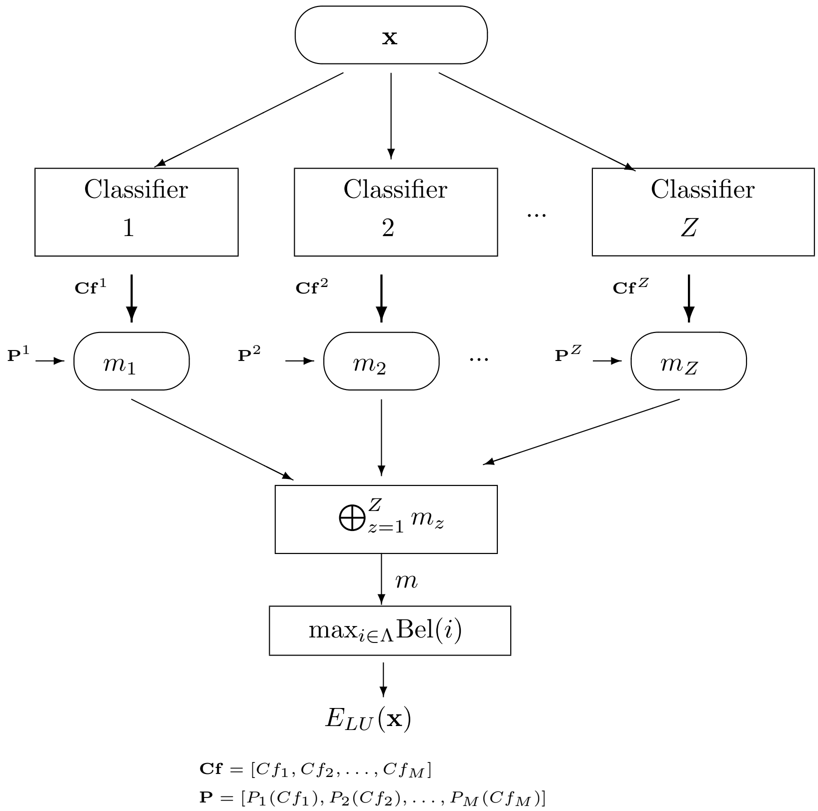


Figure 12: *Lu's classifier structure*

of classifier e_z generates a proposition formed by singleton j_z, as well as a complement proposition $\neg j_z = \Lambda - \{j_z\}$. The mass for j_z is equal to the recognition rate $\epsilon_r^{(z)}$ for classifier e_z and the mass of its complement is equal to the substitution rate $\epsilon_s^{(z)}$. Also, ignorance has a mass related to the rejection rate. Thus, the BPA with classifier z is :

$$\begin{aligned} m_z(j_z) &= \epsilon_r^{(z)} \\ m_z(\neg j_z) &= \epsilon_s^{(z)} \\ m_z(\Lambda) &= 1 - \epsilon_r^{(z)} - \epsilon_s^{(z)} \end{aligned} \tag{33}$$

The combination of N_C classifiers is made according to (29) and it produces a final BPA m over which a decision rule is applied. The decision rule returns as an answer a class C_i for which the difference between Bel(i) and Bel($\neg i$) is maximum. Ignorance is returned $(M+1)$ and the decision is rejected if the maximum difference does not exceed a threshold λ or if more than one class is returned by the decision rule. λ is set arbitrarily.

$$E_{XU}(\mathbf{x}) = \begin{cases} j, & \text{if } [\mathrm{Bel}(j) - \mathrm{Bel}(\neg j)] = \max_{i \in \Lambda} [\mathrm{Bel}(i) - \mathrm{Bel}(\neg i)] > \lambda \\ M+1, & \text{otherwise} \end{cases} \tag{34}$$

So, Xu's method uses the final answers out of the classifiers, and uncertainty is represented by the recognition rate and the substitution rate (Figure 13).

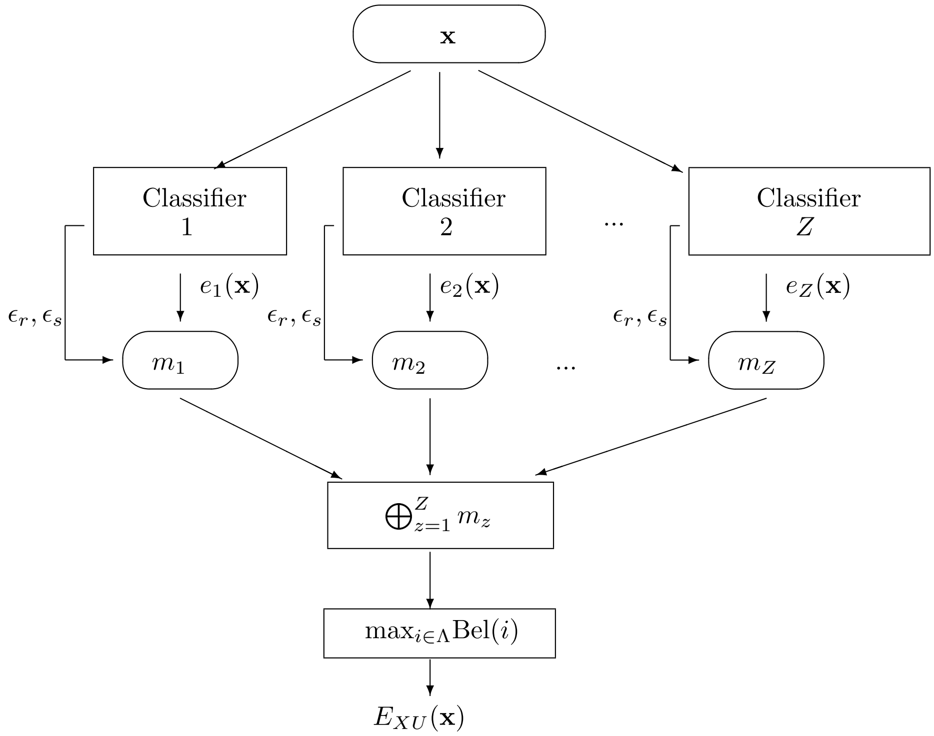


Figure 13: *Xu's classifier structure*

3.3.4 Appriou's Model of Initial Mass Assignment

In [16], Alain Appriou proposes two models of mass functions that represent sensors outputs. He shows that these two models respect three conditions that are described in [16]. Appriou's models are based on a statistic learning of prerequisite measures. Appriou pretends that his method of creating mass functions, with its two models, brings robustness against errors of the statistical model or against imprecision on measures. We have decided to apply one of its models to the output of simple classifiers to form a new multiple classifier. To have a compatible notation with the one of mass functions, *a priori* probability distributions noted $P_i^z(Cf_i^z)$ with Lu's notation are noted $p(mc_z|C_i)$ with Appriou, where mc_z corresponds to Cf_i with Lu, that is the measure related to class i and reported by classifier z.

According to Appriou's *model 1*, the initial BPAs are functions of $p(mc_z|C_i)$:

$$\begin{array}{lcll} A_i & = & i & i \in \Lambda \\ m_{iz}(A_i) & = & \frac{d_{iz}\times R_z \times p(mc_z|C_i)}{1+R_z\times p(mc_z|C_i)} & \forall i \in \Lambda \\ m_{iz}(\neg A_i) & = & \frac{d_{iz}}{1+R_z\times p(mc_z|C_i)} & \forall i \in \Lambda \\ m_{iz}(\Theta) & = & 1 - d_{iz} & \forall i \in \Lambda \end{array} \tag{35}$$

$$m_z(.) = \oplus_{i\in\Lambda} m_{iz}(.) \tag{36}$$

where d_{ij} is a confidence factor on the *a priori* probabilities and R_j is a normalization factor.

An initial BPA is then created for each class $i \in \Lambda$ and for each of the N_C classifiers. An initial BPA m_z of a classifier z is then obtained in combining each of its M BPAs m_{iz}. The BPAs of the N_C classifiers are combined using (21). The decision step follows the maximum plausibility law:

$$E_{AP}(\mathbf{x}) = \mathrm{Arg}\{\max_{i \in \Lambda} \mathrm{Pl}(i)\} \tag{37}$$

3.4 Conclusion

Four different multiple classifiers have been examined. They differ one from the other by at least one of the four facets of a multiple classifier:

1. choices of information and *a priori* knowledge related to the individual classifiers;
2. representation of the information;
3. combination;
4. decision.

The information to combine in BKS and Xu's classifiers comes from the final answers of the individual classifiers, while Appriou and Lu's methods rely on confidence values. BKS *a priori* knowledge lies in a multi-dimensional space, whereas Xu's classifier *a priori* knowledge lies in performance rates. Appriou and Lu's methods use density probability distribution as their *a priori* knowledge. BKS classifier distinguishes itself from the others by its own combination method that is related to the BKS space. The three others use Dempster-Shafer's theory for the combination and the decision steps.

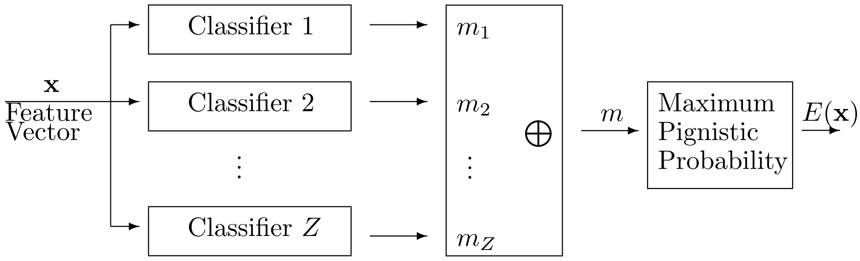


Figure 14: *General structure of the proposed multiple classifiers*

4 Proposed Combination Methods

In the last section, the problem of choosing and representing information in classifiers has been analyzed. Following that, two multiple classifiers are proposed. One uses the abstract level in simple classifiers (Abstract Level Based multiple classifier or ALB multiple classifier) and the other uses the measure level (Measure Level Based multiple classifier or MLB multiple classifier). These two multiple classifiers share the same general structure. Because of the benefits presented in Section 3.2, we propose the general structure of Figure 14, with Dempster's combination rule and a decision rule based on the pignistic probability. Also, the Measure Level Based multiple classifier has the particularity that the methodology to create the basic probability assignments is specific for each single classifier. Some choices must then be made arbitrarily, like the choice of the simple classifiers. In this work the bayesian classifier, the k-NN classifier and the neural network are used. These are presented in Subsections 2.2.2, 2.2.3 and 2.2.4, respectively.

Section 4.1 first presents the available information in simple classifiers (in Subsection 4.1.1), The basic probability assignment creation is described in 4.1.2 for the Abstract Level Based multiple classifier and in 4.1.3 for the Measure Level Based multiple classifier. In Section 4.2, the combination method and the decision rule are discussed. They are the same for both ALB and MLB multiple classifiers. Finally, Section 4.3 summarizes the different choices made in the creation of the multiple classifiers.

4.1 Information Representation using Basic Probability Assignments

Given a specific problem, every classifier has some particular information related to the problem. In order to make a good multiple classifier, it is important to determine what information must be collected on a simple classifier. Also, because the suggested approach has to deal with uncertain reasoning, the evidence under consideration should be associated with some uncertainty in the most representative way as possible.

4.1.1 Measure Level and Abstract Level

As was shown with the three methods of Chapter 2, every classifier can be configured to return a unique answer $e(\mathbf{x}) = j$ at its output. That answer j constitutes the final outcome

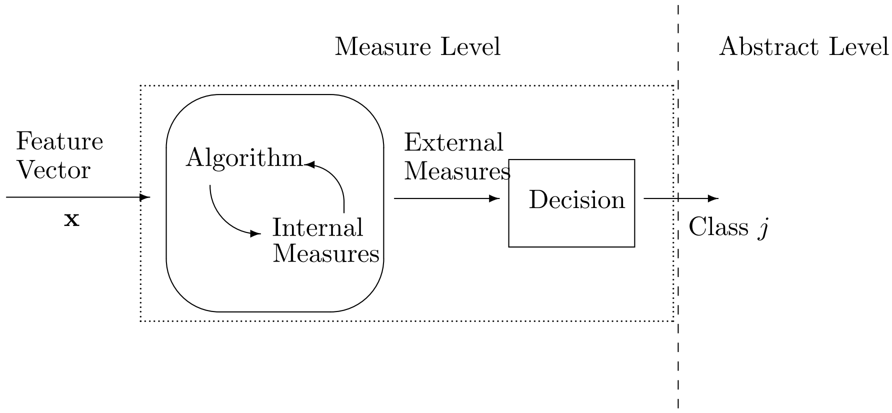


Figure 15: *Simple classifier constitution*

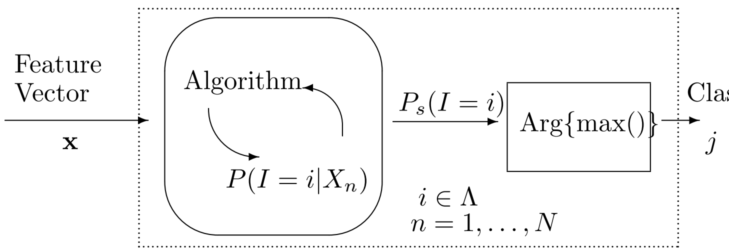


Figure 16: *Within the Bayes classifier*

of the classification process. To get that answer, some reasoning is done on the classifier's internal measures. In fact, all classifiers have internal measures on which decision rules can be applied before returning the final answers. Each calculation step in a classification algorithm usually includes some variables, where some of them can be used for the combination. For example, in [6] the information level in which a confidence value is associated to each class is called the *measurement level*, while the level where the unique answer ($e(\mathbf{x}) = j$) is returned is called the *abstract level*. In this report, the measurement level is rather called *measure level* and is defined differently. It is made of two sublevels : the *internal measure level* and the *external measure level*. The external measure level is equivalent to the measurement level defined in [6], while the internal measure level is the level related to the variables that are within the algorithm of a classifier. The values of the external measure level are denoted *external measures* and the values of the internal measure level are denoted *internal measures*. Figure 15 shows that the external measure level is situated before the decision system and preceded by the internal measure level.

With the Bayesian classifier shown in Figure 16, the external measures are the *a posteriori* probabilities $P_s(I = i)$ (3) and the internal measure level comprises the conditional probabilities $P(I = i|X_n)$, $i \in \Lambda$ and $n = 1, ..., N$. In the neural network of Figure 17, the internal measures are the inter-neurons weights, while the external measures are the $O_i, i \in \Lambda$, on the network final layer.

The k-NN classifier can also be divided into internal and external levels. The external mea-

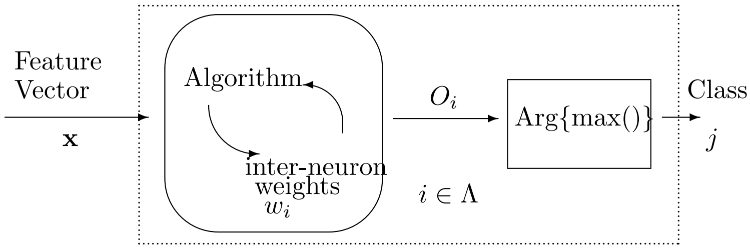


Figure 17: *Within the Neural Network classifier*

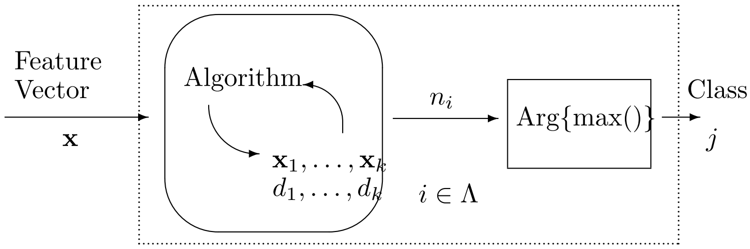


Figure 18: *Within the k-NN classifier*

sures used in the decision of (10) are the n_i values, that are the number of objects belonging to class i among the k nearest neighbors. The n_i values are defined in Subsection 2.2.3. These measures depend on the k objects in which the membership class is known as well as on the distances of the k objects from the unknown object. As shown on Figure 18, the k nearest neighbor distances constitute the internal measure level.

Also, Figures 16, 17 and 18 show that the unique answers j of the classifiers constitute an information reduction compared to the measure level. The measure level is followed with the selection of a final answer j at the output of a classifier. In a multiple classifier, the measure level could be useful because it has more information available than the abstract level. Two solutions then come out for determining the evidence for the combination : using measures (measure level) or using answers $e(\mathbf{x}) = j$ (abstract level).

As mentioned earlier, the uncertainty related to the evidence provided by a classifier must be evaluated. At the abstract level, the information returned is categorical and absolute, what is rarely true in classification. Then the answer $e(\mathbf{x}) = j$ must be pondered in a way to represent the uncertainty related to it. The recognition rate ϵ_r and the substitution rate ϵ_s are examples of some knowledge that could ponder the information at the abstract level. Such weighting gives a general representation of uncertainty and does not change with the answer $e(\mathbf{x}) = j$ [6].

Specifically to each class, the uncertainty at the abstract level can be represented by the credibility rates for each class (19). According to the answer $e(\mathbf{x}) = j$, M credibility rates are calculated and can be used to represent the uncertainty on each class.

At the measure level, uncertainty can often be extracted from the measures themselves, so that it is not necessary to get support from other knowledge sources to represent it.

Nevertheless, it is possible to associate *a priori* knowledge with the measures in order to get new more adequate information and to try diminishing the classification data imperfections. Lu's classifier is one example of this. However, although in some situations it may be a good method for multiple classifiers, such a method assigns weights to measures that are already weights, so that there is a possibility of adding more imprecision and errors. In fact, the *a priori* knowledge needed in that method necessitates a training period. Normally, that knowledge is not without imperfections and errors, as discussed in Section 2.4, and so it is uncertain. The risk of having more errors can be increased by the fact that the uncertainty related to the *a priori* knowledge on the measures will add to the uncertainty in the measures themselves. So, we conclude that at the measure level, when the uncertainty related to *a priori* knowledge on the measures is high (higher than the uncertainty in the measures themselves), it should be better to consider the measures alone. On the other hand, at the abstract level the use of *a priori* knowledge is appropriate to quantify the uncertainty.

4.1.2 Abstract Level Based Multiple Classifier (ALB Multiple Classifier)

The basic probability assignments in the ALB multiple classifier are created from the information available at the abstract level of the simple classifiers. Thus, the method we propose uses credibility rates on each class and is defined in (19). It works the same with any of the simple classifiers that are used. For a simple classifier with an answer $e(\mathbf{x}) = j$, the classes in which consecutive variations between credibility rates $Cr(i), i \in \Lambda$, are smaller than a threshold $\lambda_t \in [0, 1]$ form a proposition. The propositions are then formed with the following algorithm :

Algorithm 1

1. Order in a vector V all classes i by decreasing $Cr(i)$ order, where $V = [V(1), V(2), ..., V(M)]$ and where $V(1)$ corresponds to the class with the highest credibility rate.
2. Let a and j be two indexes initialized to $a = 1$ and $j = 1$.
3. Create a new proposition $A = \{\o\}$.
4. Put $V(a)$ in A.
5. Increment a of 1.
6. If $a = M + 1$, go to step 8.
7. If $Cr(V(a-1)) - Cr(V(a)) < \lambda_t$, return to step 4,
   if not, increment j by 1 and return to step 3.
8. End

The associated mass for each proposition A_j is determined by the average credibility rate μ_{A_j} of all classes included in the proposition :

$$m(A_j) = \frac{1}{S}\mu_{A_j}, \tag{38}$$

with

$$\mu_{A_j} = \frac{\sum_{i \in A_j} Cr(i)}{|A_j|}, \tag{39}$$

where S is a normalization constant :

$$S = \sum_{j \in \Lambda} \mu_{A_j}. \tag{40}$$

The number of propositions created is then between 1 and M.

The particular case where $\lambda_t = 0$ is equivalent to defining basic probability assignments in which there are M singletons $A_i, i \in \Lambda$, each containing a class where the associated mass is equal to the credibility rate of the class :

$$\begin{array}{lcll} A_i & = & i, & \forall i \in \Lambda \\ m_z(A_i) & = & Cr_z(i), & \forall i \in \Lambda \end{array} \tag{41}$$

Dempster's combination of the BPAs developed in (41) reduces to the addition of the credibility rates of all classifiers for each class :

$$m(i) = \frac{1}{SC} \sum_{z=1}^{N_C} Cr_z(i), \tag{42}$$

with SC a normalization constant :

$$SC = \sum_{i=1}^{M} Cr_z(i). \tag{43}$$

Example Suppose a classification problem with $M = 6$. Then we have $\Theta = \{C_1, C_2, C_3, C_4, C_5, C_6\}$, and $\Lambda = \{1, 2, 3, 4, 5, 6\}$. Let e_z be a classifier for which a decision has returned the answer $e_z(\mathbf{x}) = 4$, with the following credibility rates :

$$\begin{array}{rl} Cr_z(1) = & 0.02 \\ Cr_z(2) = & 0.22 \\ Cr_z(3) = & 0.03 \\ Cr_z(4) = & 0.48 \\ Cr_z(5) = & 0.20 \\ Cr_z(6) = & 0.05 \end{array}$$

Figure 19 shows how the propositions are made using the credibility rates when $\lambda_t > 0$. For example, with $\lambda_t = 0.025$, the successive variations between 0.02, 0.03 and 0.05 are lower than λ_t, as well as for 0.2 and 0.22. Thus, the following propositions are created :

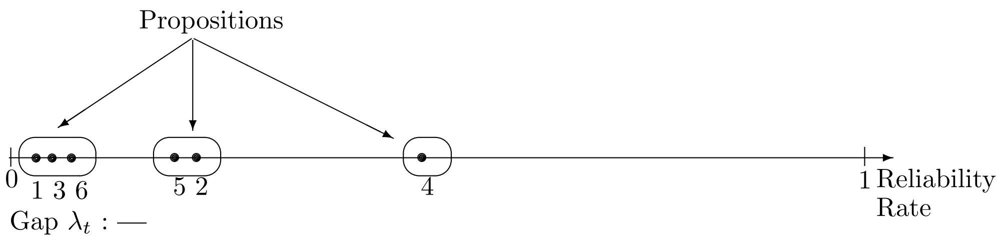


Figure 19: *Creation of propositions with credibility rates and with* $\lambda_t > 0$

$$\begin{aligned} A_1 = &\ \{4\} \\ A_2 = &\ \{2,5\} \\ A_3 = &\ \{1,3,6\} \end{aligned}$$

with, $\mu_{A_1} = Cr(4)/1 = 0.48$, $\mu_{A_2} = (Cr(2) + Cr(5))/2 = 0.21$, $\mu_{A_3} = (Cr(1) + Cr(3) + Cr(6))/3 = 0.1/3$ and $S = 0.48{+}0.21{+}0.1/3$, according to (39) and (40). These values allow for the calculation of masses $m(A_1)$, $m(A_2)$ and $m(A_3)$ using (38). Finally, $m(A_1) = 0.664$, $m(A_2) = 0.29$ and $m(A_3) = 0.046$.

The suggested approach groups the classes with close credibility rates into a same proposition, according to the credibility rate imprecision threshold λ_t. The higher the imprecision is on the credibility rates, the higher λ_t should be. The specific case of $\lambda_t = 0$ means that no imprecision is related to the credibility rates. To determine λ_t conveniently for a given simple classifier, series of tests with different training sets should be performed to calculate the average standard deviation of the credibility rates. If λ_t is fixed arbitrarily, it is better to have a small value rather than a high one to avoid precision loss. Section 5.3 presents some examples of λ_t values in a practical case.

For the three classifiers of Section 4.1, the basic probability assignments are created using their answers j and the associated credibility rates $\mathbf{Cr} = [Cr(1)Cr(2)\ldots Cr(M)]^T$:

$$\begin{array}{llll} \text{Bayes} & \Rightarrow & j_B,\ \mathbf{Cr}_B & \Rightarrow \quad m_B \\ \text{Neural Network} & \Rightarrow & j_{RN},\ \mathbf{Cr}_{RN} & \Rightarrow \quad m_{RN} \\ k\text{-NN} & \Rightarrow & j_{kPPV},\ \mathbf{Cr}_{kPPV} & \Rightarrow \quad m_{kPPV} \end{array}$$

The resulting multiple classifier is called $AbstractLevelBased$ multiple classifier, symbolized by E_{ALB}.

4.1.3 Measure Level Based Multiple Classifier

At the measure level, the basic probability assignments can be created using the classifiers' internal values, at the internal measure level, or using the external measure level. That choice depends on the individual classifiers' nature. Some may have more relevant information at the internal level and other may have more relevant information at the external level. In this work, the BPAs for the Bayesian and the neural network classifiers are created using some information at the external measure level, while the BPA for the k-NN classifier is made using the internal measure level.

4.1.3.1 External Measure Level

The external measure level is composed of confidence values related to each class, that are M values Cf_1 Cf_2 ... Cf_M grouped into a vector $\mathbf{Cf}$. Using this level, the suggested method to create basic probability assignments is similar to the method developed in 4.1.2, except that the information at the abstract level is not considered, so measures **Cf** are used instead of credibility rates Cr.

At the external measure level, the measure format is the same as the M credibility rates used in 4.1.2, and then the BPAs can be made the same way. Although they share the same format, the M credibility rates and the external measures come from different sources. Determining the uncertainty related to the external measures is harder than determining it for the credibility rates. Consequently, because the uncertainty cannot be measured on **Cf** values, λ_t must be fixed arbitrarily.

With the Bayesian classifier, the BPAs are created with the normalized summed probabilities $P_s(I)$ and following **algorithm 1**, becoming :

Algorithm 1.1

1. Order in a vector V all classes i by decreasing order of $P_s(i)$, where $V = [V(1), V(2), ..., V(M)]$ and where $V(1)$ corresponds to the class with the highest credibility rate.
2. Let a and j be two indexes initialized to $a = 1$ and $j = 1$.
3. Create a new proposition $A_j = \{\emptyset\}$.
4. Put $V(a)$ in A_j.
5. Increment a of 1.
6. If $a = M + 1$, go to step 8.
7. If $P_s(V(a-1)) - P_s(V(a)) < \lambda_t$, return to step 4,
   if not, increment j of 1 and return to step 3.
8. End

Masses are set according to the average summed probability $\mu^B_{A_j}$:

$$m_B(A_j) = \frac{1}{S}\mu^B_{A_j}, \tag{44}$$

with

$$\mu^B_{A_j} = \frac{\sum_{i \in A_j} P_s(i)}{|A_j|}, \tag{45}$$

$$S = \sum_j \mu^B_{A_j}, \tag{46}$$

and when $\lambda_t = 0$

$$\begin{array}{rcll} A_i & = & i & i \in \Lambda, \\ m_B(A_i) & = & P_s(i) & \forall i \in \Lambda. \end{array} \tag{47}$$

Measures $P(I|X_n)$ in the internal level could be used to create the BPAs. But even though the information at the external level ($P_s(I)$) is a reduction of the information at the interval level ($P(I|X_n)$), the $P_s(I)$ measures are viewed as a condensed representation of the $P(I|X_n)$ that is relevant and adequate for creating BPAs.

The BPAs of the neural network are made similarly to the ones of the Bayesian classifier :

Algorithm 1.2

1. Order in a vector V all classes i by decreasing order of O_i, where $V = [V(1), V(2), ..., V(M)]$ and where $V(1)$ corresponds to the class with the highest credibility rate.
2. Let a and j be two indexes initialized to $a = 1$ and $j = 1$.
3. Create a new proposition $A_j = \{\emptyset\}$.
4. Put $V(a)$ in A_j.
5. Increment a of 1.
6. If $a = M + 1$, go to step 8.
7. If $O_{(V(a-1))} - O_{(V(a))} < \lambda_t$, return to step 4,
   if not, increment j of 1 and return to step 3.
8. End

The masses are determined according to the average $\mu_{A_j}^{RN}$ of the outputs O and related to a proposition A_j :

$$m_{RN}(A_j) = \frac{1}{S}\mu_{A_j}^{RN}, \tag{48}$$

with

$$\mu_{A_j}^{RN} = \frac{\sum_{i \in A_j} O_i}{|A_j|}, \tag{49}$$

$$S = \sum_j \mu_{A_j}^{RN}, \tag{50}$$

and when $\lambda_t = 0$

$$\begin{array}{rcll} A_i & = & i & i \in \Lambda, \\ m_{RN}(A_i) & = & P_s(I = i) & \forall i \in \Lambda. \end{array} \tag{51}$$

In a neural network, the information provided by the internal measures can be hardly interpreted to create BPAs. That is why the external measures O_i are used instead in this work.

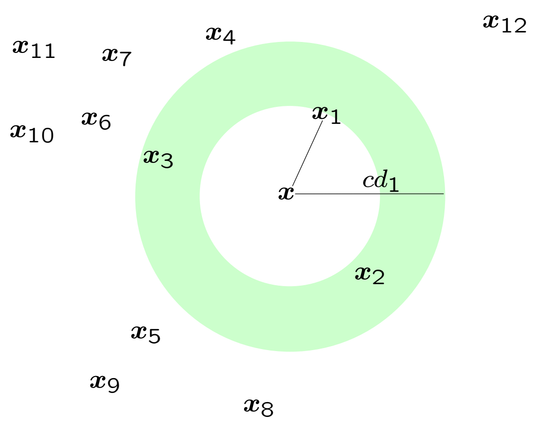


Figure 20: *Generating proposition A_1 with the k-NN classifier*

4.1.3.2 Internal Measure Level

Algorithm 1 could be used with k-NN classifier if the information at the internal measure level was transformed into confidence values $\mathbf{Cf}$. For example, each Cf_i could be equal to n_i/k. n_i is defined in Subsection 2.2.3 as the number of objects belonging to class C_i among the k nearest neighbors. However, such a method does not take into account the significant information that are distances $d_i = d(\mathbf{x}, \mathbf{x_i})$,$i = 1, ..., k$ associated to the k nearest neighbors $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_k}$. These are internal measures. The meaning of these measures is well understood and they can be easily interpreted, supporting their utilization rather than the external measures $\mathbf{Cf}$.

Thus, the k-NN classifier generates two propositions, A_1 and A_2. Proposition A_1 is made by taking the nearest neighbor $\mathbf{x_1}$ and the neighbors for which the distance does not exceed an amount c of the distance d_1, where d_1 is the shortest distance ($d_1 = d(\mathbf{x}, \mathbf{x_1})$). Proposition A_1 is then made of the classes that are represented by all neighbors in which the distance lie in the range $[d_1, cd_1]$ (Figure 20). The membership classes of the neighbors lying in $[d_1, cd_1]$ form the set of proposition A_1 :

$$A_1 = \{\text{Class}(\mathbf{x}) \in \Lambda | d(\mathbf{x}, \mathbf{x_1}) \leq cd(\mathbf{x}, \mathbf{x_1})\}, \tag{52}$$

where Class($\mathbf{x}$) is the class of $\mathbf{x}$, and c a constant such that $c \geq 1$.

The remaining proposition, A_2, is made with the classes of the k nearest neighbors :

$$A_2 = \{\text{Class}(\mathbf{x_i}) \in \Lambda | i = 1, \ldots, k\} \tag{53}$$

and represents the ignorance of the simple classifier. Choosing k will be discussed in Chapter 5.

The mass of A_1 is proportional to the number k_1 of neighbors lying in the ring $[d_1, cd_1]$ and is inversely proportional to the sum of their distances (54). In a similar way, A_2 receives a mass proportional to k and inversely proportional to the sum of the k shortest neighbor distances (55). Then we have :

$$m_{kPPV}(A_1) = \frac{1}{T}\frac{k_1}{\sum_{i=1}^{k_1} d_i}, \tag{54}$$

$$m_{kPPV}(A_2) = \frac{1}{T}\frac{k}{\sum_{i=1}^{k} d_i} \tag{55}$$

where k_1 is the number of neighbors in $[d_1, cd_1]$ and with

$$T = \frac{k}{\sum_{i=1}^{k} d_i} + \frac{k_1}{\sum_{i=1}^{k_1} d_i} \tag{56}$$

a normalization constant.

In short, the BPAs created with the Bayesian classifier uses its M measures $P_s(I = i)$, the BPA created with the neural network uses the M measures O_i and the BPAs made with the k-NN classifier uses distances $d_1, d_2, ..., d_k$ of the k nearest neighbors $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_k}$:

$$\begin{array}{llll} \text{Bayes} & \Rightarrow P_s(I=1), P_s(I=2), \ldots, P_s(I=M) & \Rightarrow & m_B \\ \text{Neural Network} & \Rightarrow O_1, O_2, \ldots, O_M & \Rightarrow & m_{RN} \\ k\text{-NN} & \Rightarrow d_1, d_2, \ldots, d_k & \Rightarrow & m_{kPPV} \end{array}$$

The resulting multiple classifier is called $MeasureLevelBased$ multiple classifier, symbolized by E_{MLB}.

4.2 Combination and Final Decision

Whatever the BPAs are made according to the abstract level (Subsection 4.1.2) or according to the measure level (Subsection 4.1.3), their combination, that is the combination of the BPAs for the Bayes classifier, the BPAs for the neural network and the BPAs for the k-NN classifier, is made following Dempster's rule (21) :

$$m = m_B \oplus m_{RN} \oplus m_{kPPV} \tag{57}$$

Decision The resulting function m is subjected to the maximum pignistic probability decision rule (25). That rule returns an answer $j \in \Lambda$.

$$E(\mathbf{x}) = \begin{cases} \text{Arg}\{\max_{i\in\Lambda} \text{p(i)}\} & \text{if } K < \lambda_{REJ} \\ M+1 & \text{if } K \geq \lambda_{REJ} \end{cases} \tag{58}$$

Rejection Condition The decision is rejected if the conflict factor K exceeds a pre-established threshold λ_{REJ}. K is a result of the combination. In such a situation the multiple classifier returns the answer $M + 1$. As λ_{REJ} lowers, the rejection rate increases. The rejection rate is equal to $1 - \epsilon_r - \epsilon_s$. If λ_{REJ} is 0, all decisions are rejected and then the rejection rates is 100%. Inversely, if λ_{REJ} is 1, all decisions are accepted and the rejection rate is null unless being in the case where more than one class share the maximum pignistic probability. For the results presented in Chapters 5, 6 and 7, all decisions in which the maximum pignistic probability is shared among more than one class are rejected.

4.3 Conclusion

For the purpose of combination, Dempster-Shafer's theory has been chosen among other methods like the Bayesian method, voting methods or fuzzy logic for example. Its flexibility along with the ability to represent different uncertain situations and to perform combination makes it a good choice. Moreover, the maximum pignistic probability rule expresses a coherent judgment on the class to be chosen. Based on that, two ways of creating BPAs have been proposed. One uses the final answers of the simple classifiers with their related credibility rates and the other uses the measures that precede the decision step of simple classifiers. Because the final answers depend on the measures, the multiple classifiers created with the two BPA creation methods will be dependent. For that reason, it is not a good idea to combine the final BPAs of the two proposed multiple classifiers. Doing that would risk reducing the performance compared to the best one of the two classifiers. Tests have been carried out to compare the performance of each multiple classifiers. They are presented in Chapters 5, 6 and 7.

5 Application to Infrared Image Recognition

Automatic target recognition has much to do with classification. Its goal is to identify a target observed by different sensors. Here the classification methods studied in the previous chapters will be applied to the recognition of military or merchant ships. Ships are represented on infrared (IR) images treated beforehand and characterized by 11 parameters [23].

5.1 Database

This project has been achieved with a database called *IR_SHIP data* and originally built by the *United States Naval Air Warfare Center* in China Lake, California. The database has 2545 infrared images characterized by 11 parameters and representing 8 classes of ships. Table 3 presents the name of the eight classes and Table 4 is an example of some of the 2545 ships in the database. The parameters have been extracted by Youngtee Park within the framework of its PhD [24, 23] and under direction of professor Jack Sklansky, University of California, Irvine. There are seven moment invariant functions defined by Hu [3, 25] and four parameters of the auto-regressive model.

Class	Type of Ship
1	Destroyer
2	Container
3	Civilian Freighter
4	Auxiliary Oil Replenishment
5	Landing Assault Tanker
6	Frigate
7	Cruiser
8	Destroyer with Guided Missile

Table 3: *Class numbers and types of ships*

Λ	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
1	833.9	0.6596	0.2435	0.02676	6.056e-05	-0.00658	-0.002159	1.793	-1.22	0.3694	0.5318
4	1683	0.9755	0.00713	0.006135	4.057e-05	0.006059	-4.959e-07	0.653	0.1547	-0.2882	2.261
2	2386	0.8139	0.08303	0.05657	0.00386	0.05066	0.0003688	1.245	-0.5276	0.08514	0.9454
6	1090	0.9258	0.03277	0.005395	2.384e-05	-0.00058	6.766e-05	0.9141	0.07718	-0.1959	0.955
3	2119	0.9662	0.02237	0.02428	0.0005659	0.02369	5.081e-06	0.7598	-0.2499	-0.08692	1.586
2	1498	0.8307	0.1124	0.07573	0.006955	0.06872	0.0006801	0.9378	-0.1369	-0.0765	1.065
5	952.4	0.7736	0.04926	0.02562	0.0009074	0.02111	-6.817e-05	1.182	-0.6471	0.2592	1.401
4	1610	0.9743	0.00958	0.008161	7.214e-05	0.008031	-1.339e-07	0.9664	-0.331	-0.1018	3.105
7	671.6	0.5972	0.2871	0.03733	0.002513	0.008616	-0.002935	1.434	-0.6501	0.134	0.4857
8	1077	0.9167	0.04537	0.008292	8.651e-05	0.000524	-0.0001356	1.019	-0.5566	0.3413	0.6828

Table 4: *IR_SHIP database sample*

Table 5 shows the proportion of each class in the database. Class 4 is the most numerous while class 3 is the rarest.

Class	1	2	3	4	5	6	7	8
Number	340	455	186	490	348	279	239	208
Proportion (%)	13.36	17.88	7.31	19.25	13.67	10.96	9.39	8.17

Table 5: *Proportion of each class in the IR_SHIP database*

From the 2545 images in the database, two files have been created. Each have 1250 images and the proportions of each class are the same for both files. Forty-five images were excluded in order to have identical proportions in both files. One file is used as a training file for the simple classifiers and the other is used as a test file. Normally, a third file should have been used for the multiple classifiers that need a training period, such as BKS classifier, Lu's classifier and Appriou's model method. Also, Xu and ALB classifiers use rates that must be determined according to prior tests. However, making three files out of 2545 images would have resulted in too small sample sets. For example, only 62 ships of class 3 would have been placed in each file, what might be an incomplete representation of class 3 objects. That is why the training phase to obtain prerequisite statistics related to the multiple classifiers has been done using the same file of 1250 ships used in tests. Consequently, the results should be slightly better using only two files rather than three if we suppose that the third one (a second training file) as been created independently of the two others.

5.2 Simple Classifiers

The classifiers studied in Chapter 2, that are the Bayes classifiers, the k-NN classifier and the neural network classifier, are tested individually. As Table 6 shows, the k-NN classifier and the neural network classifier have similar recognition rates of 92.88% and 92.72%, respectively. This is relatively high compared to the 77.12% rate obtained with the Bayesian classifier. The rejection rate (ϵ_{rej}) is zero with all three classifiers. It is normal since they do not have rejection criteria. Table 6 also shows reliability rates (ϵ_f).

Rates interpretation is relative to the application. With some applications, the number of rejections is not a problem as long as reliability is high, but in other applications it is important to have as less rejections as possible in order to obtain the highest recognition rate possible notwithstanding the risk of errors.

Classifier	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
e_B	0.7712	0.2288	0	0.7712
e_{RN}	0.9272	0.0728	0	0.9272
e_{kPPV}	0.9288	0.0712	0	0.9288

Table 6: *Individual classifier results - Infrared image recognition*

Table 7 shows the confusion matrix for Bayes classifier. The matrix has the following parameters : The M classes are equiprobable and thus, $P(I = i) = 1/M$, $i \in \Lambda$; *a priori* distributions are calculated with 40 intervals ($Q = 40$). Note that the whole database is used (2545 objects) to calculate the *a priori* distributions. The decision rule used is the one defined by (4).

Input \ Ouput	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_1	**0.8210**	0.037	0	0	0	0.0309	0.0864	0.0247
C_2	0	**0.7589**	0.0491	0.0848	0.1027	0	0.0045	0
C_3	0.02	0.06	**0.52**	0.28	0.08	0.0100	0.02	0.01
C_4	0.0042	0.0252	0.0378	**0.895**	0.021	0.0084	0	0.0084
C_5	0.0057	0.12	0.0514	0.0457	**0.7371**	0.0114	0.0114	0.0171
C_6	0.1449	0.0145	0	0	0	**0.7029**	0	0.1377
C_7	0.1504	0.0885	0	0	0	0	**0.6991**	0.0619
C_8	0	0.01	0	0	0	0.08	0	**0.91**

Table 7: *Bayes classifier (e_B) confusion matrix - Infrared image recognition*

The neural network is configured with two layers of respectively 50 and 30 neurons. The confusion matrix shown in Table 8 is produced with the following configuration [26, 27]:

- gain: 0.2;
- momentum: 0.5;
- maximum output error: 0.0001;
- epsilon (ϵ): 0.1;
- maximum iterations: 100.

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_1	**0.858**	0	0	0	0	0.080	0.0556	0.0062
C_2	0	**0.9866**	0	0.0134	0	0	0	0
C_3	0	0	**0.84**	0.15	0	0.01	0	0
C_4	0	0	0.0084	**0.9916**	0	0	0	0
C_5	0	0	0.0115	0.0457	**0.9429**	0	0	0
C_6	0.029	0	0	0	0.029	**0.9058**	0.0073	0.029
C_7	0.1062	0	0	0.0089	0	0	**0.885**	0
C_8	0.01	0	0	0	0	0.03	0.02	**0.94**

Table 8: *Neural Network classifier (e_{NN}) confusion matrix - Infrared image recognition*

Table 9 shows the confusion matrix for the k-NN classifier, where the distances are calculated according to (8) and (9). The value of k is set to 3 according to the results presented in [18] for the same IR image database. Note that if a tested image has three nearest neighbors of three different classes, then there is no majority. In that case, the class of the nearest neighbor is chosen. The results for the k-NN classifier are presented in Table 6. Besides, in the case of no majority, if ignorance is returned (rejection) instead of returning the class of the nearest neighbor, then we have a rejection rate of 1.44%, along with respectively 92%, 6.56% and 93.34% for recognition, substitution and reliability rates.

Tables 7, 8 and 9 show that there is a relation between the recognition rate and the proportion of each class in the database (Table 5). Thus, class 3 is the worst recognized by all three simple classifiers. Besides, it seems that there is some noticeable confusion between classes 3 and 4, mainly with the Bayes classifier where 28% of class 3 images are reported to be class 4, compared to 15% with the neural network and to 6% with the k-NN classifier. Moreover, class 7 has a quite low recognition rate where the highest result of the three simple classifier is 83.19% (with k-NN). Class 7 is slightly confused with class 1 : the rates for class 7 images that are classified in class 1 are 15.04%, 10.62% and 6.19%, respectively, for the Bayes classifier, the neural network classifier and the k-NN classifier. Those errors are more likely to be reproduced in a multiple classifier because they are mostly spread in all of the three simple classifiers.

5.3 Existing Multiple Classifiers

BKS (Subsection 3.3.1), Lu's multiple classifier (Subsection 3.3.2), Xu's multiple classifier (Subsection 3.3.3) and Appriou's model multiple classifier (Subsection 3.3.4) are tested on the IR image database, where the Bayes, the k-NN and the neural network classifiers are combined. Even though BKS and Xu's multiple classifier do not have recognition rates very high above those of the k-NN and of the neural network individual classifiers, all multiple classifiers were able to improve reliability over the individual classifier, including the BKS one. Their reliability rates range from 95.36% to 96.63%, as shown in Table 10, whereas simple classifiers rates range from 77.12% to 92.88%.

5.3.1 Behavior Knowledge Space Multiple Classifier

As shown in Table 10, the BKS method [8] has a recognition rate of 94.08%. That was done with $\lambda = 0$. Its low substitution rate (3.28%) along with a 2.64% rejection rate explain why the BKS method has the highest reliability rate (96.63%). Since $\lambda = 0$, rejections are only caused by empty units in the BKS space. For example, if during the test phase for an unknown object $\mathbf{x}$, the Bayes classifier, the k-NN classifier and the neural network classifiers return respectively decisions $e_B(\mathbf{x}) = 3$, $e_{kNN}(\mathbf{x}) = 4$ and $e_{NN}(\mathbf{x}) = 6$, and that the corresponding unit $BKS(3, 4, 6)$ is empty, that is to say that in the training phase no training samples yield the answers $e_B(\mathbf{x}) = 3$, $e_{kNN}(\mathbf{x}) = 4$ and $e_{NN}(\mathbf{x}) = 6$, then $\mathbf{x}$ is rejected. Furthermore, according to the automatic threshold finding method in [8], λ should be 0.43. With λ=0.43, the results stay the same, so there is no difference between a 0 or 0.43 threshold. Note that if there are more than one class returned, we have decided that in this case the answer would be the class returned by k-NN classifier instead of rejecting the decision. However, it does not happen in that application but it may happen in others.

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_1	**0.9074**	0	0	0	0	0.0432	0.0494	0
C_2	0	**1**	0	0	0	0	0	0
C_3	0	0.1	**0.8**	0.06	0.03	0	0	0.01
C_4	0.0042	0.0126	0.0084	**0.9706**	0.0168	0	0	0
C_5	0	0.0057	0.0057	0.0114	**0.9486**	0.0229	0.0057	0
C_6	0	0	0	0	0.0217	**0.8768**	0.0072	0.0942
C_7	0.0619	0.0354	0	0	0.0088	0.0619	**0.8319**	0
C_8	0.01	0	0	0	0	0.02	0	**0.98**

Table 9: *k-Nearest Neighbor classifier (e_{kNN}) confusion matrix - Infrared image recognition*

Classifier	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
E_{BKS}	0.9408	0.0328	0.0264	0.9663
E_{XU}	0.9448	0.0336	0.0216	0.9657
E_{LU}	0.9632	0.0368	0	0.9632
E_{AP}	0.9536	0.0464	0	0.9536

Table 10: *Multiple classifier results - Infrared image recognition*

The confusion matrix presented in Table 11 shows that classes 3 and 6 ships are the least recognized, with recognition rates of 90.53% and 90.54%, respectively. With the simple classifiers, it has been mentionned that class 3 was much confused with class 4. This tendency is strongly reduced with the BKS multiple classifier since only 2.11% of class 3 tested ships are wrongly classified in class 4. However, class 3 is confused with class 2 at the rate of 7.37%. At 97.22%, the recognition rate of class 7 is much increased compared to those of the simple classifiers and the confusion with class 1 is reduced to 2.78%. Then the most important weaknesses in simple classifiers have been diminished and the reliability rate is relatively high at 96.63%. With a substitution rate of 3.28%, there are about as many ships that have been substituted with the BKS classifier than with the other tested multiple classifiers, though there are a few more rejected ships.

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_1	**0.9427**	0	0	0	0	0.0573	0	0
C_2	0	**1.0000**	0	0	0	0	0	0
C_3	0	0.0737	**0.9053**	0.0211	0	0	0	0
C_4	0	0	0.0087	**0.987**	0.0043	0	0	0
C_5	0	0.006	0	0.006	**0.9881**	0	0	0
C_6	0.0074	0	0	0	0	**0.9044**	0	0.0882
C_7	0.0278	0	0	0	0	0	**0.9722**	0
C_8	0	0	0	0	0	0.0204	0	**0.9796**

Table 11: *BKS multiple classifier (E_{BKS}) confusion matrix , $\lambda = 0$ - Infrared image recognition*

5.3.2 Lu's Multiple Classifier

Lu's multiple classifier [7] has a recognition rate of 95.84% when frequency histograms are used for the reliability function $R_i^k(Cf)$ and gets a 96.32% recognition rate when interpolated frequency histograms are used instead, that is better than Xu, Appriou and BKS multiple classifiers. Cf values have been calculated using the probabilities defined in (3), (11) and (15), and with $k = 15$ in (11). Applying that method to IR ship recognition needs a training phase in which the reliability function $R_i^k(Cf)$ is computed. In order to do so, the reliability function has been calculated according to intervals Cf. As done in [7], 10% width intervals have been chosen. For example, the first interval is $Cf =$ [0,10]. The reliability function can then be approximated in 10 points. Figure 21 shows an example of a frequency histogram produced with k-NN classifier for class 8 and Figure 22 presents the results of interpolating the frequency histogram of Figure 21. Also, it has been determined that when more than one class are returned, the decision is rejected.

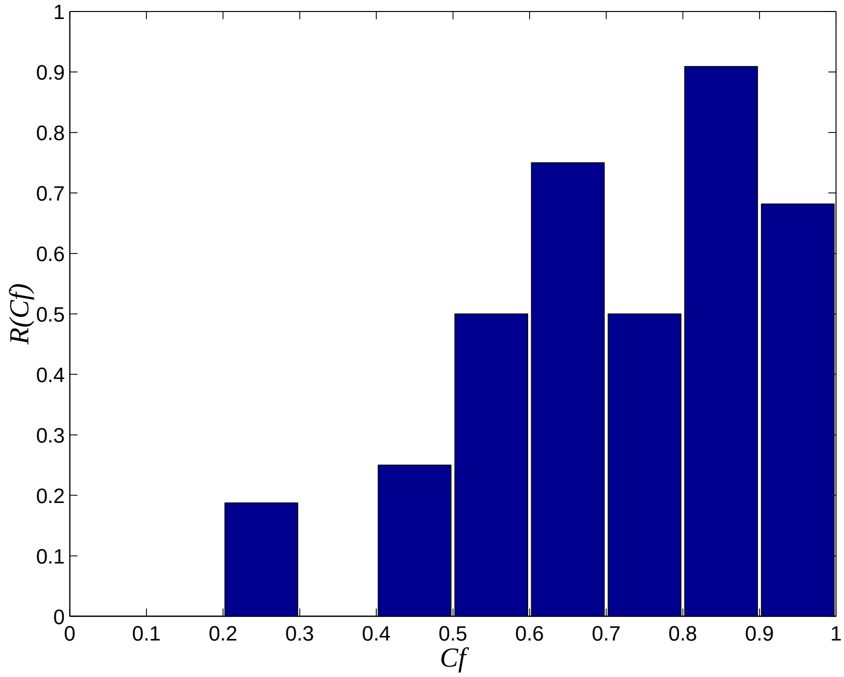


Figure 21: *10 interval frequency histogram of the reliability function R in terms of Cf for class 8 with k-NN classifier - Infrared image recognition*

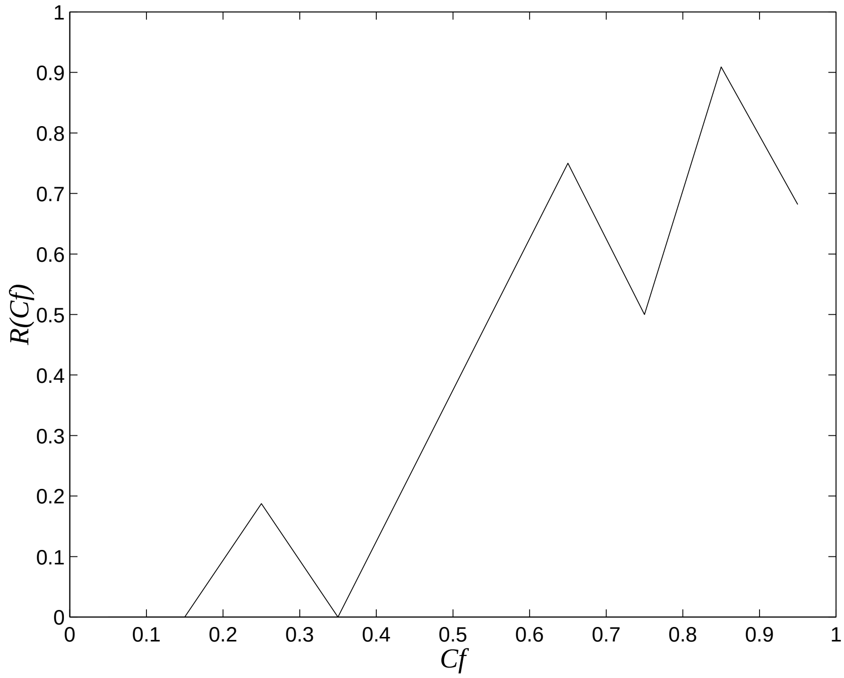


Figure 22: *Interpolation of the 10 interval frequency histogram of the reliability function R in terms of Cf for class 8 with k-NN classifier - Infrared image recognition*

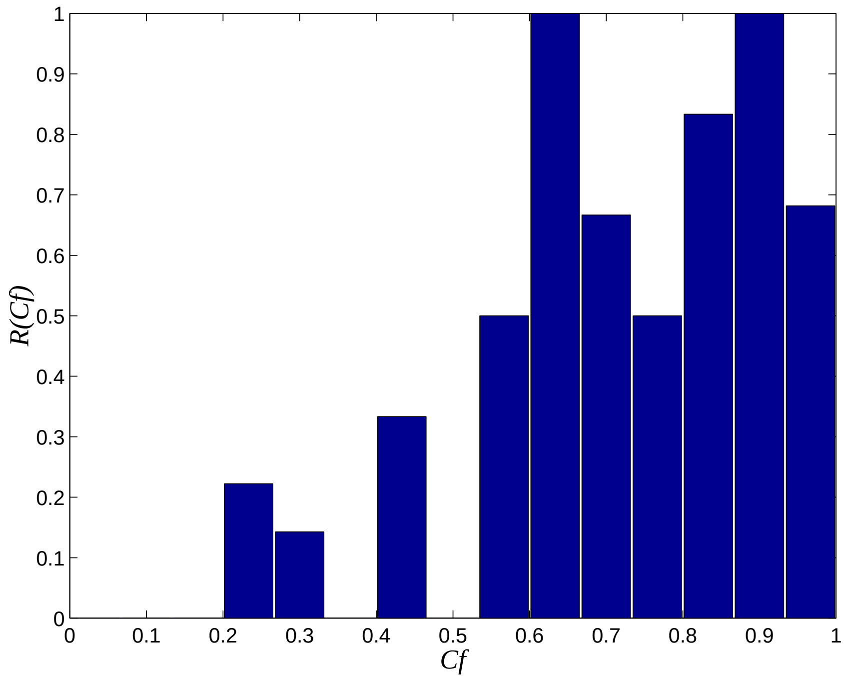


Figure 23: *15 interval frequency histogram of the reliability function R in terms of Cf for class 8 with k-NN classifier - Infrared image recognition*

Figures 23 and 24 present, as in Figure 21, the frequency histogram for class 8 with the k-NN classifier for 15 and 20 intervals, respectively instead of 10. They show that irregularities tend to increase as the number of intervals increases, and where empty intervals appear. With a too high interval number, that is higher than 10 in this case, the training data are thus represented too precisely and inadequately, so that errors have more chances to occur. Thus, the probability distribution density of Cf eggs on having a small interval number (near 10) rather than more.

Most of the $R_i^k(Cf)$ curves with 10 intervals are nearly monotonic. For example, the curve on Figure 22 can be estimated to be monotonic with some degree of tolerance. For classes were the $R_i^k(Cf)$ curves are not entirely monotic, Lu's method adjusts a threshold λ_{UC} to a value such that for all $Cf > \lambda_{UC}$, $R_i^k(Cf)$ can be approximated by monotonic curve. However, applying this rule brings on information losses when creating the BPAs. Also, the generated curves from the three individual classifier can be approximated, with some tolerance, by monotonic curves on the whole Cf domain. For these reasons and because it is difficult to determine the tolerance for judging whether or not a curve is monotonic, λ_{UC} threshold have all been set to 0.

The uncertainty added by the $R_i^k(Cf)$ functions over the uncertainty of Cf may represent an additional error source in that method. The confusion matrix in Table 12 shows that classes 1, 3 and 6 have relatively low recognition rates nearing 90%, so we can presume that the $R_i^k(Cf)$ functions for classes 1, 3 and 6 are not very good models of the problem.

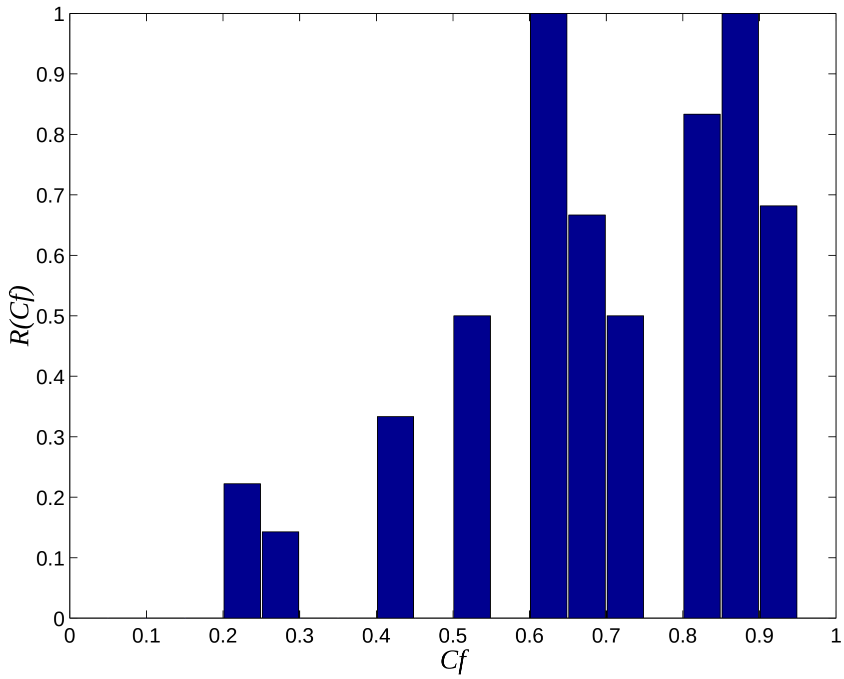


Figure 24: *20 interval frequency histogram of the reliability function R in terms of Cf for class 8 with k-NN classifier - Infrared image recognition*

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_1	**0.9074**	0.0185	0	0	0	0.0617	0.0123	0
C_2	0	**0.9955**	0	0.0045	0	0	0	0
C_3	0	0.0400	**0.9000**	0.0600	0	0	0	0
C_4	0	0	0	**0.9958**	0.0042	0	0	0
C_5	0	0	0	0.0057	**0.9886**	0.0057	0	0
C_6	0.0072	0	0	0	0	**0.9058**	0	0.0870
C_7	0	0.0088	0	0	0	0	**0.9912**	0
C_8	0	0.0100	0	0	0	0.0200	0	**0.9700**

Table 12: *Lu's classifier (E_{LU}) confidence matrix with interpolated frequency histograms, 10 intervals - Infrared image recognition*

5.3.3 Multiple Classifier with Appriou's Model of Initial Mass Assignment

Applying Appriou's model method to the IR ship images gives slightly poorer results than Lu's classifier, with a 95.36% recognition rate, a 4.64% substitution rate and a 95.36% reliability rate.

These tests are performed with *a priori* probability distributions $p(m_j|C_i)$ represented in 10 interval histograms, same as those used in Lu's classifier and with an example illustrated in Figure 21, where the $R(Cf)$ notation corresponds to $p(m_j|C_i)$ and where Cf corresponds to m_j. The probability distributions have been made using the probabilities defined in (3), (11) and (15), with $k = 15$ in (11). Confidence factors d_{ij} have been set to 1 for all simple classifiers because the confidence on each measure is considered to be the same, owing to the impossibility in such circumstances to know the precision related to the measures of each particular class and of each simple classifier. Also, $R_j = 1$ (normalization factor) for all combined classifiers j. The confusion matrix in Table 13 shows that this multiple classifier behaves the same as BKS, Lu and Xu's multiple classifiers, where objects of classes 3 and 6 get the lowest recognition rates. Class 3 is mostly confused with classes 2 and 4, while class 6 is confused with class 8 at 7.97%.

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_1	**0.9136**	0	0	0	0	0.0556	0.0309	0
C_2	0	**0.9955**	0	0.0045	0	0	0	0
C_3	0	0.0700	**0.9000**	0.0300	0	0	0	0
C_4	0	0	0.0168	**0.9748**	0.0042	0.0042	0	0
C_5	0	0.0057	0	0.0057	**0.9714**	0.0171	0	0
C_6	0.0072	0	0	0	0	**0.8841**	0.0290	0.0797
C_7	0.0088	0.0088	0	0	0.0088	0.0088	**0.9646**	0
C_8	0	0	0	0	0	0.0200	0	**0.9800**

Table 13: *Appriou's model multiple classifier (E_{AP}) confusion matrix with frequency histograms, 10 intervals - Infrared image recognition*

5.3.4 Xu's Multiple Classifier

Xu's method has a 94.48% recognition rate and a 96.57% reliability rate. As with the BKS method, the rejection rate is non-zero at 2.16%. The rejections do not come from the application of threshold α but from decisions that return more than one class as a final result. The initial BPAs are created using the values in Table 6. The results for that method, presented on Tables 10 and 14, are valid for λ between 0 and 0.7. When λ is above 0.7 , the rejection rate increases and the recognition rate diminishes. Thus, with λ=0.8 as well as with λ=0.9 , the results are : ϵ_r=91.76%, ϵ_s=1.68%, ϵ_{rej}=6.56% and ϵ_f=98.2021. Thereby, with $0.7 \leq \alpha < 1$ the reliability rate increases, but in counterpart the rejection rate is higher. With $0 \leq \lambda < 0.7$, the confusion matrix in Table 14 shows that class 3 is the least recognized again, with only 82% of recognitions, and that classes 6 and 7 give relatively poor results despite a small improvement compared to the simple classifiers. All objects belonging to class 2 are correctly identified, as with the BKS multiple classifier and the k-NN classifier.

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_1	**0.9444**	0	0	0	0	0.0309	0.0247	0
C_2	0	**1**	0	0	0	0	0	0
C_3	0	0.0900	**0.8200**	0.0600	0.0200	0	0	0.0100
C_4	0	0	0.0042	**0.9790**	0.0168	0	0	0
C_5	0	0.0057	0	0.0114	**0.9600**	0.0171	0.0057	0
C_6	0.0072	0	0	0	0.0145	**0.8841**	0.0072	0.0870
C_7	0.0354	0.0177	0	0	0.0088	0.0265	**0.9115**	0
C_8	0	0	0	0	0	0.0200	0	**0.9800**

Table 14: *Xu's classifier confusion matrix (E_{XU}) - Infrared image recognition*

5.4 Proposed Multiple Classifiers

5.4.1 Abstract Level Based Multiple Classifier

λ_t	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
0	0.9552	0.0448	0	0.9552
0.01	0.9584	0.0408	0.008	0.9592
0.02	**0.9632**	**0.0368**	**0**	**0.9632**
0.03	0.9592	0.0408	0	0.9592
0.04	0.9584	0.0416	0	0.9584
0.05	0.9488	0.0512	0	0.9488
0.07	0.9504	0.0496	0	0.9504
0.1	0.9528	0.0472	0	0.9528
0.2	0.9512	0.0488	0	0.9512
0.5	0.9520	0.048	0	0.9520
0.7	0.9488	0.0512	0	0.9488
0.8	0.936	0.0552	0	0.936
0.9	0.4616	0.0224	0.516	0.9537
1	0.1776	0.004	0.8184	0.978

Table 15: *Abstract Level Based multiple classifier (E_{ALB}) results in terms of λ_t - Infrared image recognition*

Applying the suggested method in 4.1.2 for creating the initial BPAs yields better results than BKS, Lu, Appriou's model and Xu's methods. Credibility rates are calculated using the confusion matrices in Tables 7, 8 and 9 and in terms of the decisions of the simple classifiers expressed in (4), (10) and (16). The k-NN classifier was tested with $k = 3$. According to that, the highest recognition rate obtained is 96.32% with λ_t=0.02. When $\lambda_t = 0$, where the initial BPAs are made of singletons defined in (41), the recognition rate drops to 95.52% but is still over those of the classical methods tested in 5.3. Table 15 shows that variating λ_t over a wide range of values does not change much the results. Thus, with $0 \leq \lambda_t \leq 0.8$ the difference between the maximum and the minimum recognition rates is lower than 2%. The more λ_t increases, the least is the number of propositions made for each simple classifier and the more elements there are in each propositions. With 0.05$\leq \lambda_t \leq$0.7 approximatively, the examination of the credibility rate values tells that most BPAs are created with two propositions, one having the class for the highest credibility rate and the other being its complement. With λ_t >0.9, the rejection rate is very high. Setting λ_t near 1 almost always results in a unique proposition containing all of the M classes, which represent ignorance. In that case, the pignistics probabilities for each class are all equal and then the object under consideration is rejected.

Thus, for small values of λ_t, that is for λ_t <0.05, the creation of the initial BPAs is more sensitive to λ_t variations and it is inside that range that the maximum recognition rate is reached. With 0.05< λ_t <0.7, most of the time the BPAs do not change because of the relatively high gap between the highest credibility rate and the remainder, so that the results are stable according to λ_t. Because the results with λ_t <0.05 are a slightly better than with 0.05< λ_t <0.7, we can assert that the imprecision on the credibility rates is not that big. Hence the BPAs made with λ_t <0.05 are a better representation of the evidence.

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_1	**0.9506**	0	0	0	0	0.0185	0.0309	0
C_2	0	**0.9911**	0.0045	0	0.0045	0	0	0
C_3	0	0.0300	**0.9400**	0.0200	0.0100	0	0	0
C_4	0.0042	0.0084	0.0210	**0.9496**	0.0168	0	0	0
C_5	0	0	0	0	**0.9943**	0.0057	0	0
C_6	0.0145	0	0	0	0.0072	**0.9565**	0	0.0217
C_7	0.0265	0.0265	0	0	0	0	**0.9469**	0
C_8	0.0100	0	0	0	0	0.0400	0	**0.9500**

Table 16: *Abstract Level Based multiple classifier (E_{ALB}) confusion matrix - Infrared image recognition*

Table 16 shows the confusion matrix for λ_t =0.02. Class 3 ships are still the most misclassified ones although their recognition rate is up to 94%, that is 10% better than the best rate of the three simple classifiers for that class. Confusions with class 4 are reduced to 2%. However, class 3 has a 3% confusion rate with class 2. This is higher than with the neural network and the k-NN classifiers. Also, 4% of class 8 ships are classified in class 6, compared to 2% with the k-NN classifier, and the recognition rate for class 2 is lower to the 100% rate obtained with the k-NN classifier. New errors that were not existing with the simple classifiers are then introduced, because of the fact that credibility rates are an imprecise information source. In order to illustrate that, let us suppose that a class 2 ship, represented by vector $\mathbf{x}$, is tested with classifier E_{ALB} and suppose that the three individual classifiers return $e_B(\mathbf{x}) = 4$, $e_{RN}(\mathbf{x}) = 4$ and $e_{kPPV}(\mathbf{x}) = 2$. To have $E_{ALB}(\mathbf{x}) = 2$, the credibility rates for class 2 associated to the answers $e_B(\mathbf{x}) = 4$ and $e_{RN}(\mathbf{x}) = 4$ would have to be very high. For example, with λ_t=0, the sum of the credibility rates for class 2 should be higher than the sum of those associated to class 4, what is unlikely to happen because two simple classifiers out of three have returned class 4 as the answer.

5.4.2 Measure Level Based Multiple Classifier

Unlike the other multiple classifiers previously tested, including the Abstract Level Based multiple classifier, the Measure Level Based multiple classifier does not need any training or *a priori* information, like the recognition and substitution rates in Xu's multiple classifier for example. The Measure Level Based multiple classifier is the one that gives the best results. It is demonstrated in Table 17 where the results of all tested classifiers are shown. The recognition rate for the MLB multiple classifier reaches 98.24% with λ_t=0.01 for Bayes and neural network BPAs, and with parameters k=15 and c=1.3 for k-NN. The threshold λ_{REJ} has been set to 1 and thereby the rejection rate is zero. The confusion matrix is shown in Table 18. Class 6 is the one with the lowest recognition rate at 94.93%. It is confused with class 8 by a 3.62% rate, which corresponds to 6 ships in the database. Ships in classes 8 and 2 are all recognized without any exceptions, while classes 4 and 5 give almost perfect recognition rates, with 99.58% and 98.86%, respectively, and which corresponds to one and two misclassified ships only.

Classifier	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
e_B	0.7712	0.2288	0	0.7712
e_{NN}	0.9272	0.0728	0	0.9272
e_{kNN}	0.9288	0.0712	0	0.9288
E_{BKS}	0.9408	0.0328	0.0264	0.9663
E_{XU}	0.9448	0.0336	0.0216	0.9657
E_{LU}	0.9632	0.0368	0	0.9632
E_{AP}	0.9536	0.0464	0	0.9536
E_{ALB}	0.9632	0.0368	0	0.9632
E_{MLB}	**0.9824**	**0.0176**	**0**	**0.9824**

Table 17: *Results summary - Infrared image recognition*

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_1	**0.9630**	0	0	0	0	0.0370	0	0
C_2	0	**1**	0	0	0	0	0	0
C_3	0	0	**0.9700**	0.0300	0	0	0	0
C_4	0	0.0042	0	**0.9958**	0	0	0	0
C_5	0	0	0	0.0057	**0.9886**	0.0057	0	0
C_6	0.0145	0	0	0	0	**0.9493**	0	0.0362
C_7	0.0177	0	0	0	0.0088	0	**0.9735**	0
C_8	0	0	0	0	0	0	0	**1**

Table 18: *Measure Level Based multiple classifier (E_{MLB}) confusion matrix, λ_t=0.01, k=15, $c = 1.3$ and $\lambda_{REJ} = 1$ - Infrared image recognition*

Of the 1250 IR ship images tested, only 22 are misclassified. Among those 22 images, six can be considered impossible cases because all three decisions of the simple classifiers are wrong, so it can be deduced that the measures are wrong too. These six ships could possibly be involved in rejected decisions if λ_{REJ} was lowered. The 16 other ships are classified correctly by at least two simple classifiers out of the three, but combining the related measures has lead to wrong decisions. Among them, ten can almost be considered to be impossible cases because they turned out to be wrongly substituted not only with the Measure Level Base multiple classifier but also with the Abstract Level Base multiple classifier (Table 19). This can be caused by feature values that are dissimilar from the conventional pattern for that class. Six of the 22 misclassified ship images by E_{MLB} are well classified by E_{ALB}, indicating that, in some cases, the algorithm for creating the initial BPAs using the abstract level can bring better decisions than the algorithm using measures. Nevertheless, the comparison of the substitution rates of both methods, that is 1.76% for E_{MLB} compared to 3.68% for E_{ALB}, shows that their are much more errors done by E_{ALB} that are corrected by E_{MLB} than the opposite.

The decision threshold λ_{REJ} has a great influence over the results. By diminishing λ_{REJ}, it is possible to obtain a 100% reliability rate. On the other hand, the rejection rate is higher. Table 20 shows that ϵ_f is 100% for λ_{REJ} ranging from 0 to 0.4. With λ_{REJ}=0.4, 68.64% of ships are classified successfully while all others are rejected. Figures 25 and 26 illustrate how the recognition, the rejection and the reliability rates vary in terms of λ_{REJ}.

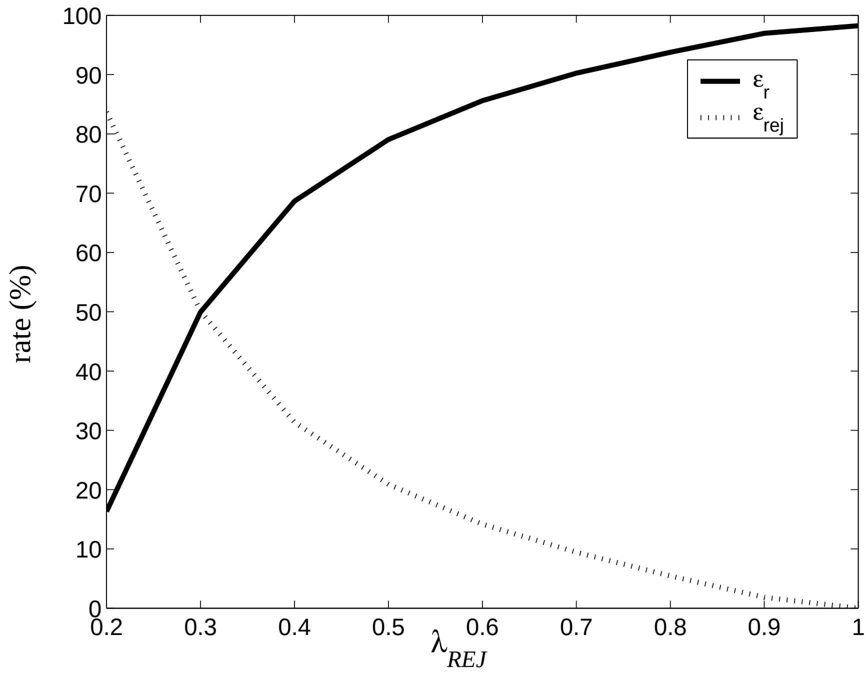


Figure 25: *Recognition rate and rejection rate in terms of λ_{REJ} for Measure Level Based multiple classifier, with λ_t=0.01, k=15 and c=1.3 - Infrared image recognition*

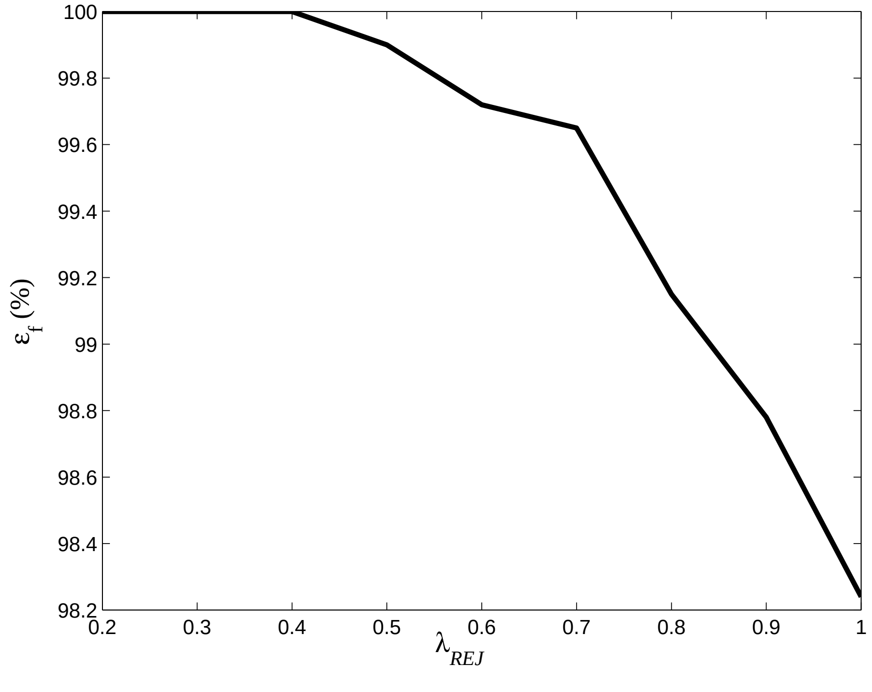


Figure 26: *Reliability rate in terms of λ_{REJ} for Measure Level Based multiple classifier, with λ_t=0.01, k=15 and c=1.3 - Infrared image recognition*

Class	j_B	j_{kNN}	j_{NN}	j_{MLB}	E_{ALB} Substitution	Impossible case
1	6	6	6	**6**	•	•
1	1	1	6	**6**		
1	1	1	6	**6**		
1	1	6	1	**6**		
1	1	1	6	**6**		
1	1	6	6	**6**	•	
3	3	4	4	**4**	•	
3	4	4	3	**4**	•	
3	4	4	4	**4**	•	•
4	4	3	4	**2**		
5	4	6	5	**6**	•	
5	3	2	5	**4**	•	
6	8	8	6	**8**	•	
6	8	8	6	**8**	•	
6	8	8	6	**8**	•	
6	1	6	1	**1**	•	
6	8	8	7	**8**	•	•
6	8	8	1	**1**	•	•
6	8	8	5	**8**	•	•
7	2	1	1	**1**	•	•
7	2	5	7	**2**	•	
7	7	1	7	**1**		

Table 19: *Misclassified ships by Measure Level Based multiple classifier*

Variables λ_t, k and c have less influence on the results. Generally, their values can be set in a wide range without affecting significantly the performance of the multiple classifier. In a similar way to the ALB multiple classifier, variable λ_t enables to have the best results when it is near 0 (0.01 in the tests with the IR ship database). From there, as λ_t increases the recognition rate drops (Figure 27) and the rejection rate rises (Figure 28). With optimum values for c and λ_t, changing k has not much influence on the performance although the latter is affected when k is under 15, as shows Figure 29. The stability with the variations of k is caused by the fact that k has influence almost only on proposition A_2 (53) of the k-NN classifier, if we consider that k is two high (or c too small) to have considerable effect on proposition A_1 (52). Also, tests have been conducted with different values of c and they demonstrate that the results are quite stable for $[0 \leq c \leq 0.5]$ (Figure 30). As for k, c has a partial influence in the making of the propositions since it plays a role only in proposition A_1 of k-NN, and Dempster's combination makes so that the results do not fluctuate much because when c changes only one proposition changes among all. Since λ_t, k and c have some effects in different propositions in the creation of the BPAs, the results are not much affected by the change of only one of these variable. However, if λ_t, k and c would all have extreme values far from the ones presented above, then the results would drop. For example, with λ_t =0.01, $k = 50$ and $c = 50$, the recognition rate falls to 96.32% and the substitution rate is 3.68%. As more, with λ_t =0.5, $k = 50$ and $c = 50$, although the substitution rate is 0.32%, the initial BPAs created resulted in a high rejection rate of

λ_{REJ}	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
0.2	16.32	0	83.68	100
0.3	49.92	0	50.08	100
0.4	68.64	0	31.36	100
0.5	79.04	0.08	20.88	99.9
0.6	85.6	0.24	14.16	99.72
0.7	90.24	0.32	9.44	99.65
0.8	93.76	0.8	5.44	99.15
0.9	96.96	1.2	1.84	98.78
1	98.24	1.76	0	98.24

Table 20: *Measure Level Based multiple classifier results in terms of λ_{REJ}, with $\lambda_t = 0.01$, k=15 and $c = 1.3$ - Infrared image recognition*

30.88%, along with a 68.8% recognition rate.

In the light of these results, the MLB multiple classifier is the best one among all of the tested multiple classifiers for the recognition of IR ships images. Its recognition rate is at least 4% higher than any of the other existing multiple classifiers and 2% higher than the ALB multiple classifier, what is a relatively good improvement considering that it is difficult to improve results that are already high (above 90%). With λ_{REJ} nearing 1, high reliability can be reached without having any rejection, as opposed to the simple classifiers. Because the results of this multiple classifier are better than those of the ALB multiple classifier, it seems more obvious that for the combination of simple classifiers, even if the abstract level is weighted with credibility rates, the loss of information it represents compared with the measure level justifies the use of the measures instead of the final answers.

5.5 Conclusion

Of the three simple classifiers tested with the IR ship database [23], two have similar results with a recognition rate around 92%. The other, the Bayes one, has a recognition rate of 77%. It is not that good compared with the k-NN and neural network classifiers. The common weaknesses of these simple classifiers are the confusions of class 3 with class 4, class 7 with class 1 and class 1 with class 7. Most of the multiple classifiers tested reduce these specific problems, but they sometimes create new ones elsewhere. The MLB multiple classifier is the one that reduces the most the class confusions and it is the one that gives the best results overall. The analysis of the 22 misclassified ships with the MLB classifier shows that 16 of them can be considered as exceptions whose features values are probably wrong or too much similar to typical values of other classes (see Table 4 and Table B.1). Therefore only six errors can be attributed to some lacks in the algorithm itself. Compared with others, the Measure Level Based multiple classifier shows that the information that represents best the uncertainty are the measures of the simple classifiers. Finally, the results on the IR ship images reveal that, although the measures are themselves uncertain and inaccurate, their uncertainty and imprecision seem less important than those related to the various *a priori* knowledge used by the other multiple classifiers.

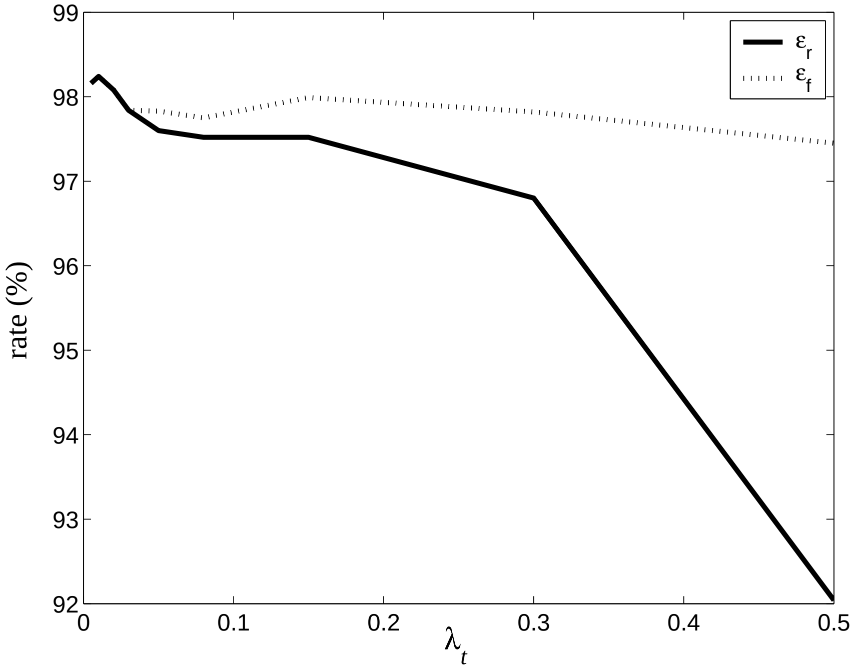


Figure 27: *Recognition rate and reliability rate in terms of λ_t for Measure Level Based multiple classifier, with $k = 15$, c=1.3 and $\lambda_{REJ} = 1$ - Infrared image recognition*

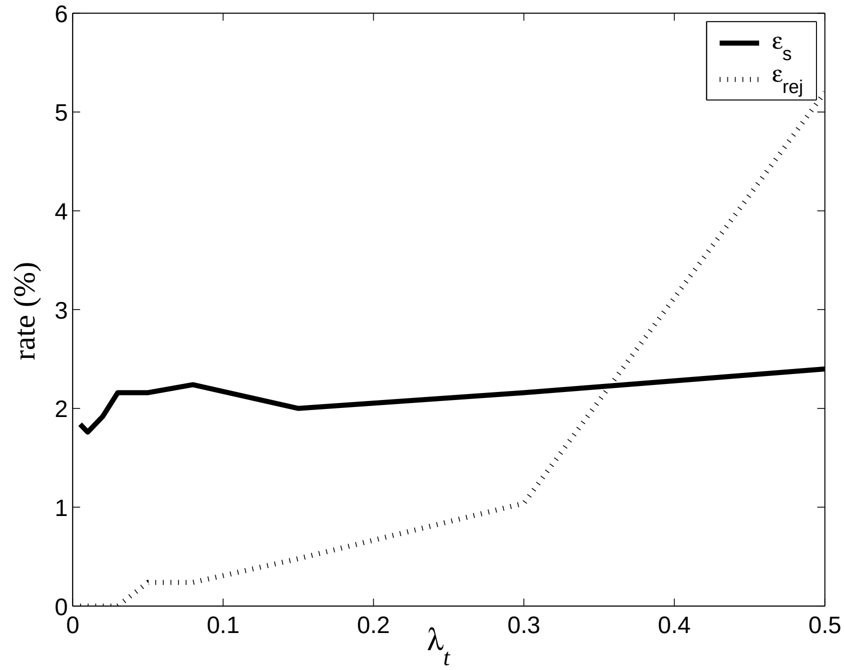


Figure 28: *Substitution rate and rejection rate in terms of λ_t for Measure Level Based multiple classifier, with $k = 15$, c=1.3 and $\lambda_{REJ} = 1$ - Infrared image recognition*

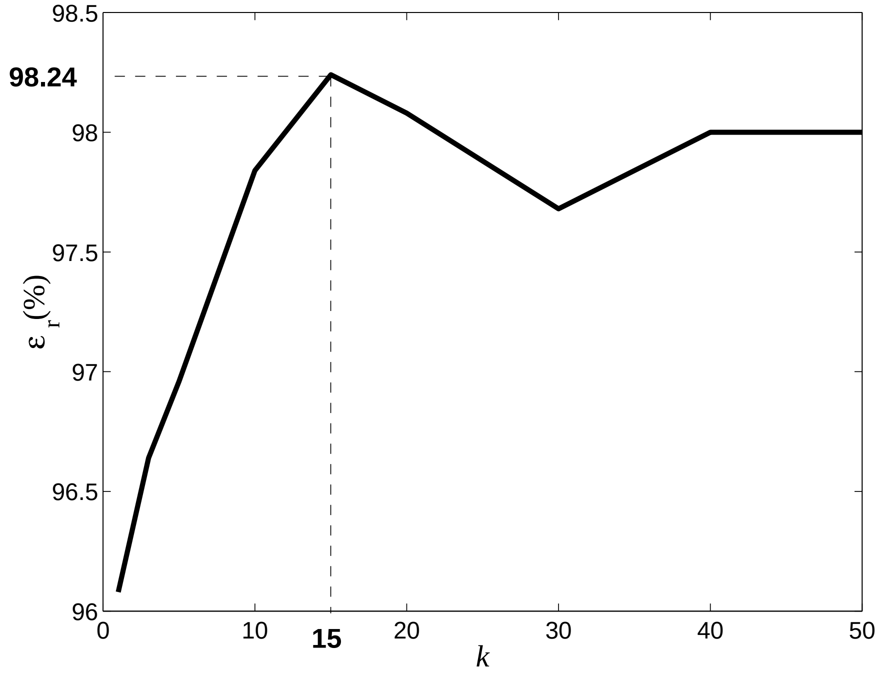


Figure 29: *Recognition rate in terms of k (number of neighbors of k-NN classifier) for Measure Level Based multiple classifier, with λ_t=0.01, c=1.3 and λ_{REJ} = 1 - Infrared image recognition*

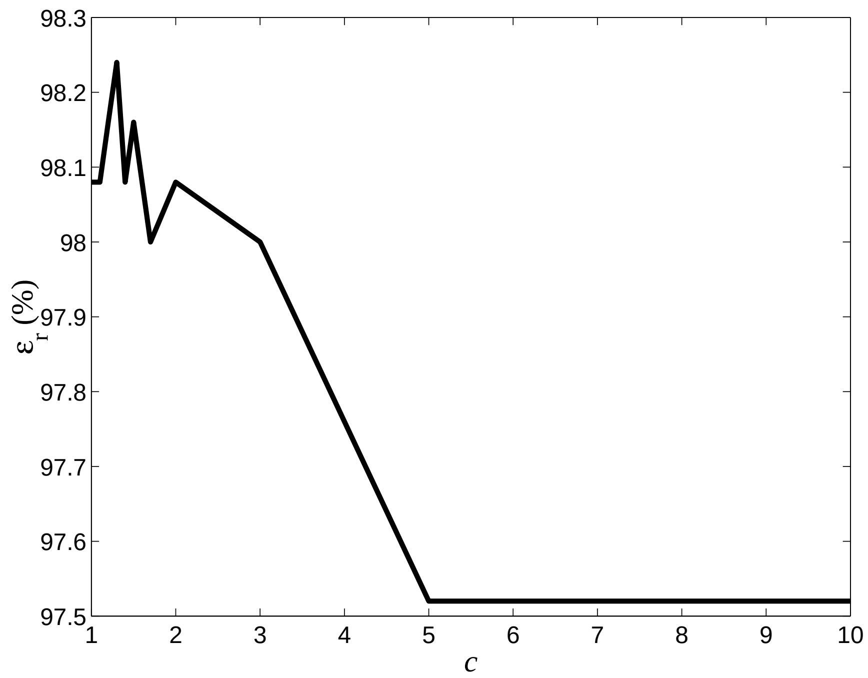


Figure 30: *Recognition rate in terms of c for Measure Level Based multiple classifier, with λ_t=0.01, k=15 and λ_{REJ} = 1 - Infrared image recognition*

6 Application to Handwritten Digit Recognition

Handwritten digit recognition is performed using the different methods studied and proposed in this work. A digit database created by the Department of Computer Engineering at Bogazici University, Istanbul, Turkey, is used, where the sampled digits are represented by 16 attributes. All tests are generally conducted the same way as in Chapter 5 and the results will be presented the same way. Any difference with the procedures in Chapter 5 is indicated.

6.1 Database

The database was created by collecting 250 samples from 44 writers [28]. Writers were asked to write 250 digits in random order inside boxes of 500 by 500 tablet pixel resolution. There are 7494 training instances written by 30 persons and 3498 testing instances written by the other 14 persons. All 16 attributes are integers in the range $[0, 100]$. They correspond to different normalized (x, y) coordinate information. Table 21 is an example of some of the digits in the database.

Λ	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
9	91	69	48	57	9	79	60	100	100	75	95	40	64	8	0	0
3	30	74	55	100	89	87	66	56	100	38	92	8	41	0	0	20
7	5	65	0	89	37	100	88	97	100	79	71	53	48	26	59	0
0	42	93	19	88	0	42	24	0	83	11	100	56	75	100	17	97
4	4	100	0	72	15	44	79	50	100	76	90	51	83	22	85	0
6	100	100	79	80	54	59	32	38	21	16	50	0	79	16	0	16
8	0	76	30	48	53	9	11	0	47	34	97	66	100	100	38	85

Table 21: *Handwritten digit database sample*

Tables 22 and 23 show the proportion of each class in the training and testing databases. The proportion of each class is about 10% in both the training and the testing databases.

Class	0	1	2	3	4	5	6	7	8	9
Number	780	779	780	719	780	720	720	778	719	719
Proportion (%)	10.41	10.39	10.41	9.59	10.41	9.61	9.61	10.38	9.59	9.59

Table 22: *Proportion of each class in the training database (7494 instances)*

Class	0	1	2	3	4	5	6	7	8	9
Number	363	364	364	336	364	335	336	364	336	336
Proportion (%)	10.38	10.4	10.4	9.61	10.4	9.58	9.61	10.4	9.58	9.58

Table 23: *Proportion of each class in the testing database (3498 instances)*

As in Chapter 5, for the multiple classifiers that need a training period, such as BKS classifier, Lu's classifier and Appriou's model method, training is done by inverting the training and testing databases, though tests are still made with the testing database of 3498 instances. For example, the *a priori* probability distributions in Appriou and Lu's classifiers are calculated using the training database (7494 instances) as inputs to the simple classifiers and the testing database (3498 instances) is used to train the simple classifiers.

6.2 Simple Classifiers

As Table 24 shows and similarly to the tests in Chapter 5, k-NN and neural network have close recognition rates, with 97.83% and 96.86%, respectively. Bayes classifier still has the worst rate with 74.64%. The rejection rate (ϵ_{rej}) is zero with all three classifiers.

Classifier	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
e_B	0.7464	0.2536	0	0.7464
e_{NN}	0.9686	0.0314	0	0.9686
e_{kNN}	0.9783	0.0217	0	0.9783

Table 24: *Individual classifier results - Handwritten digit recognition*

Table 25 shows the confusion matrix for the Bayes classifier. The matrix is obtained with the following parameters : The M classes are equiprobable and thus, $P(I = i) = 1/M$, $i \in \Lambda$; *a priori* distributions have been calculated with 40 intervals ($Q = 40$). The training database is used (7494 digits) to calculate the *a priori* distributions. The decision rule used is the one defined by (4).

Input \ Ouput	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_0	**0.8044**	0.0165	0	0	0.0055	0.0110	0.0716	0	0.0909	0
C_1	0	**0.4423**	0.3654	0.1374	0.0027	0.0055	0.0137	0.0192	0	0.0137
C_2	0	0.0385	**0.9121**	0.0440	0	0.0027	0.0027	0	0	0
C_3	0.0030	0.0238	0	**0.9673**	0	0	0	0	0	0.0060
C_4	0	0.0082	0.0027	0.0027	**0.8956**	0	0.0907	0	0	0
C_5	0.0209	0	0	0.3433	0	**0.4866**	0.0567	0.0030	0	0.0896
C_6	0	0	0.0119	0	0	0	**0.9762**	0.0119	0	0
C_7	0	0.1264	0.0027	0.0027	0.0082	0	0.0357	**0.7857**	0.0055	0.0330
C_8	0.1786	0.0060	0.0208	0	0	0.0982	0.0774	0.0893	**0.5298**	0
C_9	0	0.0744	0	0.2143	0.0506	0	0.0030	0	0.0030	**0.6548**

Table 25: *Bayes classifier (e_B) confusion matrix - Handwritten digit recognition*

The neural network is configured with two layers, of respectively 50 and 30 neurons. The confusion matrix shown in Table 26 is generated with the following configuration [26, 27]:

- gain: 0.2;
- momentum: 0.5;
- maximum output error: 0.0001;
- epsilon (ϵ): 0.1;

- maximum iterations: 100.

Input \ Output	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_0	**0.9421**	0	0	0	0	0	0.0138	0	0.0413	0.0028
C_1	0	**0.9780**	0.0082	0.0055	0.0055	0	0	0.0027	0	0
C_2	0	0.0055	**0.9918**	0	0	0	0	0.0027	0	0
C_3	0	0.0030	0	**0.9911**	0	0	0	0	0	0.0060
C_4	0	0.0082	0	0	**0.9780**	0.0027	0.0027	0	0	0.0082
C_5	0	0	0	0.0090	0	**0.9791**	0	0	0	0.0119
C_6	0	0.0030	0	0	0	0	**0.9970**	0	0	0
C_7	0	0.0659	0.0110	0.0110	0	0	0.0027	**0.8681**	0.0027	0.0385
C_8	0.0030	0.0030	0	0	0	0.0030	0	0.0030	**0.9851**	0.0030
C_9	0	0.0030	0	0	0	0.0030	0	0.0089	0.0030	**0.9821**

Table 26: *Neural Network classifier (e_{NN}) confusion matrix - Handwritten digit recognition*

Table 27 shows the confusion matrix for k-NN classifier, where the distances are calculated according to (8) and (9). The value of k is set to 3 as in Chapter 5 (only $k = 4$ produces a higher recognition rate of 97.88% instead of 97.83% with k=3). Tests results with the k-NN classifier are presented in Table 24.

Input \ Output	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_0	**0.9752**	0	0	0	0	0	0.0138	0	0.0083	0.0028
C_1	0	**0.9615**	0.0302	0	0.0027	0	0	0.0055	0	0
C_2	0	0.0055	**0.9945**	0	0	0	0	0	0	0
C_3	0	0.0030	0	**0.9911**	0	0	0	0	0	0.0060
C_4	0	0	0	0	**0.9725**	0.0275	0	0	0	0
C_5	0	0	0	0.0149	0	**0.9761**	0	0.0030	0	0.0060
C_6	0	0	0	0	0	0	**1.0000**	0	0	0
C_7	0	0.0275	0.0027	0	0	0	0.0027	**0.9643**	0.0027	0
C_8	0.0030	0	0	0	0	0.0030	0	0	**0.9940**	0
C_9	0	0.0030	0	0.0149	0.0030	0.0089	0	0.0119	0.0030	**0.9554**

Table 27: *k-Nearest Neighbor classifier (e_{kNN}) confusion matrix - Handwritten digit recognition*

Here fluctuations in the recognition rate for each class cannot be related to their proportions in the database, because each one makes about 10% of both training and testing databases. k-NN is the one where those fluctuations are the less, where the lowest recognition rate is 95.54% with class 9 and the highest is 100% with class 6. The results of the neural network are also stable except with class 7 that gives 86.81 %, that is the poorer recognition rate and the only below 94%. With the Bayes classifier, many classes have bad recognition rates. Classes 1, 5 and 8 all have rates below 55% and class 9 only have 65.48%. The main points here are that Bayes classifier gives results way below the ones of k-NN and NN. The recognition rate of k-NN is very high and may be difficult to improve with a multiple classifier.

6.3 Existing Multiple Classifiers

BKS (Subsection 3.3.1), Lu (Subsection 3.3.2), Xu (Subsection 3.3.3) and the Appriou's model (Subsection 3.3.4) multiple classifiers are tested on the 3498 digit testing database, where the Bayes, the k-NN and the neural network classifiers are combined.

None of them has been able to improve over k-NN, in fact they all have recognition rates below the one of k-NN, as shown in Table 28. Xu and BKS methods provide reliability rates about the same as k-NN. In all four of these methods, the problem is with class 7. All other classes are between 94% and 100% while class 7 is always below 91% : 90.61% with BKS, 82.14% with Lu, 87.64% with Appriou and 90.34% with Xu. The reason for this must be related to the fact that Bayes and NN get low rates for that class (78.57% and 86.81%, respectively).

Classifier	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
E_{BKS}	0.9651	0.0212	0.0137	0.9786
E_{XU}	0.9626	0.0263	0.0111	0.9734
E_{LU}	0.9577	0.0423	0	0.9577
E_{AP}	0.9663	0.0337	0	0.9663

Table 28: *Multiple classifier results - Handwritten digit recognition*

6.3.1 Behavior Knowledge Space Multiple Classifier

The BKS multiple classifier is tested with $\lambda = 0$. This classifier has the highest reliability rate at 97.86% and its recognition rate is of 96.51%. Its confusion matrix is presented in Table 29.

Input \ Output	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_0	**0.9856**	0	0	0	0	0	0	0	0.0144	0
C_1	0	**0.9860**	0.0056	0.0028	0.0028	0	0	0.0028	0	0
C_2	0	0.0055	**0.9945**	0	0	0	0	0	0	0
C_3	0	0.0030	0	**0.9911**	0	0	0	0	0	0.0060
C_4	0	0	0	0	**0.9943**	0.0029	0	0	0	0.0029
C_5	0	0	0	0.0150	0	**0.9731**	0	0	0.0030	0.0090
C_6	0	0	0	0	0	0	**1.0000**	0	0	0
C_7	0	0.0663	0.0028	0	0	0	0	**0.9061**	0.0193	0.0055
C_8	0	0	0	0	0	0.0030	0	0	**0.9970**	0
C_9	0	0.0030	0	0	0	0.0089	0	0.0089	0.0179	**0.9613**

Table 29: *BKS multiple classifier (E_{BKS}) confusion matrix , λ = 0 - Handwritten digit recognition*

6.3.2 Lu's Multiple Classifier

Lu's multiple classifier confusion matrix is shown in Table 30. The recognition rate as well as the reliability rate are the lowest of all at 95.77%. Cf values have been calculated using the probabilities defined in (3), (11) and (15), and with $k = 15$ (11). For the same reasons as in Chapter 5, 10% width intervals are chosen. Most of the $R_i^k(Cf)$ curves with 10 intervals are nearly monotonic. λ_{UC} thresholds were all set to 0. When tested with 15 intervals, the recognition rate drops slightly to 95.31% instead of 95.77% with 10 intervals.

Input \ Output	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_0	**0.9449**	0	0	0	0	0	0.0055	0	0.0496	0
C_1	0	**0.9698**	0.0192	0	0.0027	0	0	0	0	0.0082
C_2	0	0	**0.9918**	0	0	0	0	0	0	0.0082
C_3	0	0.0030	0	**0.9851**	0	0	0	0	0	0.0119
C_4	0	0.0082	0	0	**0.9560**	0	0.0027	0	0.0137	0.0192
C_5	0	0	0	0.0179	0	**0.9582**	0	0	0.0030	0.0209
C_6	0	0	0	0	0	0	**0.9792**	0	0.0149	0.0060
C_7	0	0.0934	0.0082	0	0	0	0	**0.8214**	0.0302	0.0467
C_8	0	0	0	0	0	0.0030	0	0	**0.9970**	0
C_9	0	0.0060	0	0	0	0	0	0.0089	0.0030	**0.9821**

Table 30: *Lu's classifier (E_{LU}) confidence matrix with interpolated frequency histograms, 10 intervals - Handwritten digit recognition*

6.3.3 Multiple Classifier with Appriou's Model of Initial Mass Assignment

The confusion matrix for Appriou's model is presented in Table 31 and is obtained with *a priori* probability distributions $p(m_j|C_i)$ represented in 10 interval histograms. The probability distributions have been made using the probabilities defined in (3), (11) and (15), and with $k = 15$ (11). The recognition rate and reliability rate are at 96.63%. Confidence factors d_{ij} have been set to 1 for all simple classifiers. $R_j = 1$ for all combined classifiers j.

Input \ Output	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_0	**0.9449**	0	0	0	0	0	0.0028	0	0.0523	0
C_1	0	**0.9670**	0.0302	0	0.0027	0	0	0	0	0
C_2	0	0.0055	**0.9945**	0	0	0	0	0	0	0
C_3	0	0.0060	0	**0.9881**	0	0	0	0	0	0.0060
C_4	0	0.0055	0	0	**0.9698**	0.0165	0.0027	0	0	0.0055
C_5	0	0	0	0.0179	0	**0.9672**	0	0	0.0030	0.0119
C_6	0	0	0	0	0	0	**0.9970**	0	0.0030	0
C_7	0	0.0852	0.0055	0	0	0	0	**0.8764**	0.0082	0.0247
C_8	0.0060	0	0	0	0	0.0030	0	0.0089	**0.9821**	0
C_9	0	0.0089	0	0	0	0	0	0.0089	0	**0.9821**

Table 31: *Appriou's model multiple classifier (E_{AP}) confusion matrix with frequency histograms, 10 intervals - Handwritten digit recognition*

6.3.4 Xu's Multiple Classifier

Xu's multiple classifier provides a recognition rate of 96.26% and a reliability rate of 97.34%. Again, that is below k-NN's performance. The initial BPAs are created using the values in Table 32. Table 33 shows the results for $\lambda = 0$.

Classifier	ϵ_r	ϵ_s
e_B	0.77.69	0.2231
e_{NN}	0.9606	0.0394
e_{kNN}	0.9485	0.0515

Table 32: *Individual classifier results using the 7494 digit database for testing and the 3498 digit database for training - Handwritten digit recognition*

Input \ Output	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_0	**0.9634**	0	0	0	0	0	0.0141	0	0.0225	0
C_1	0	**0.9615**	0.0330	0.0027	0.0027	0	0	0	0	0
C_2	0	0.0055	**0.9945**	0	0	0	0	0	0	0
C_3	0	0.0030	0	**0.9911**	0	0	0	0	0	0.0060
C_4	0	0	0	0	**0.9944**	0.0028	0.0028	0	0	0
C_5	0	0	0	0.0181	0	**0.9759**	0	0	0	0.0060
C_6	0	0	0	0	0	0	**1.0000**	0	0	0
C_7	0	0.0710	0.0028	0	0	0	0	**0.9034**	0	0.0227
C_8	0.0060	0	0	0	0	0.0030	0	0	**0.9910**	0
C_9	0	0.0060	0	0.0150	0.0030	0.0030	0	0.0090	0.0030	**0.9611**

Table 33: *Xu's classifier confusion matrix (E_{XU}) - Handwritten digit recognition*

6.4 Proposed Multiple Classifiers

6.4.1 Abstract Level Based Multiple Classifier

The Abstract Level Based multiple classifier, although it gives poorer results than k-NN, has a recognition rate slightly higher than BKS, Lu, Appriou and Xu's methods. A recognition rate of 97.03% is obtained with λ_t=0.02. Credibility rates are calculated using the confusion matrices in Tables 34, 35 and 36, that are created by inverting the testing and training databases, that is with the 7494 digit database for testing and with the 3498 digit database for training. The credibility rates comes from the decisions of the simple classifiers expressed in (4), (10) and (16). k-NN was tested with $k = 3$ to obtain the credibility rates. Table 38 presents the confusion matrix for $\lambda_t = 0.02$ and Table 37 shows the results in terms of λ_t. The confusion matrix shows that the main weakness is still with class 7 where only 88.46% digits are recognized.

6.4.2 Measure Level Based Multiple Classifier

Of all the multiple classifier tested for handwritten digit recognition, the Measure Level Based multiple classifier is the one that gives the best results. With λ_t=0.01, $k = 100$, $c = 1.3$ and $\lambda_{REJ} = 1$, it gives the highest recognition rate (97.43%). Again, it is still lower than k-NN but it is the highest among all multiple classifiers. Lowering λ_{REJ} increases the reliability rate (Table 41). With $\lambda_{REJ} = 0.6$ a recognition rate of 98.11% is reached while maintaining a relatively high recognition rate of 96.51%. See Section C for more details.

Input \ Output	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_0	**0.9256**	0.0013	0	0	0.0077	0	0.0103	0	0.0538	0.0013
C_1	0.0013	**0.7176**	0.1284	0.0385	0.0064	0.0077	0	0.0026	0	0.0976
C_2	0	0.0333	**0.8500**	0.0987	0	0.0013	0.0064	0.0064	0	0.0038
C_3	0	0.0181	0.0014	**0.9569**	0	0	0	0	0	0.0236
C_4	0.0205	0.0013	0	0	**0.9346**	0	0.0410	0	0.0013	0.0013
C_5	0.1264	0.0028	0	0.1917	0	**0.5153**	0.0167	0.0014	0.0236	0.1222
C_6	0.0069	0	0.0042	0	0.0500	0.0014	**0.9375**	0	0	0
C_7	0	0.0951	0.0077	0.0039	0.0026	0	0.0129	**0.8445**	0.0039	0.0296
C_8	0.2921	0.0779	0.0250	0.0181	0.0028	0.0306	0.0389	0.1057	**0.3936**	0.0153
C_9	0.0306	0.0640	0	0.0570	0.1739	0.0070	0.0028	0.0028	0.0014	**0.6606**

Table 34: *Bayes classifier confusion matrix with the 7494 digit database for testing and the 3498 digit database for training - Handwritten digit recognition*

Input \ Output	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_0	**0.9744**	0	0	0	0.0192	0	0.0026	0.0013	0	0.0026
C_1	0	**0.8947**	0.0385	0.0449	0.0013	0.0013	0.0051	0.0128	0	0.0013
C_2	0	0.0077	**0.9923**	0	0	0	0	0	0	0
C_3	0	0.0056	0.0014	**0.9875**	0	0.0028	0	0.0028	0	0
C_4	0	0.0013	0	0	**0.9962**	0	0	0.0013	0	0.0013
C_5	0	0.0014	0	0.0042	0	**0.9875**	0	0	0.0028	0.0042
C_6	0	0	0	0	0.0028	0.0014	**0.9944**	0.0014	0	0
C_7	0	0.0231	0.0013	0	0	0	0	**0.9679**	0	0.0077
C_8	0.0028	0.0083	0.0070	0.0097	0	0.0181	0.0167	0.0946	**0.8234**	0.0195
C_9	0.0153	0.0209	0	0.0111	0.0376	0.0542	0	0.0014	0	**0.8595**

Table 35: *k-NN classifier confusion matrix with the 7494 digit database for testing and the 3498 digit database for training - Handwritten digit recognition*

6.5 Conclusion

The use of multiple classifiers has not been advantageous for the recognition of handwritten numerals with the database made by Alimoglu and Alpaydin [28]. The k-NN classifier performed very well in that particular application, with a recognition rate over 97%. The combination of the k-NN classifier with weaker classifiers (Bayes and neural network) has reduced the performance whatever the combination method. However, the MLB multiple classifier still has the best recognition rate over all other multiple classifiers tested.

Output / Input	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_0	**0.9769**	0.0013	0.0013	0	0.0051	0	0.0077	0.0013	0.0051	0.0013
C_1	0.0026	**0.9371**	0.0141	0.0308	0.0051	0.0013	0.0039	0.0026	0	0.0026
C_2	0	0.0077	**0.9923**	0	0	0	0	0	0	0
C_3	0	0	0.0042	**0.9875**	0	0.0070	0	0.0014	0	0
C_4	0	0.0038	0.0013	0	**0.9936**	0	0	0.0013	0	0
C_5	0.0014	0	0	0.0014	0	**0.9861**	0.0014	0	0.0056	0.0042
C_6	0	0	0	0	0.0028	0.0014	**0.9958**	0	0	0
C_7	0	0.0103	0.0051	0.0013	0	0	0	**0.9704**	0	0.0129
C_8	0.0111	0.0111	0.0181	0.0056	0	0.0070	0.0195	0.0668	**0.8331**	0.0278
C_9	0.0097	0.0070	0	0.0028	0.0195	0.0334	0	0	0	**0.9277**

Table 36: *NN classifier confusion matrix with the 7494 digit database for testing and the 3498 digit database for training - Handwritten digit recognition*

λ_t	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
0	0.9626	0.0374	0	0.9626
0.01	0.9697	0.0303	0.	0.9697
0.02	**0.9703**	**0.0297**	**0**	**0.9703**
0.03	0.9688	0.0312	0	0.9688
0.04	0.968	0.032	0	0.968
0.05	0.9688	0.0312	0	0.9688
0.07	0.9686	0.0314	0	0.9686
0.1	0.9668	0.032	0	0.9668
0.2	0.9668	0.032	0	0.9668
0.5	0.9674	0.0326	0	0.9674
0.7	0.9688	0.0312	0	0.9688
0.9	0.8816	0.0277	0.0906	0.9695
1	0	0	1	-

Table 37: *Abstract Level Based multiple classifier (E_{ALB}) results in terms of λ_t - Handwritten digit recognition*

Output / Input	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_0	**0.9504**	0	0	0	0	0	0.0138	0	0.0358	0
C_1	0	**0.9835**	0.0082	0.0027	0.0027	0	0	0.0027	0	0
C_2	0	0.0055	**0.9918**	0	0	0.0027	0	0	0	0
C_3	0	0.0030	0	**0.9911**	0	0	0	0	0	0.0060
C_4	0	0	0	0	**0.9808**	0.0137	0.0027	0	0	0.0027
C_5	0	0	0	0.0179	0	**0.9761**	0	0	0	0.0060
C_6	0	0	0	0	0	0	**1.0000**	0	0	0
C_7	0	0.0714	0.0027	0	0	0	0	**0.8846**	0.0110	0.0302
C_8	0.0030	0	0	0	0	0.0030	0	0	**0.9940**	0
C_9	0	0.0060	0	0.0149	0.0030	0.0089	0	0.0089	0.0030	**0.9554**

Table 38: *Abstract Level Based multiple classifier (E_{ALB}) confusion matrix, $\lambda_t = 0.02$ - Handwritten digit recognition*

Classifier	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
e_B	0.7464	0.2536	0	0.7464
e_{NN}	0.9686	0.0314	0	0.9686
e_{kNN}	0.9783	0.0217	0	0.9783
E_{BKS}	0.9651	0.0212	0.0137	**0.9786**
E_{XU}	0.9626	0.0263	0.0111	0.9734
E_{LU}	0.9577	0.0423	0	0.9577
E_{AP}	0.9663	0.0337	0	0.9663
E_{ALB}	0.9703	0.0297	0	0.9703
E_{MLB}	**0.9743**	0.0257	0	0.9753

Table 39: *Results summary - Handwritten digit recognition*

Input \ Output	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
C_0	**0.9449**	0	0	0	0	0	0.0165	0	0.0386	0
C_1	0	**0.9780**	0.0192	0	0.0027	0	0	0	0	0
C_2	0	0.0027	**0.9973**	0	0	0	0	0	0	0
C_3	0	0.0030	0	**0.9911**	0	0	0	0	0	0.0060
C_4	0	0	0	0	**0.9835**	0.0110	0.0027	0	0	0.0027
C_5	0	0	0	0.0149	0	**0.9761**	0	0	0	0.0090
C_6	0	0	0	0	0	0	**1.0000**	0	0	0
C_7	0	0.0604	0.0055	0.0055	0	0	0	**0.8984**	0.0027	0.0275
C_8	0	0	0	0	0	0.0030	0	0	**0.9970**	0
C_9	0	0.0060	0	0	0	0	0	0.0089	0.0030	**0.9821**

Table 40: *Measure Level Based multiple classifier (E_{MLB}) confusion matrix, λ_t=0.01, k=100, $c = 1.3$ and $\lambda_{REJ} = 1$ - Handwritten digit recognition*

λ_{REJ}	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
0.2	0.5509	0.0046	0.4445	0.9918
0.3	0.8416	0.0089	0.1495	0.9896
0.4	0.9385	0.0114	0.05	0.988
0.5	0.9571	0.014	0.0289	0.9856
0.6	0.9651	0.0186	0.0163	0.9811
0.7	0.9697	0.0206	0.0097	0.9792
0.8	0.9737	0.0249	0.0014	0.9751
0.9	0.9743	0.0257	0	0.9743
1	0.9743	0.0257	0	0.9743

Table 41: *Results for Measure Level Based multiple classifier in terms of λ_{REJ}, with $\lambda_t = 0.01$, k=100 and $c = 1.3$ - Handwritten digit recognition*

7 Application to Satellite Image Classification

Satellite scene observation is an area in which classification methods can be needed. In applications like earth observing systems, images collected by satellites sometimes require to be classified among different categories. Here the simple classifiers presented in Chapter 2 and the multiple classifiers presented in Chapters 3 and 4 are tested using data originating from NASA's Landsat satellite (Landsat Multi-Spectral Scanner). The data consists of a small section (82x100 pixels) of a whole scene data purchased by the Australian Center for Remote Sensing [29]. Classifications are made among six classes of terrain and according to 36 features. Unless noted otherwise, tests were made using the same classifier configurations as in Chapter 6.

7.1 Database

The Landsat database is composed of the multi-spectral values of pixels in 3x3 neighborhoods and a class is associated to the central pixel [29]. Each pixel in the neighborhood is represented by 4 spectral values in different spectral bands (2 in the visible region and 2 in the (near) infra-red). The values range from 0 (black) to 255 (white). There are then 36 features per central pixel (3x3x4=36). Table 42 presents the six classes of terrain remotely measured by Landsat. A training and a testing databases of respectively 4435 and 2000 samples were created.

Class	Class of terrain
1	Red soil
2	Cotton crop
3	Grey soil
4	Damp grey soil
5	Soil with vegetation stubble
6	Very damp grey soil

Table 42: *Class numbers and types of terrains*

Tables 43 and 44 show the proportion of each class in the training and testing databases. The proportion of each class vary considerably and range from 9% to 24%, but are about the same in the training and testing sets. For example, there are more than twice class 1 samples than class 2 samples in the training and testing databases.

Class	1	2	3	4	5	6
Number	1072	479	961	415	470	1038
Proportion (%)	24.17	10.8	21.67	9.36	10.6	23.4

Table 43: *Proportion of each class in the training database (4435 samples)*

Class	1	2	3	4	5	6
Number	461	224	397	211	237	470
Proportion (%)	23.05	11.2	19.85	10.55	11.85	23.5

Table 44: *Proportion of each class in the testing database (2000 samples)*

7.2 Simple Classifiers

Just like with infrared images recognition and handwritten digit recognition presented in Chapters 5 and 6, respectively, k-NN classifier and neural network classifier have similar recognition rates, with 90.65% and 89.15%, respectively (Table 45). The performance of the Bayes classifier does not change. It remains low with a recognition rate of 74.55%.

Classifier	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
e_B	0.7455	0.2545	0	0.7455
e_{NN}	0.8915	0.1085	0	0.8915
e_{kNN}	0.9065	0.0935	0	0.9065

Table 45: *Individual classifier results - Satellite image classification*

Table 46 shows the confusion matrix for the Bayes classifier. The matrix has been obtained with the following parameters : $P(I = i) = 1/M$, $i \in \Lambda$; *a priori* distributions have been calculated with 40 intervals ($Q = 40$). The training database was used (4435 images) to calculate the *a priori* distributions.

Input \ Ouput	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**0.5076**	0.0065	0.2104	0.1518	0.0998	0.0239
C_2	0	**0.9241**	0.0089	0.0402	0.0179	0.0089
C_3	0	0	**0.9244**	0.0655	0.0050	0.0050
C_4	0	0	0.1185	**0.7393**	0	0.1422
C_5	0.0844	0.0127	0.0084	0.0633	**0.7300**	0.1013
C_6	0	0	0.0213	0.2128	0.0128	**0.7532**

Table 46: *Bayes classifier (e_B) confusion matrix - Satellite image classification*

The neural network has been configured with two layers, of respectively 50 and 30 neurons. The confusion matrix shown in Table 47 as been obtained according to the same configuration parameters as the one in Chapters 5 and 6.

Table 48 shows the confusion matrix for the k-NN classifier, where the distances have been calculated according to (8) and (9). The value of k has been fixed to 3. Results for the k-NN classifier are presented in Table 45.

Clearly, weaknesses can be observed for class 4 with both NN and k-NN, as well as for class 1 in the Bayes classifier.

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**0.9892**	0	0.0043	0	0.0065	0
C_2	0	**0.9509**	0	0	0.0446	0.0045
C_3	0.0101	0	**0.9496**	0.0126	0.0025	0.0252
C_4	0	0	0.1848	**0.5355**	0.0142	0.2654
C_5	0.0253	0.0169	0.0127	0.0042	**0.8819**	0.0591
C_6	0.0021	0	0.0383	0.0404	0.0362	**0.8830**

Table 47: *Neural Network classifier (e_{NN}) confusion matrix - Satellite image classification*

7.3 Existing Multiple Classifiers

BKS (Subsection 3.3.1) classifier, Lu (Subsection 3.3.2), Xu (Subsection 3.3.3) and Appriou's model (Subsection 3.3.4) multiple classifiers were tested on the 2000 image testing database, where Bayes, k-NN and neural network classifiers were combined.

None of the methods was able to improve over k-NN. In fact they all have recognition rates below the one of the k-NN, as shown in Table 49. Class 4 seems to be a problem, as the recognition rate for class 4 is below 69% in all of the four methods. The reason for this must be that NN and k-NN classifiers have low rates with class 4 (53.55% and 67.77%, respectively).

7.3.1 Behavior Knowledge Space Multiple Classifier

BKS multiple classifier was tested with $\lambda = 0$. This classifier has the lowest reliability and recognition rates (87.84% and 87.4%). Its confusion matrix is presented in Table 50. Note that 10 samples had decisions returning more than one class. These samples were rejected (ignored).

7.3.2 Lu's Multiple Classifier

Lu's multiple classifier confusion matrix is shown in Table 51. The recognition and reliability rates are both 89.7%. Cf values have been calculated using the probabilities defined in (3), (11) and (15), and with $k = 15$ (11). 15 intervals gives the best results (recognition rate of 89.7% compared to 89.15% with both 10 and 20 intervals).

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**0.9913**	0	0.0043	0.0022	0.0022	0
C_2	0.0045	**0.9643**	0	0.0045	0.0179	0.0089
C_3	0.0076	0.0025	**0.9270**	0.0453	0.0025	0.0151
C_4	0	0.0095	0.1327	**0.6777**	0.0047	0.1754
C_5	0.0084	0.0084	0.0042	0.0127	**0.8945**	0.0717
C_6	0	0	0.0255	0.0702	0.0170	**0.8872**

Table 48: *k-Nearest Neighbor classifier (e_{kNN}) confusion matrix - Satellite image classification*

Classifier	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
E_{BKS}	0.874	0.121	0.005	0.8784
E_{XU}	0.8755	0.1105	0.014	0.8879
E_{LU}	0.897	0.103	0	0.897
E_{AP}	0.903	0.097	0	0.903

Table 49: *Multiple classifier results - Satellite image classification*

7.3.3 Multiple Classifier with Appriou's Model of Initial Mass Assignment

The probability distributions have been made using the probabilities defined in (3), (11) and (15), and with $k = 15$ in (11). The confusion matrix for Appriou's model is presented in Table 52 and was obtained with the *a priori* probability distributions $p(m_j|C_i)$ represented by 15 intervals histograms. 15 intervals gives the best results (recognition rate of 90.3% compared to 89.95% and 85.1% with 10 and 20 intervals respectively). Confidence factors d_{ij} have been set to 1 for all simple classifiers. $R_j = 1$ for all combined classifiers j.

7.3.4 Xu's Multiple Classifier

Xu's multiple classifier gives a recognition rate of 87.55% and a reliability rate of 88.79%. Again, that is below the performance of the k-NN classifier. The initial BPAs have been created using the values in Table 53. Table 54 shows the results for $\lambda = 0$.

7.4 Proposed Multiple Classifiers

7.4.1 Abstract Level Based Multiple Classifier

The Abstract Level Based multiple classifier gives a lower recognition rate than Lu and Appriou's methods. A recognition rate of 88.2% is obtained with λ_t=0. Credibility rates were calculated using the confusion matrices in Tables 55, 57 and 56, that were created by inverting the testing and training databases (the 4435 samples database was used for testing and the 2000 samples database was used for training). Table 58 shows the results in terms of λ_t and Table 59 presents the confusion matrix for $\lambda_t = 0$. As the confusion matrix shows, the ALB multiple classifier was not able to correct the low performance obtained by NN and k-NN for class 4.

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**0.9935**	0.0022	0.0022	0.0022	0	0
C_2	0	**0.9545**	0.0045	0.0045	0.0273	0.0091
C_3	0.0101	0	**0.9394**	0.0278	0.0025	0.0202
C_4	0	0	0.1762	**0.5333**	0.0286	0.2619
C_5	0.1068	0.0128	0	0.0385	**0.7436**	0.0983
C_6	0.0043	0	0.0404	0.0383	0.0170	**0.9000**

Table 50: *BKS multiple classifier (E_{BKS}) confusion matrix , $\lambda = 0$ - Satellite image classification*

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**0.9892**	0	0.0065	0	0.0043	0
C_2	0	**0.9643**	0	0	0.0313	0.0045
C_3	0.0076	0	**0.9572**	0.0227	0.0025	0.0101
C_4	0	0.0047	0.2038	**0.5592**	0.0095	0.2227
C_5	0.0295	0.0253	0.0169	0.0169	**0.8270**	0.0844
C_6	0	0	0.0383	0.0553	0.0191	**0.8872**

Table 51: *Lu's classifier (E_{LU}) confidence matrix with interpolated frequency histograms, 10 intervals - Satellite image classification*

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**0.9913**	0	0.0065	0	0.0022	0
C_2	0	**0.9554**	0.0089	0.0089	0.0179	0.0089
C_3	0.0050	0	**0.9521**	0.0302	0.0025	0.0101
C_4	0	0	0.1422	**0.6872**	0.0047	0.1659
C_5	0.0506	0.0127	0.0084	0.0127	**0.8228**	0.0928
C_6	0.0043	0	0.0319	0.0596	0.0170	**0.8872**

Table 52: *Appriou's model multiple classifier (E_{AP}) confusion matrix with frequency histograms, 10 intervals - Satellite image classification*

7.4.2 Measure Level Based Multiple Classifier

The Measure Level Based multiple classifier provides the best results for the Satellite image classifications. Its results are not very sensitive to the variations of the different configuration parameters. With λ_t=0.01, $k = 100$, $c = 1.1$ and $\lambda_{REJ} = 1$, the recognition rate is 91.3%. That is above all other classifiers or multiple classifiers tested in this work. Table 60 compares the results of all simple classifiers and multiple classifiers tested, while Table 61 shows the confusion matrix of the MLB multiple classifier for λ_t=0.01, $k = 100$, $c = 1.1$ and $\lambda_{REJ} = 1$. Note that there is still a problem with class 4, which is much confused with class 3 and 6. Table 62 shows the impact of λ_{REJ} on the results. See Section D for more details.

7.5 Conclusion

Except the Bayes classifier, all other classifiers have recognition rates between 87% and 92%, so that there are not considerable performance differences between each classifier for the recognition of Landsat images. Anyhow, the MLB multiple classifier is the only one to overpass k-NN, according to the recognition rate. The ALB multiple classifier does not perform very well for that application and the results in Sections 5 and 6 also show that the MLB multiple classifier is better than the ALB.

Classifier	ϵ_r	ϵ_s
e_B	0.7479	0.2521
e_{NN}	0.8911	0.1089
e_{kNN}	0.8886	0.1114

Table 53: *Individual classifier results using the 4435 digits database for testing and 2000 digits database for training - Satellite image classification*

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**0.9935**	0.0022	0.0043	0	0	0
C_2	0	**0.9635**	0.0046	0	0.0274	0.0046
C_3	0.0051	0	**0.9518**	0.0330	0.0025	0.0076
C_4	0	0	0.1683	**0.6442**	0.0144	0.1731
C_5	0.0796	0.0133	0	0.0088	**0.7920**	0.1062
C_6	0.0022	0	0.0366	0.0989	0.0129	**0.8495**

Table 54: *Xu's classifier confusion matrix (E_{XU}) - Satellite image classification*

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**0.5149**	0.0019	0.1996	0.1175	0.1371	0.0289
C_2	0	**0.9290**	0.0146	0.0251	0.0125	0.0188
C_3	0	0	**0.9407**	0.0552	0.0042	0
C_4	0.0024	0.0024	0.1687	**0.7157**	0.0024	0.1084
C_5	0.0745	0.0064	0.0149	0.0468	**0.7426**	0.1149
C_6	0	0	0.0193	0.2052	0.0337	**0.7418**

Table 55: *Bayes classifier confusion matrix with the 4435 samples database for testing and the 2000 samples database for training - Satellite image classification*

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**0.9739**	0.0028	0.0159	0	0.0065	0.0009
C_2	0	**0.9520**	0.0042	0.0125	0.0209	0.0104
C_3	0.0073	0.0010	**0.9043**	0.0635	0.0021	0.0219
C_4	0.0145	0.0072	0.1639	**0.6072**	0.0193	0.1880
C_5	0.0383	0.0106	0	0.0043	**0.8638**	0.0830
C_6	0	0.0010	0.0145	0.0829	0.0212	**0.8805**

Table 56: *k-NN classifier confusion matrix with the 4435 samples database for testing and the 2000 samples database for training - Satellite image classification*

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**0.9692**	0.0009	0.0168	0	0.0131	0
C_2	0.0063	**0.9541**	0.0021	0.0084	0.0230	0.0063
C_3	0.0031	0	**0.9386**	0.0312	0.0062	0.0208
C_4	0.0048	0.0072	0.2096	**0.5446**	0.0193	0.2145
C_5	0.0319	0.0106	0.0043	0.0043	**0.8532**	0.0957
C_6	0	0.0029	0.0193	0.0530	0.0318	**0.8931**

Table 57: *NN classifier confusion matrix with the 4435 samples database for testing and the 2000 samples database for training - Satellite image classification*

λ_t	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
0	**0.882**	**0.118**	**0**	**0.882**
0.01	0.881	0.119	0	0.881
0.02	0.8815	0.1185	0	0.8815
0.03	0.8815	0.1185	0	0.8815
0.1	0.881	0.119	0	0.881
0.2	0.88	0.12	0	0.88
0.7	0.486	0.0515	0.4625	0.9042
1	0	0	1	-

Table 58: *Abstract Level Based multiple classifier (E_{ALB}) results in terms of λ_t - Satellite image classification*

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**0.9935**	0.0022	0.0022	0	0.0022	0
C_2	0	**0.9554**	0.0045	0	0.0313	0.0089
C_3	0.0101	0	**0.9395**	0.0378	0.0025	0.0101
C_4	0	0	0.1659	**0.6493**	0.0142	0.1706
C_5	0.0759	0.0127	0	0.0211	**0.7806**	0.1097
C_6	0.0021	0	0.0362	0.1021	0.0149	**0.8447**

Table 59: *Abstract Level Based multiple classifier (E_{ALB}) confusion matrix, $\lambda_t = 0$ - Satellite image classification*

Classifier	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
e_B	0.7455	0.2545	0	0.7455
e_{NN}	0.8915	0.1085	0	0.8915
e_{kNN}	0.9065	0.0935	0	0.9065
E_{BKS}	0.874	0.121	0.005	0.8784
E_{XU}	0.8755	0.1105	0.014	0.8879
E_{LU}	0.897	0.103	0	0.903
E_{AP}	0.903	0.097	0	0.903
E_{ALB}	0.882	0.118	0	0.882
E_{MLB}	**0.913**	**0.087**	**0**	**0.913**

Table 60: *Results summary - Satellite image classification*

Input \ Output	C_1	C_2	C_3	C_4	C_5	C_6
C_1	**0.9935**	0	0.0043	0	0.0022	0
C_2	0	**0.9598**	0.0089	0	0.0268	0.0045
C_3	0.0076	0.0025	**0.9572**	0.0202	0.0025	0.0101
C_4	0	0.0047	0.1469	**0.6540**	0.0047	0.1896
C_5	0.0127	0.0127	0.0084	0.0042	**0.8987**	0.0633
C_6	0.0021	0	0.0340	0.0447	0.0213	**0.8979**

Table 61: *Measure Level Based multiple classifier (E_{MLB}) confusion matrix, λ_t=0.01, k=100, $c = 1.1$ and $\lambda_{REJ} = 1$ - Satellite image classification*

λ_{REJ}	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
0.2	0.4495	0.0285	0.522	0.9404
0.3	0.853	0.067	0.08	0.9272
0.4	0.881	0.069	0.05	0.9274
0.5	0.8915	0.0715	0.037	0.9258
0.6	0.9015	0.077	0.0215	0.9213
0.7	0.9095	0.0855	0.005	0.9141
0.8	0.911	0.085	0.0005	0.9115
0.9	0.911	0.089	0	0.911
1	0.911	0.089	0	0.911

Table 62: *Results for the Measure Level Based multiple classifier in terms of* λ_{REJ}*, with* $\lambda_t = 0.01$*, k=100 and* $c = 1.3$ *- Satellite image classification*

8 Conclusion

In most supervised classification problems, reliability, precision and flexibility are required all together. Thus, some popular classification problems, like handwritten character recognition for example, are so that the classification system must be robust and must be able to adapt to different changes without reducing performance. In this report, multiple classifiers have been presented as a mean to improve reliability, precision and flexibility in supervised classifications.

Interest in multiple classifiers comes from the fact that there is no classifier able to provide flawless performance under any condition and in any application, although some classifiers perform better than others in particular situations. The diversity in classifier performance makes it possible to have complementarities between different classifiers. The combination of several complementary simple classifiers can create a multiple classifier that can provide high performance under a wider range of conditions and situations. In order to satisfy the criteria of reliability, precision and flexibility using a multiple classifier, the information provided by the individual classifiers must be chosen properly. Most importantly, it must be valuated adequately by taking into account the uncertainty. In the existing multiple classifiers, the valuation of the available information is based primarily on *a priori* knowledge. The *a priori* knowledge is obtained from specific training procedure. Moreover, since each classifier has a distinct behavior, the *a priori* knowledge must be obtained in a comparative way with the other classifiers.

Four of the most popular multiple classifiers have been examined: BKS, Lu, Appriou (the method for assigning the BPAs), and Xu's multiple classifiers. Appriou's BPA assignment method was applied to the creation of multiple classifiers. The simple classifiers involved in the creation of the multiple classifiers are the Bayes, the k-NN and the neural network classifiers. The BKS multiple classifier uses a particular knowledge space to obtain some *a priori* knowledge related to the simple classifiers, Xu's multiple classifier handles performance rates while Lu and Appriou's methods use probability density distributions. The three latter, that are Appriou's model, Xu and Lu's methods, use Dempster-Shafer's theory in the representation and combination of the information.

Dempster-Shafer's theory was also used in the suggested multiple classifiers of this report: the Abstract Level Based (ALB) multiple classifier and the Measure Level Based (MLB) multiple classifier. The ALB uses an information level, the abstract one, that is an abstraction of the measure level. The measure level is used in the MLB multiple classifier.

The Measure Level Based multiple classifier benefits from the fact that it does not need any *a priori* knowledge, so that no training phase is required besides those needed with the simple classifiers. Also, although some parameters must be adjusted, the results are not sensible to the variations of the configuration parameters. The reason for this low sensibility to parameter variations is that each individual parameter has a relatively low influence in the BPA combination.

The results generated with the IR ship images database, the handwritten digits database and the Landsat satellite images database suggest that the combination method that operates on the measure level is the one that qualifies best each individual classifier. It seems better to consider the information supplied by the measures rather than using the information supplied by the final output, even if the latter is associated with *a priori* knowledge. The results also show that there may be situation where the final output represent a loss of information compared with the measures from which the final output is generated. Adding *a priori* probability distributions to support the measures, as it is with Lu and Appriou's methods, can affect negatively the recognition rate. This can be explained by the fact that the probability distributions are uncertain. The uncertainty on the probability distributions is added to the uncertainty on the measures, so that the overall uncertainty is higher at the end.

This work presented a study of combination methods and a study of the information supplied by the combined classifiers. The different approaches that were analyzed differ in the information provided by each algorithm, the supporting knowledge, the information and knowledge representations, the combination method and the decision. The emphasis is principally on the issue of choosing and managing the information and its supporting knowledge. This issue is still a concern in the making of multiple classifiers, where various theory and methodologies can be accounted. The concern for that specific issue is augmented by the fact that the quest for better classification systems inevitably goes through multiple classifiers when flexibility and robustness are an issue. Until now, because no method exists for classifying any kind of data in any situations, only the combination of different methods allows getting closer to that goal.

References

[1] Wang, Lin-Cheng, Der, Sandor Z., and Nasrabadi, Nasser M. (1998). Composite classifier for automatic target recognition. *Optical Engineering*, **37**(3), 858–868.

[2] Chibelushi, C.C., Deravi, F., and Mason, J.S. (28-30 avril 1997). Audio-Visual Person Recognition : An Evaluation Of Data Fusion Strategies. In *European Conference on Security and Detection*.

[3] Tremblay, C. and Valin, P. (July 14-18, 2002). Experiments on Fusion of Individuals Classifiers and a Set of Classifiers. In *Accepted for 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002), Orlando, USA*.

[4] Ho, T.K., J.Hull, and S.N.Srihari (1994). Decision Combination in Multiple Classifier Systems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **16**(1), 66–75.

[5] Kang, Hee-Joong and Lee, Seong-Whan (2000). An Information-Theoretic Strategy for Constructing Multiple Classifier Systems. In *Proceedings 15th International Conference on Pattern Recognition*, Vol. 2.

[6] Xu, Lei, Kryzak, Adam, and Suen, Ching Y. (1992). Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition. *IEEE Transactions on Sytems, Man, and Cybernetics*, **22**(3).

[7] Lu, Yi (1996). Knowledge Integration in a Multiple Classifier System. *Applied Intelligence*, **6**, 75–86.

[8] Huang, Y.S. and Suen, C.Y. (1995). A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals. *IEEE Trans. on Pattern Recognition and Artificial Intelligence*, **17**(1), 90–94.

[9] Kittler, Josef, Hatef, Mohamad, Duin, Robert P. W., and Matas, Jiri (1998). On Combining Classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(3), 226–239.

[10] Binaghi, E. and Madella, P. (1999). Fuzzy Dempster-Shafer reasoning for rule-based classifiers. *International Journal of Intelligent Systems*, **14**(6), 559–583.

[11] Dietterich, Thomas G. (2000). Ensemble Methods in Machine Learning. In Kittler, J. and Roli, F., (Eds.), *Multiple Classifier Systems*, Springer-Verlag, Berlin.

[12] Impedovo, S. and Salvo, A. (2000). A New Evaluation Method for Expert Combination in Multi-expert System Designing. In Kittler, J. and Roli, F., (Eds.), *Multiple Classifier Systems*, Springer-Verlag, Berlin.

[13] Dezert, J. (2001). Optimal Bayesian Fusion of Multiple Unreliable Classifiers. In *Fusion 2001*.

[14] Denoeux, T. (1995). A k-Nearest Neighbor Classification Rule Based on Dempster-Shafer Theory. *IEEE Transactions on Systems, Man and Cybernetics*, **25**(5), 804–813.

[15] Mandler, E. and Schurmann, J. (1988). Combining the Classification Results of Independant Classifiers Based on the Dempster/Shafer Theory of Evidence. In *Pattern Recognition and Artificial Intelligence*, pp. 381–393.

[16] Appriou, A. (1991). Probabilités et incertitude en fusion de données multi-senseurs. *Revue Scientifique et Technique de la Défense*, **11**, 27–40.

[17] Baker, E. (1995). Computer-Assisted Reasoning in Cluster Analysis, Prentice Hall.

[18] Rhéaume, F., Jousselme, A.-L., and Grenier, D. (2000). Techniques de classification supervisées appliquées la reconnaissance de bateaux partir d'images FLIR. Laboratoire de Radiocommunication et de Traitement du Signal, Université Laval. (Technical Report LRTS-RT-00-01).

[19] Ho, T.K. (2000). Complexity of Classification Problems and Comparative Advantages of Combined Classifiers. In Kittler, J. and Roli, F., (Eds.), *Multiple Classifier Systems*, pp. 97–106. Springer-Verlag, Berlin.

[20] Dempster, A. (1967). Upper and Lower Probablilities Induced by Multivalued Mapping. *Ann. Math. Statist.*, **38**, 325–339.

[21] Shafer, G. (1976). A Mathematical Theory of Evidence, Princeton University Press.

[22] Smets, P. (1990). Constructing the pignistic probability function in a context of uncertainty. *Uncertainty in Artificial Intelligence*, **5**, 29–39. Elsevier Science Publishers.

[23] Park, Y. (1994). A Comparison of neural net classifiers and linear tree classifiers: their similarities and differences. *Elsevier Science Ltd*, **27**(11), 1493–1503.

[24] Park, Y. and Sklansky, J. (1990). Automated Design of Linear Tree Classifiers. *Pattern Recognition*, **23**(12), 1393–1412.

[25] Hu, M.K. (1962). Visual Pattern Recognition by moment invariant. *Visual Pattern Recognition by moment invariant*, **IT-8**, 179–187.

[26] Lehoux, N. and Ménard, E. (1998). Techniques et stratégies utilisées pour la reconnaissance des types de bateaux marchands. Laboratoire de Radiocommunication et de Traitement du Signal, Université Laval. Technical Report.

[27] Ménard, É. (1999). Rapport de Projet - Cours GEL-19464. Dpt. Génie Électrique - Université Laval. Technical Report.

[28] Alimoglu, F. and Alpaydin, E. (1996). Methods of Combining Multiple Classifiers Based on Different Representations for Pen-based Handwriting Recognition. *Proceedings of the Fifth Turkish Artificial Intelligence and Artificial Neural Networks Symposium (TAINN 96).*

[29] http://www.liacc.up.pt/ML/statlog/datasets/satimage/satimage.doc.html.

This page intentionally left blank.

Annex A: Infrared Images Recognition Tables - Measure Level Based Multiple Classifier

λ_t	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
0.005	0.9816	0.0184	0	0.9816
0.01	0.9824	0.0176	0	0.9824
0.02	0.9808	0.0192	0	0.9808
0.03	0.9784	0.0216	0	0.9784
0.05	0.976	0.0216	0.0024	0.9783
0.08	0.9752	0.0224	0.0024	0.9775
0.15	0.9752	0.02	0.0048	0.9799
0.3	0.968	0.0216	0.0104	0.9782
0.5	0.9204	0.024	0.052	0.9745

Table A.1: *Measure Level Based multiple classifier results in terms of λ_t, with $k = 15$, c=1.3 and $K_{REJ} = 1$*

k	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
1	0.9608	0.0392	0	0.9608
3	0.9664	0.0336	0	0.9664
5	0.9696	0.0304	0	0.9696
10	0.9784	0.0216	0	0.9784
15	0.9824	0.0176	0	0.9824
20	0.9808	0.0192	0	0.9808
30	0.9768	0.0232	0	0.9768
40	0.98	0.02	0	0.98
50	0.98	0.02	0	0.98

Table A.2: *Measure Level Based multiple classifier results in terms of k, with λ_t=0.01, c=1.3 and $K_{REJ} = 1$*

c	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
1	0.9808	0.0192	0	0.9808
1.1	0.9808	0.0192	0	0.9808
1.2	0.9816	0.0184	0	0.9816
1.3	0.9824	0.0176	0	0.9824
1.4	0.9808	0.0192	0	0.9808
1.5	0.9816	0.0184	0	0.9816
1.7	0.98	0.02	0	0.98
2	0.9808	0.0192	0	0.9808
3	0.98	0.02	0	0.98
5	0.9752	0.0248	0	0.9752
10	0.9752	0.0248	0	0.9752

Table A.3: *Measure Level Based multiple classifier results in terms of* c, *with* λ_t=*0.01*, $k = 15$ and $K_{REJ} = 1$

Annex B: Misclassified Ships Features Values - Measure Level Based Multiple Classifier - Infrared Images Recognition

Λ	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
1	874.6	0.915	0.05585	0.006803	-0.000052	-0.004201	-0.000122	1.077	-0.1788	-0.03977	0.7274
1	1091	0.8941	0.06695	0.007954	-0.000066	-0.004677	-0.000171	0.8207	0.149	-0.08843	0.6017
1	1059	0.9142	0.1334	0.02972	0.001252	0.005677	-0.001391	1.222	-0.5427	0.1764	0.5246
1	453.1	0.869	0.0438	0.003424	-0.000042	-0.00315	-0.000006	1.038	-0.2113	-0.1547	1.096
1	891.9	0.8748	0.08274	0.01058	-0.000043	-0.004434	-0.00031	1.309	-0.5439	0.1265	0.6346
1	375.8	0.6589	0.3439	0.02673	-0.002557	-0.02169	-0.000167	1.023	-0.01972	-0.09335	0.6274
3	613.2	0.9299	0.04677	0.03721	0.001552	0.03574	-0.000034	0.9487	0.2137	-0.4555	2.133
3	621.8	0.9179	0.0451	0.02769	0.000972	0.02517	-0.000117	0.9211	-0.07824	-0.2193	2.600
3	787.4	0.871	0.08707	0.04642	0.002912	0.04163	-0.000483	1.040	-0.0811	-0.3728	2.481
4	667.8	0.9345	0.4263	0.3772	0.1513	0.3629	-0.000767	1.033	-0.286	0.05511	1.064
5	1159	0.8755	0.007453	0.006328	0.000043	0.0059	0.000002	0.8801	0.06279	-0.2243	1.272
5	752.3	0.9678	0.2287	0.2184	0.04881	0.2137	0.00004	0.8273	-0.01743	-0.1997	1.332
6	1674	0.9091	0.02409	0.004293	0.000024	0.000104	-0.000037	1.082	-0.5869	0.3399	0.5803
6	1652	0.9082	0.02544	0.003021	-0.000004	-0.001443	-0.000026	0.9001	-0.3513	0.2349	0.6136
6	1733	0.913	0.02542	0.003406	0.000003	-0.001086	-0.000032	0.9213	-0.4631	0.2819	0.6686
6	415.5	0.4075	0.337	0.0139	-0.000456	-0.005507	0.000835	1.751	-1.060	0.2325	0.7312
6	119	0.8918	0.05245	0.006904	0.000022	-0.001619	-0.000129	0.9985	-0.5507	0.3427	0.6637
6	978.1	0.8888	0.1111	0.03885	0.00244	0.02408	-0.000748	0.9533	-0.4715	0.2953	0.6456
6	1448	0.916	0.03414	0.00448	0.000002	-0.001672	-0.000055	0.9316	-0.4543	0.2847	0.6316
7	1078	0.8614	0.1348	0.07705	0.00783	0.06441	0.000594	0.9834	0.1223	-0.2634	0.7684
7	1163	0.8716	0.1438	0.09015	0.01026	0.07933	0.000373	1.210	-0.3831	0.03763	0.6812

Table B.1: *Misclassified ships with the Measure Level Based multiple classifier.*

This page intentionally left blank.

Annex C: Recognition of Handwritten Digits Tables - Measure Level Based Multiple Classifier

λ_t	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
0	0.974	0.026	0	0.974
0.02	0.9728	0.0266	0.0006	0.9734
0.05	0.9726	0.0252	0.0023	0.9748
0.1	0.9734	0.0229	0.0037	0.977
0.3	0.9706	0.0194	0.01	0.9804
0.5	0.96	0.0109	0.0291	0.9888

Table C.1: *Measure Level Based multiple classifier results in terms of λ_t, with $k = 100$, c=1.3 and $K_{REJ} = 1$*

k	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
50	0.9731	0.0269	0	0.9731
75	0.9737	0.0263	0	0.9737
100	0.9743	0.0257	0	0.9743
200	0.9743	0.0257	0	0.9743
2000	0.9743	0.0257	0	0.9743

Table C.2: *Measure Level Based multiple classifier results in terms of k, with λ_t=0.01, c=1.3 and $K_{REJ} = 1$*

c	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
1.1	0.974	0.026	0	0.974
1.2	0.9743	0.0257	0	0.9743
1.5	0.9734	0.0266	0	0.9734
2	0.9706	0.0294	0	0.9706
5	0.9686	0.0314	0	0.9686

Table C.3: *Measure Level Based multiple classifier results in terms of c, with λ_t=0.01, $k = 100$ and $K_{REJ} = 1$*

This page intentionally left blank.

Annex D: Satellite Image Classification Tables - Measure Level Based Multiple Classifier

λ_t	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
0	0.9105	0.0895	0	0.9105
0.01	0.911	0.089	0	0.911
0.02	0.9105	0.0895	0	0.9105
0.05	0.911	0.087	0.002	0.9128
0.1	0.906	0.0895	0.0045	0.9101
0.3	0.886	0.0685	0.0455	0.9282
0.5	0.858	0.0495	0.0925	0.9455

Table D.1: *Measure Level Based multiple classifier results in terms of λ_t, with $k = 100$, c=1.3 and $K_{REJ} = 1$*

k	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
5	0.9095	0.0905	0	0.9095
15	0.9105	0.0895	0	0.9105
50	0.9105	0.0895	0	0.9105
75	0.9105	0.0895	0	0.9105
100	0.911	0.089	0	0.911
200	0.9125	0.0875	0	0.9125
300	0.913	0.087	0	0.913

Table D.2: *Measure Level Based multiple classifier results in terms of k, with λ_t=0.01, c=1.3 and $K_{REJ} = 1$*

c	ϵ_r	ϵ_s	ϵ_{rej}	ϵ_f
1.1	0.913	0.087	0	0.913
1.2	0.912	0.088	0	0.912
1.5	0.9105	0.0895	0	0.9105
2	0.9055	0.0945	0	0.9055
5	0.8995	0.1005	0	0.8995

Table D.3: *Measure Level Based multiple classifier results in terms of c, with λ_t=0.01, $k = 100$ and $K_{REJ} = 1$*

Distribution list

This page intentionally left blank.

Distribution list

DRDC Valcartier TR 2003-318

Internal distribution

1	Director General
3	Document Library
1	Head/Decision Support Systems (author)
1	Dr R. Breton
1	Dr P. Valin
1	Dr A. Guitouni
1	Head/Information and Knowledge Management
1	R/Visualization & Geo-spatial Systems
1	R/Knowledge Management Systems
1	R/Informations System Warfare
1	Head/System of Systems
1	R/Metrics & Experimentation
1	R/Distributed Synthetic Environment
1	R/System Engineering & Architecture
1	LtCol. B. Carrier
1	LCdr É. Tremblay
1	Maj. B. Deschênes
1	Dr M. Allouche
1	M. Bélanger
1	Dr A. Benaskeur
1	J. Berger
1	M. Blanchette
1	Dr A. Boukhtouta
1	Dr A.-C. Bourry-Brisset

1 C. Daigle

1 Dr H. Irandoust

1 Dr A.-L. Jousselme (author)

1 P. Maupin

1 S. Paradis

1 F. Rhéaume (author)

1 A. Sahi

1 J.M.J. Roy

1 A. Jouan

Total internal copies: 35

External distribution

1 M. C. Florea
Informatique WGZ inc.

1 DRD KIM (PDF file)

2 Director Science & Technology Maritime(DSTM)

2 Director Science & Technology Land (DSTL)

2 Director Science & Technology Air (DSTA)

2 Director Science & Technology C4ISR (DSTC4ISR)

1 Director Maritime Requirements Sea (DMRS) 4

1 Director Maritime Requirements Sea (DMRS) 6

1 Director Aerospace Requirements (DAR) 4

1 Director Aerospace Requirements (DAR) 4-2

2 Director Maritime Ship Support (DMSS) 6

2 Director Maritime Ship Support (DMSS) 8

2 DRDC - Atlantic:
Attn: Dr. Bruce MacArthur
Attn: Dr. Jim S. Kennedy

2 DRDC - Ottawa:
Attn: Barbara Ford
Attn: Dan Brookes

2 CF Maritime Warfare School CFB Halifax
Halifax, Nova Scotia
Attn: TAC AAW
OIC Modeling and Simulation

2 Canadian Forces Naval Operations School CFB Halifax
Halifax, Nova Scotia
Attn: Tactic
CT AWW

1 Canadian Forces Naval Engineering School CFB Halifax
Halifax, Nova Scotia
Attn: CSTC

1 Operational Requirements Analysis Cell CFB Halifax
Halifax, Nova Scotia
Attn: Commanding Officer

1 Canadian Forces Fleet School CFB Esquimalt
Esquimalt, British Columbia
Attn: Commanding Officer/WTD

1 Operational Requirements Analysis Cell CFB Esquimalt
Esquimalt, British Columbia
Attn: Commanding Officer

Total external copies: 30

Total copies: 65

FICHE DE CONTRÔLE DU DOCUMENT

1. PROVENANCE (le nom et l'adresse) **Francois Rheaume** **RDDC - Valcartier** **2459, boul. Pie-Xi Nord** **Québec Qc, G3J 1X5 Canada**	2. COTE DE SÉCURITÉ (y compris les notices d'avertissement, s'il y a lieu) UNCLASSIFIED

3. TITRE (Indiquer la cote de sécurité au moyen de l'abréviation (S, C, R ou U) mise entre parenthèses, immédiatement après le titre.)

Fusion of supervised classifier using theory of evidence (U)

4. AUTEURS (Nom de famille, prénom et initiales. Indiquer les grades militaires, ex.: Bleau, Maj. Louis E.)

Francois Rhéaume, Anne-Laure Jousselme et Éloi Bossé

5. DATE DE PUBLICATION DU DOCUMENT (mois et année) **2006**	6a. NOMBRE DE PAGES **90**	6b. NOMBRE DE REFERENCES **29**

7. DESCRIPTION DU DOCUMENT (La catégorie du document, par exemple rapport, note technique ou mémorandum. Indiquer les dates lorsque le rapport couvre une période définie.)

Rapport Technique

8. PARRAIN (le nom et l'adresse)

9a. NUMÉRO DU PROJET OU DE LA SUBVENTION (Spécifier si c'est un projet ou une subvention) **11ba18**	9b. NUMÉRO DE CONTRAT
10a. NUMÉRO DU DOCUMENT DE L'ORGANISME EXPÉDITEUR **TR 2003-318**	10b. AUTRES NUMÉROS DU DOCUMENT N/A

11. ACCÈS AU DOCUMENT (Toutes les restrictions concernant une diffusion plus ample du document, autres que celles inhérentes à la cote de sécurité.)

- ☐ Diffusion illimitée
- ☐ Diffusion limitée aux entrepreneurs des pays suivants (spécifier)
- ☐ Diffusion limitée aux entrepreneurs canadiens (avec une justification)
- ☐ Diffusion limitée aux organismes gouvernementaux (avec une justification)
- ☐ Diffusion limitée aux ministères de la Défense
- ☒ Autres

12. ANNONCE DU DOCUMENT (Toutes les restrictions à l'annonce bibliographique de ce document. Cela correspond, en principe, aux données d'accès au document (11). Lorsqu'une diffusion supplémentaire (à d'autres organismes que ceux précisés à la case 11) est possible, on pourra élargir le cercle de diffusion de l'annonce.)

13. SOMMAIRE (Un résumé clair et concis du document. Les renseignements peuvent aussi figurer ailleurs dans le document. Il est souhaitable que le sommaire des documents classifiés soit non classifié. Il faut inscrire au commencement de chaque paragraphe du sommaire la cote de sécurité applicable aux renseignements qui s'y trouvent, à moins que le document lui-même soit non classifié. Se servir des lettres suivantes: (S), (C), (R) ou (U). Il n'est pas nécessaire de fournir ici des sommaires dans les deux langues officielles à moins que le document soit bilingue.)

In the field of pattern recognition, more specifically in the area of supervised and feature-vector-based classifications, various classification methods exist, but none of them is flawless given any kind of data. Each classifier behaves differently, with its own strengths and weaknesses. Some are more efficient than others in particular situations. Performance of these individual classifiers can be improved by combining them into one multiple classifier. In order to make more realistic decisions, the multiple classifier can use some measures generated by each classifier and use some *a priori* knowledge, such as probability distributions, reliability rates and confusion matrices. Individual classifiers studied in this project are Bayes, *k*-nearest neighbors and neural network classifiers. They are combined using Dempster-Shafer's theory. The problem simplifies in finding weights that best represent individual classifier evidences. We suggest basic probability assignments (BPAs) based on measures preceding the decision step.

Following the study of some classical multiple classifiers in the literacy, we compare them with our approach that yielded in two distinct multiple classifiers. Tests are made on three different databases that are infrared images of ships, handwritten digits and satellite images of terrains. One of the suggested multiple classifier gives better results than all other classical multiple classifiers tested in this work.

14. MOTS-CLÉS, DESCRIPTEURS OU RENSEIGNEMENTS SPÉCIAUX (Expressions ou mots significatifs du point de vue technique, qui caractérisent un document et peuvent aider à le cataloguer. Il faut choisir des termes qui n'exigent pas de cote de sécurité. Des renseignements tels que le modèle de l'équipement, la marque de fabrique, le nom de code du projet militaire, la situation géographique, peuvent servir de mots-clés. Si possible, on doit choisir des mots-clés d'un thésaurus, par exemple le "Thesaurus of Engineering and Scientific Terms (TESTS)". Nommer ce thésaurus. Si l'on ne peut pas trouver de termes non classifiés, il faut indiquer la classification de chaque terme comme on le fait avec le titre.)

Supervised classification, multiple classifier, evidential theory, infrared image recognition, handwritten digit recognition, satellite image classification.

Defence R&D Canada

Canada's Leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE DÉFENSE

WWW.drdc-rddc.gc.ca