



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE
LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABDELHAMID IBN BADIS MOSTAGANEM
Faculté des Sciences Exactes & de l'Informatique
Département de Mathématiques et d'Informatique
Filière Informatique

MEMOIRE DE FIN D'ETUDES
Pour l'Obtention du Diplôme de Master en Informatique
Option : Ingénierie des Systèmes d'Information

Reconnaissance de panneaux routiers

Présenté par :

SAIDANI Nadjat

Encadré par:

Mr. BOUMEDIENE Mohammed

Année Universitaire 2015/ 2016

Le transport routier est le mode de transport le plus utilisé générant ainsi un trafic important qui ne cesse d'augmenter notamment dans les pays émergents. Malheureusement, cette augmentation du trafic routier a des conséquences sur la santé publique. Cependant, d'après les experts en accidentologies, le facteur humain représente la première cause dans 90% des accidents. C'est pourquoi les industriels automobiles intègrent de plus en plus de systèmes d'aide à la conduite lors de la conception de nouveaux véhicules.

Les systèmes avancés d'aide à la conduite (*Advanced Driver Assistance System - ADAS*) ont un grand intérêt dans la sécurité routière, ils assistent le conducteur en contrôlant son mode de conduite, et en le tenant informé des situations potentiellement dangereuses, ou en changeant son comportement afin d'éviter une collision ou limiter les conséquences d'un accident. Parmi ces systèmes il y a le système de reconnaissance de panneaux routiers (TSR).

Le système de reconnaissance de panneaux routiers (**TSR**) est un des systèmes avancés d'aide à la conduite. Ce système assiste le conducteur en le tenant informé de la signalisation présente sur la route parcourue. Un système TSR est usuellement basé sur une caméra embarquée sur l'avant du véhicule. L'objectif est d'analyser le flux vidéo fourni par la caméra et de reconnaître les panneaux présents. Pour cela, un TSR est composé de trois étapes : la détection, la reconnaissance (Classification), et le suivi temporel des panneaux reconnus.

Dans ce mémoire nous nous intéressons à l'étape de reconnaissance. Cette dernière identifie les panneaux détectés dans les images fournies par la caméra en exploitant leurs caractéristiques visuelles. La reconnaissance nécessite deux phases : l'extraction des caractéristiques qui va être faite en exploitant les caractéristiques de HOG (histogramme des gradients orientés), et la classification des descripteurs est basée sur les Séparateurs à vaste marges.

Mots clés

Système avancé d'aide à la conduite, système de reconnaissance de panneaux routiers, descripteurs visuels, classifieur, Histogramme des Gradients Orientés, Séparateur à Vaste Marge.

Road transport is the mode of transport most used thus generating significant traffic that is increasing particularly in emerging countries. Unfortunately, this increase in traffic affects public health.

Advanced Driver Assistance System (ADAS) have great interest in road safety, they support the driver by monitoring his driving mode, and taking it informed of potentially dangerous situations, or changing his behavior to avoid a collision or mitigate the consequences of an accident.

The road signs recognition (TSR) is one of the advanced driver assistance systems. This system assists the driver by holding informed of this signage on the route traveled. TSR system is usually based on a camera mounted on the front of the vehicle. The objective is to analyze the video stream provided by the camera and to recognize these signs. For this, a TSR has three stages: detection, recognition (Classification), and time tracking recognized panels.

In this paper we investigate the recognition step. The latter identifies the signs detected in the images provided by the camera by using their visual characteristics. Recognition requires two phases: the extraction of features that will be made by exploiting the characteristics of HOG, and classification of descriptors using a classifier based on SVM. The implementation of our application is done in C ++ using the OpenCV library.

Keywords

Advanced Driver Assistance System, Traffic Sign Recognition, visual descriptor, classifier, Histogram of Oriented Gradient, Support Vector Machine.

Résumé	i
Abstract	ii
Sommaire.....	iii
Liste des tableaux.....	v
Liste des figures.....	v
Liste des abréviations	vii
Remerciements.....	viii
Dédicace	ix
Introduction générale	1
Chapitre I : Système de reconnaissance de panneaux routiers(TSR)	2
I.1. Introduction.....	2
I.2. Système de reconnaissance de panneaux routiers (TSR).....	3
I.2. 1. La détection.....	4
I.2. 2. La reconnaissance	4
I.2. 3. Le suivi temporel.....	6
I.3 Objectif du projet	6
I.4 conclusion	6
Chapitre II : la reconnaissance de panneaux routiers (la classification).....	7
II .1. Introduction.....	7
II .2. Descripteurs de panneaux routiers.....	7
II .2.1 Le descripteur SIFT	7
II.2 .2. Histogramme de Gradients Orientés.....	8
II .3. Les différents types d'apprentissage :.....	12
II .3.1 Apprentissage supervisé	12
II .3.1 Apprentissage non-supervisé	13

II .3.1 Apprentissage semi-supervisé :.....	13
II .3.1 Apprentissage par renforcement :	13
II .4. Classifieur de panneaux routiers.....	14
II .4.1 Les réseaux de neurones	14
II .4.2 Les k plus proches voisins	15
II .4.3 Séparateurs à Vastes Marges (SVM).....	16
II .5. Conclusion	24
Chapitre 3 : Conception et implémentation.....	25
III.1. Introduction	25
III.2. Ressources utilisées	25
III.3. Pourquoi Visual Studio Express 2013.....	25
III.4. Pourquoi C++	25
III.5. Pourquoi OpenCV :	26
III.6. La base de données :.....	26
III.7. Diagramme de l'application	27
III.8. L'extraction des descripteurs.....	27
III.9. Classification des descripteurs	27
III.10. Résultats obtenus.....	28
III.10. Validation	28
III.11. Présentation de l'application	30
III.11.1. l'apprentissage.....	31
III.11.2. la classification	39
III.12. Conclusion.....	48
Conclusion générale.....	49
Bibliographie	50

Liste des tableaux

Tableau.1. Les paramètres de descripteur de HOG [11]	8
Tableau.2. Les paramètres de descripteur de HOG de notre système	27
Tableau.3. Le taux d'évaluation pour chaque configuration de HOG.	28
Tableau.4. Le CRR pour chaque configuration de HOG.	30

Liste des figures

Figure.1. Gravité des accidents de la route.....	2
Figure.2. Exemple de résultats fournis par un TSR. Un panneau de limitation de vitesse (110 Km/h) est reconnu.	3
Figure.3. Schéma général d'un système de reconnaissance de panneaux routiers [3].....	4
Figure 4. Les différentes catégories de panneaux [5].....	4
Figure 5. Schéma de l'étape de reconnaissance des panneaux [7].....	5
Figure 6. Les points d'intérêts liés aux panneaux [6]	5
Figure 7. Descripteur SIFT [8]	7
Figure 8. Le principe de décomposition pour le calcul des HOG [10]. L'image est divisée en blocs qui peuvent se recouvrir. Chaque bloc est subdivisé en cellules où un histogramme d'orientations est calculé.....	8
Figure 9. Image d'un piéton en infrarouge (gauche), image du gradient horizontal (milieu) et vertical (droite).....	9
Figure 10. Exemple d'image de piéton en image infrarouge (gauche), l'orientation du gradient associée à cette image (milieu) et la norme du gradient (droite).	10
Figure 11. Découpage d'une image infrarouge (gauche) en 12×6	10
Figure 12. Calcul des histogrammes définis pour 4 niveaux d'orientation (droite) pour chaque cellule (gauche).	11
Figure 13. Réseau de neurones à $l - 2$ couches cachées [19].....	14
Figure 14. Unité de traitement k d'un réseau de neurones. Les données reçues des cellules en amont y_j et le biais θ_k sont pondérés par les coefficients w_{jk} . Le niveau de sortie y_k est fonction de cette somme s_k et d'une fonction d'activation F_k [19].....	14
Figure.15. Principe de K-PPV [25].....	15
Figure 16. Principe de K-PPV [25]	17
Figure 17. Séparation linéaire des données de la base d'apprentissage par l'hyperplan $H : wx + b = 0$. Les vecteurs de support sont les points ayant une distance minimale avec H	18
Figure 18. Illustration de la construction d'un arbre K -d. le nœud A est séparé en 2 nœuds en utilisant la caractéristique f_1 , puis B et C sont séparés chacune en deux nœuds aussi en utilisant les caractéristiques f_2 et f_3 en $\{D,E\}$ et $\{F,G\}$ [4]	19
Figure 19. Classification d'un exemple en utilisant les forêts aléatoires [6]	23
Figure 20. Exemples des panneaux utilisés dans notre application	26
Figure 21. Diagramme de l'application.....	27
Figure 22. La matrice de confusion et le CRR obtenus avec la configuration HOG1	29
Figure 23. La matrice de confusion et le CRR obtenus avec la configuration HOG2	29
Figure 24. La matrice de confusion et le CRR obtenus avec la configuration HOG3	30
Figure 25. La première fonctionnalité de l'application : l'apprentissage.....	31
Figure 26. Le choix de la première configuration de HOG.....	32
Figure 27. L'affichage des paramètres de HOG1	33

Figure 28. Le choix des images de l'apprentissage.....	33
Figure 29. La sélection du répertoire des panneaux de 50 Km/h.....	34
Figure 30. La sélection du répertoire des panneaux de 60 Km/h.....	34
Figure 31. La sélection du répertoire des panneaux de 80 Km/h.....	35
Figure 32. La sélection du répertoire des panneaux de 100 Km/h.....	35
Figure 33. La sélection du répertoire des panneaux de 120 Km/h.....	36
Figure 34. Lecture des images, extraction et sauvegarde des descripteurs de HOG.....	36
Figure 35. Fenêtre d'information : SVM sauvegardé.....	37
Figure 36. Affichage du taux d'évaluation pour HOG1.....	37
Figure 37. Affichage du taux d'évaluation pour HOG2.....	38
Figure 38. Affichage du taux d'évaluation pour HOG3.....	39
Figure 39. La deuxième fonctionnalité de l'application : la classification	40
Figure 40. Choix de l'image du panneau à classifier	41
Figure 41. Un panneau de 120 Km/h reconnu	41
Figure 42. Un panneau de 100 Km/h reconnu	42
Figure 43. Un panneau de 50 Km/h reconnu	42
Figure 44. Un panneau de 60 Km/h reconnu	43
Figure 45. Un panneau de 80 Km/h reconnu	44
Figure 46. Exemple des panneaux non reconnu par notre système.....	46
Figure 47. Exemple d'un panneau de 120 Km/h non reconnu par notre système.....	47
Figure 48. Exemple d'un panneau inconnu, classifié comme panneau de 60 Km/h.....	47
Figure 49. Exemple d'un panneau inconnu, classifié comme panneau de 50 Km/h.....	47
Figure 50. Exemple d'un panneau inconnu, classifié comme panneau de 80 Km/h.....	48

Liste des abréviations

ADAS : Advanced Driver Assistance System, Système Avancé d'Aide à la Conduite

TSR : Traffic Sign Recognition, Reconnaissance de Panneaux Routiers

HOG : Histogram of Oriented Gradients, Histogramme de Gradients Orientés

GTSRB: German Traffic Sign Recognition Benchmark

SIFT: Scale Invariant Feature Transform

SVM : Support Vector Machine, Séparateur à Vaste Marge

K-D Tree: K-Dimensionnel Tree

K-PPV: K-Plus Proches Voisins, K-Nearest Neighbor(**KNN**)

CRR: Correct Rate Recognition, taux de bonne reconnaissance.

Remerciements

Tout d'abord, je tiens à remercier ALLAH le tout puissant et miséricordieux, qui m'a donné la force et la patience d'accomplir ce modeste travail.


Je souhaiterais exprimer ma gratitude à mon encadrant, *Mr BOUMEDIENE Mohammed* qui m'a prodigué ses conseils tout au long de ce travail. Il a fait preuve d'une grande écoute et de compréhension lors de nos discussions portant sur le projet. Son soutien et son enthousiasme m'ont beaucoup aidée à surmonter les difficultés inhérentes à tout travail.

Mon vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à mon mémoire en acceptant d'examiner mon travail et de l'enrichir par leur *propositions*.

Je garde une place particulière à mes parents, mon frère et mes sœurs qui sont toujours à mon cotés.

Enfin, j'adresse mes plus sincères remerciements à tous mes proches, mes amis, et mes collègues, qui m'ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire.

Dédicace

Je dédie ce travail à tous ceux que **j'aime** 

Introduction générale

Un conducteur peut être distrait de sa tâche principale, la conduite, ce qui occasionne un manque de vigilance vis à vis de la signalisation courante. Cette situation augmente le risque d'accident. En effet, manquer un panneau de limitation de vitesse ou d'interdiction de doubler génère une situation à risque en plus du fait d'exposer le conducteur à des sanctions. Afin de réduire ce risque, le système de reconnaissance de panneaux routiers (TSR-Traffic Signs Recognition) assiste le conducteur soit en le tenant informé de la signalisation liée à la route parcourue, soit de réguler la vitesse d'un véhicule autonome par exemple [1].

Le TSR est l'un des ADAS ayant obtenu, ces dernières années, le plus d'intérêt auprès des constructeurs automobiles. En effet, le TSR offre une perception pertinente de l'environnement afin de développer la conduite autonome. Un **véhicule autonome** est capable de rouler automatiquement et en toute autonomie dans le trafic réel et sur une infrastructure non spécifique sans l'intervention d'un être humain [2].

Le chapitre 1 nous présente le **TSR** et ses étapes principales: la Détection et la Reconnaissance (aussi appelée classification), et le suivi temporel. Ses deux premières étapes exploitent les caractéristiques des panneaux (la forme, la couleur, et le pictogramme) afin de les localiser et de les reconnaître dans les images fournies par la caméra.

Le chapitre 2 nous présente quelques descripteurs utilisés lors de la reconnaissance comme SIFT (Scale Invariant Feature Transform), HOG (Histogram of Oriented Gradients), pour l'extraction des caractéristiques des panneaux, ces caractéristiques sont ensuite classées en utilisant des classifieurs tels que les réseaux de neurones, SVM (Support Vector Machine), et les K-PPV (K-plus proches voisins) etc.

Le chapitre 3 présente notre application, reconnaissance des panneaux routiers, en spécifiant son diagramme de fonctionnement, et discute les résultats obtenus.

Chapitre I : Système de reconnaissance de panneaux routiers(TSR)

I.1. Introduction

D'après l'Organisation Mondiale de la Santé (OMS), les accidents de la route représentaient la huitième cause de mortalité au monde en 2010 et devraient être la cinquième cause d'ici 2030. Les conséquences des accidents sont néfastes sur le plan humain, économique, et sociétal. En effet, en plus des souffrances physiques et psychologiques des victimes (décès, blessures et handicaps), les accidents occasionnent des pertes économiques (Cf. **Figure 1**). Les blessures et les lésions causées nécessitent souvent des soins coûteux (séjour à l'hôpital, transport des victimes, mobilisation des pompiers, etc.) et parfois de longues convalescences ce qui entraîne une baisse de productivité chez les victimes ainsi que leurs proches. Face aux dangers de la route, une multitude de recherches ont été menées dans le domaine automobile afin de réduire les accidents ainsi que leurs conséquences. Pour cela, la sécurité routière est organisée en trois niveaux : primaire, secondaire, et tertiaire. La sécurité primaire consiste à éviter au maximum les accidents. Dans le cas où ces derniers n'ont pas pu être évités, la sécurité secondaire intervient en minimisant leurs conséquences. Pour améliorer la prise en charge des victimes, la sécurité tertiaire travaille sur le développement des moyens de secours.



Figure.1. Gravité des accidents de la route.

Beaucoup de pays ont réduit leur taux de mortalité sur les routes grâce à une politique de sécurité routière rigoureuse basée sur la sensibilisation et la répression. Cependant, avec tous les dispositifs mise en place le taux utopique de 0 morts n'est encore atteint dans aucun pays. Cela veut dire qu'une baisse significative des accidents est encore envisageable à travers le développement de systèmes d'aide à la conduite. En effet, 90% des accidents ont pour cause la défaillance humaine. C'est à partir de ce constat, qu'un grand nombre de chercheurs dans le

domaine des Systèmes de Transport intelligents (ITS - Intelligent Transportation Systems) travaillent sur l'assistance du conducteur afin de réduire le risque d'accident. C'est sur ce principe que les systèmes avancés d'aide à la conduite communément appelés "ADAS" (Advanced Driver Assistance System) sont développées. Ces derniers sont intégrés aux véhicules dits "intelligents" pour assister le conducteur durant sa tâche de conduite en lui assurant plus de sécurité et de confort. ils assistent le conducteur en contrôlant son mode de conduite, et en le tenant informé des situations potentiellement dangereuses, ou en changeant son comportement afin d'éviter une collision ou limiter les conséquences d'un accident [1].

Parmi ces systèmes il y'a le système de reconnaissance de panneaux routiers (*Traffic Sign Recognition - TSR*). Ce dispositif assiste le conducteur en le tenant informé de la signalisation liée à la route parcourue ce qui réduit le risque d'accident. En effet, Un conducteur peut être distrait de sa tâche principale, la conduite, ce qui occasionne un manque de vigilance vis à vis de la signalisation courante et augmente le risque d'accident. C'est pourquoi les TSR ont été développés, l'idée est de reconnaître les panneaux présents sur la route parcourue et d'en informer le conducteur. Dans ce projet de fin d'étude, nous étudions les systèmes TSR afin d'en développer un classifieur de panneaux routiers.

I.2. **Système de reconnaissance de panneaux routiers (TSR)**

Un **TSR** est usuellement basé sur une caméra embarquée sur l'avant du véhicule. Son principe consiste à analyser le flux vidéo fourni par la caméra afin de reconnaître les panneaux routiers présents dans la scène (Cf. **Figure 2**). Le système est basé sur deux principales étapes : la Détection et la Reconnaissance (aussi appelée classification). La première étape localise les panneaux dans les images en se basant sur leurs caractéristiques visuelles, comme la couleur et la forme. La deuxième étape reconnaît le type de chaque panneau détecté grâce à son pictogramme. En général, les panneaux reconnus sont signalés au conducteur ou transmis à un autre ADAS. Il est possible d'ajouter une étape de Suivi temporel afin de ne pas re-signaliser au conducteur les mêmes panneaux reconnus (Cf. **Figure 3**). En effet, les panneaux apparaissent sur plusieurs images avant leur disparition du champ de vision de la caméra ce qui permet leur suivi [3].



Figure.2. Exemple de résultats fournis par un TSR. Un panneau de limitation de vitesse (110 Km/h) est reconnu.

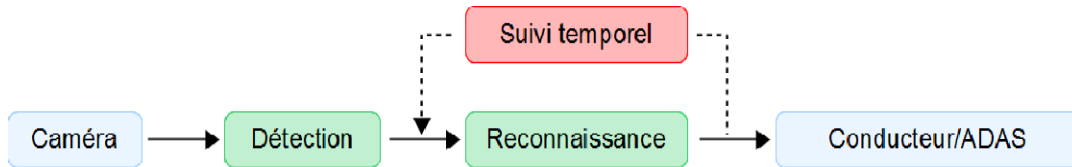


Figure.3. Schéma général d'un système de reconnaissance de panneaux routiers [3]

1.2. 1. La détection

Localise les panneaux dans les images. Cette étape exploite les caractéristiques visuelles des panneaux, comme la couleur et la forme [4], afin de les distinguer par rapport aux autres objets présents dans la scène. Cependant, en situations réelles la Détection n'est jamais parfaite. En plus des panneaux (Vrais Positifs) (VP), des fausses détections (Faux Positifs) (FP) peuvent être générées. En effet, la scène routière peut être assez encombrée et les panneaux peuvent être occultés ou très proches des autres objets, ce qui ne facilite pas la tâche de Détection et génère des FP.

La Détection doit prendre en considération différents facteurs comme la diversité des catégories de panneaux (panneaux d'interdiction, panneaux de danger, panneaux d'obligation, panneaux d'indication, panneaux de cédez le passage et d'intersection) (Cf. **Figure 4**) [5]; le changement d'éclairage est le faible contraste qui rendent la distinction difficile entre les panneaux et arrière-plan.



Figure 4. Les différentes catégories de panneaux [5]

1.2. 2. La reconnaissance

Après la Détection, il y a la Reconnaissance. Cette étape reconnaît le type de chaque panneau détecté grâce à son pictogramme en se basant sur un classifieur qui a été déjà entraîné sur les différentes classes des panneaux, donc il peut attribuer la classe appropriée aux panneaux en question. En général, les panneaux reconnus sont signalés au conducteur ou transmis à un autre ADAS [6].

La reconnaissance se déroule en deux phases: l'extraction des caractéristiques et leur classification (Cf. **Figure 5**).

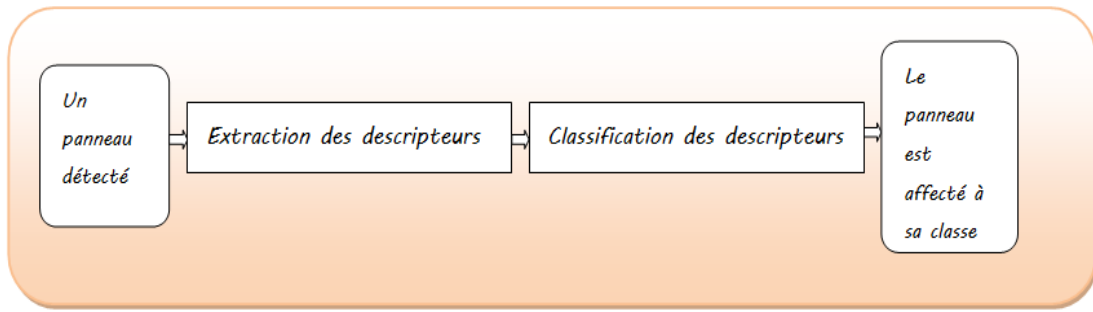


Figure 5. Schéma de l'étape de reconnaissance des panneaux [7]

1.2. 2.1 L'extraction des caractéristiques

Cette première phase consiste à extraire de chaque pictogramme (motif) un ensemble de caractéristiques propres à sa classe et le distinguant de façon nette des autres.

On distingue deux approches pour l'extraction des caractéristiques : l'approche local, et l'approche global.

Approche local :

Le motif peut être décrit sur de petites régions situées autour de points d'intérêt comme illustré dans la figure 6. Ces zones sont considérées comme riches en information et traçables sur différentes instances du même objet. Il faut prêter attention d'une part à la détection des points d'intérêt et d'autre part à une bonne représentation locale de l'image.

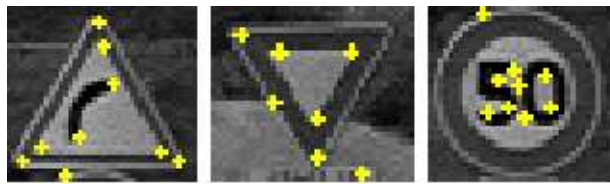


Figure 6. Les points d'intérêts liés aux panneaux [6]

La représentation de l'objet ainsi obtenue est compacte et ne concerne que les parties suffisamment discriminantes. Elle est également robuste aux occlusions car constituée d'un ensemble de petits descripteurs locaux. La majorité des descripteurs utilisés sont également rapides à calculer. En revanche, l'aspect spatial est perdu, la position relative des points n'est pas conservée et il faut pouvoir extraire de l'image des points d'intérêt, ce qui nécessite une bonne résolution de départ sur l'image.

Approche globale :

Lorsque l'ensemble de l'objet, image ou contour, est utilisé pour établir le descripteur, les caractéristiques sont dites globales. Le plus connu, et le plus facile à mettre en œuvre, est l'histogramme qui fournit la distribution de pixels par intervalles de valeurs.

Les approches globales visent à transformer l'image dans un autre espace de manière à en faire ressortir les caractéristiques. Leur avantage principal est qu'elles fournissent une représentation complète de l'image. Toutefois, elles ne sont pas souvent robustes aux occlusions et aux transformations affines.

1.2. 2.2 La classification des descripteurs

L'étape de détection a permis d'extraire de l'image les régions s'approchant le plus d'un panneau. Toutefois, cet algorithme est loin d'être parfait. Ces candidats doivent

maintenant être filtrés pour ne retenir que les panneaux réels et supprimer toutes les fausses alarmes qui pourraient alerter inutilement le conducteur. L'objectif de la reconnaissance est ainsi double, éliminer les négatifs et classifier les panneaux pour déchiffrer l'information qui y est contenue.

La classification a pour but d'identifier la classe à laquelle appartient un panneau et cela en utilisant des classifieurs comme les Séparateurs à Vastes Marges (SVMs). Dans ce cas, une phase d'apprentissage est nécessaire afin que les classifieurs apprennent les différents pictogrammes [6].

I.2. 3. Le suivi temporel

Dans certains TSR, il y a une étape de suivi temporel. Cette étape est intégrée souvent après l'étape de Reconnaissance, le Suivi temporel identifie les panneaux déjà reconnus afin de ne pas les re-classifier ce qui réduit le temps de traitement (Cf. **Figure 1**). Cette identification est assurée par une étape d'association de données qui met en relation les panneaux connus (reconnus précédemment) avec les panneaux perçus (détectés à l'instant courant). L'association permet également de ne pas re-signaliser les mêmes panneaux au conducteur afin de ne pas le perturber [3]. À l'aide d'une étape de filtrage, le Suivi temporel peut également procurer un a priori sur les positions et les résolutions des panneaux dans les images suivantes facilitant ainsi leur Détection.

I.3 Objectif du projet

Dans ce projet, Nous nous sommes intéressés à la deuxième phase d'un TSR, la classification des panneaux routiers, en traitant ses deux étapes, l'extraction des caractéristiques et leur classification.

I.4 conclusion

Nous avons présenté dans ce chapitre l'intérêt d'un système de reconnaissance des panneaux routiers ainsi que ses étapes : la détection, la reconnaissance, et le suivi des panneaux routiers.

Après la détection d'un panneau, une étape de classification est nécessaire pour la reconnaissance de son contenu, suivi par une étape de suivi temporel, pour ne pas re-signaliser les mêmes panneaux au conducteur.

Chapitre II : la reconnaissance de panneaux routiers (la classification)

II .1. Introduction

Nous abordons dans ce mémoire la deuxième principale étape d'un TSR qui est la reconnaissance (classification). C'est pourquoi, ce chapitre discute quelques descripteurs et classifieurs proposés dans la littérature pour la reconnaissance de panneaux routiers.

II .2. Descripteurs de panneaux routiers

La première phase de la reconnaissance consiste à extraire les caractéristiques des panneaux détectés. Pour cela plusieurs descripteurs ont été proposés. On cite dans ce qui suit les plus utilisés dans les TSR.

II .2.1 Le descripteur SIFT

Lowe propose un modèle (**SIFT**, *Scale Invariant Feature Transform*) qui combine un détecteur de points d'intérêt invariant aux changements d'échelle et un descripteur basé sur la distribution des gradients calculés sur la région d'intérêt. Dans un premier temps, une région d'intérêt est définie autour de chaque point d'intérêt. Par la suite, l'amplitude et l'orientation du gradient sont calculées pour l'ensemble des pixels de la région d'intérêt. Les amplitudes obtenues sont pondérées par une gaussienne qui prend en considération la distance des pixels de la région d'intérêt par rapport au centre c.-à-d. le point d'intérêt. Ensuite, des histogrammes d'orientation, où le vote est pondéré par les amplitudes, sont générés à partir de chaque sous-région [8]. La figure 7 illustre une région de 8x8 pixels, découpée en 2x2 sous-régions. Il est nécessaire de préciser, qu'en pratique, **SIFT** travaille sur des régions de 16x16 pixels découpées en 4x4 sous-régions. Au final, la concaténation des histogrammes procure le descripteur **SIFT** qui est un vecteur de 128 éléments, correspondant aux 8 orientations pour chacune des 16 sous-régions. Pour plus de robustesse aux changements de luminosité, les histogrammes sont normalisés.

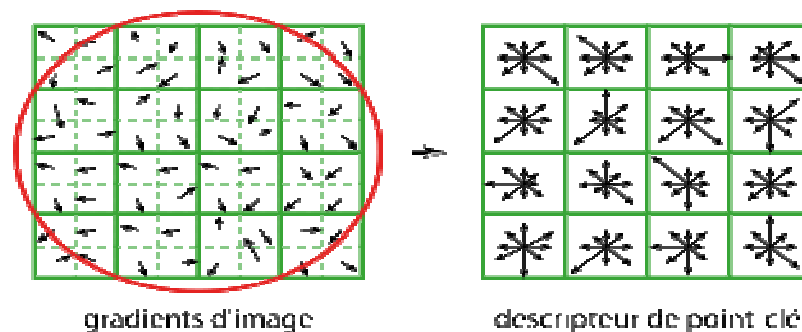


Figure 7. Descripteur SIFT [8]

Ce descripteur a été exploité dans la reconnaissance de panneaux routiers par [9]. L'idée été de comparer l'illustration d'un panneau avec une base de données de référence, Les descripteurs SIFT sont utilisés pour apparier les pictogrammes d'un panneau candidat avec ceux des bons panneaux dans la base de référence.

II.2 .2. Histogramme de Gradients Orientés

Proposés par **Dalal et Triggs**, les Histogramme de Gradients Orientés (**HOGs**) sont basés principalement sur l'algorithme **SIFT** et connus pour leurs performances dans les systèmes de détection de piétons [10]. Ces histogrammes ont également été utilisés pour la reconnaissance de panneaux dans [11] où ils ont utilisé 4 descripteurs de **HOG** de taille 1568, 1568, 2916 et 2592 pour évaluer les performances de deux classifieurs, K-D tree et Random Forest (discutés dans la section II .4.4), tels que les dimensions des blocks et des cellules diffèrent de l'un à l'autre comme le montre tableau 1, et la résolution des images est de 40*40 pixels. Le même descripteur est utilisé aussi dans [12] et [13].

Nom	Dimension	Cellule	Bloc
HOG1	1568	5*5	10*10
HOG2	1568	5*5	10*10
HOG3	2916	4*4	8*8
HOG4	2592	4*4	8*8

Tableau.1. Les paramètres de descripteur de **HOG** [11]

Le principe des **HOGs** consiste à décrire l'apparence de l'objet par des histogrammes d'orientation. Le calcul de ces histogrammes nécessite en premier lieu le découpage de l'image ou d'une région en blocs, eux-mêmes subdivisés en cellules (**Cf. Figure 7**).

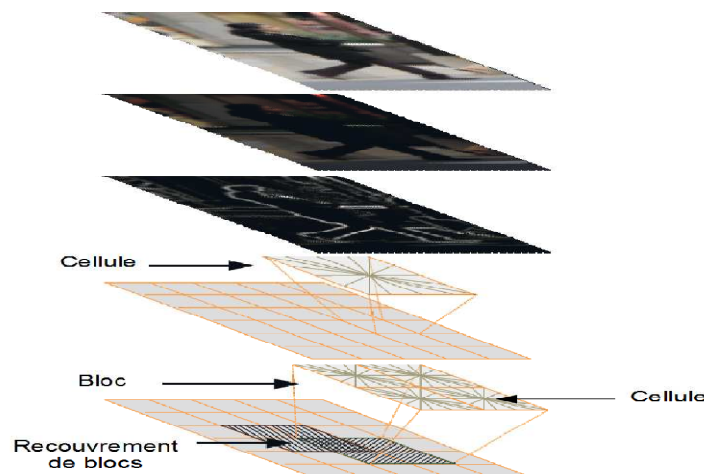


Figure 8. Le principe de décomposition pour le calcul des HOG [10]. L'image est divisée en blocs qui peuvent se recouvrir. Chaque bloc est subdivisé en cellules où un histogramme d'orientations est calculé.

Ensuite, l'amplitude et l'orientation du gradient de chaque pixel sont calculées. Dans chaque cellule, un histogramme d'orientations est formé en se basant sur l'amplitude du gradient pour pondérer les votes. Au niveau de chaque bloc les histogrammes de ces cellules sont concaténés pour former l'histogramme du bloc.

II .2.2.1 Calcul du gradient :

Comme le nom de la méthode le laisse entendre, la caractérisation utilise l'orientation du gradient. Pour chaque pixel de l'image, il est donc nécessaire de déterminer l'angle du gradient. Nous utilisons donc un filtrage de l'image selon les deux dimensions :

- horizontal : $(-1, 0, 1)$
- vertical : $(-1, 0, 1)^T$

Nous obtenons ainsi deux matrices GH et GV correspondant au gradient horizontal et vertical (Cf. **Figure 9**).

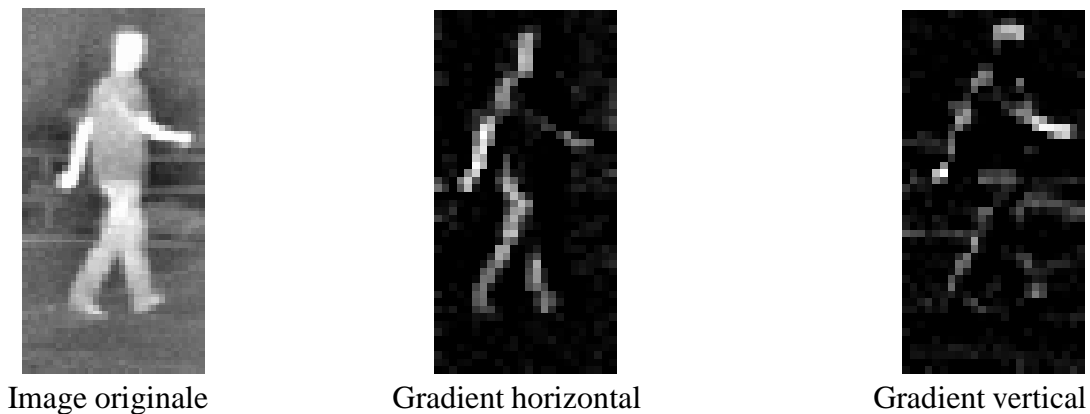


Figure 9. Image d'un piéton en infrarouge (gauche), image du gradient horizontal (milieu) et vertical (droite).

L'orientation du gradient (Cf. **Figure 10**) est définie pour chaque pixel par :

$$OG(x,y) = \text{atan} \frac{Gv(x,y)}{Gh(x,y)} \quad (1)$$

Nous calculons également la norme du gradient en chaque point (Cf. **Figure 10**):

$$NG(x,y) = \sqrt{GH(x,y)^2 + GV(x,y)^2} \quad (2)$$

Cette norme sera utilisée lors du vote dans les histogrammes.

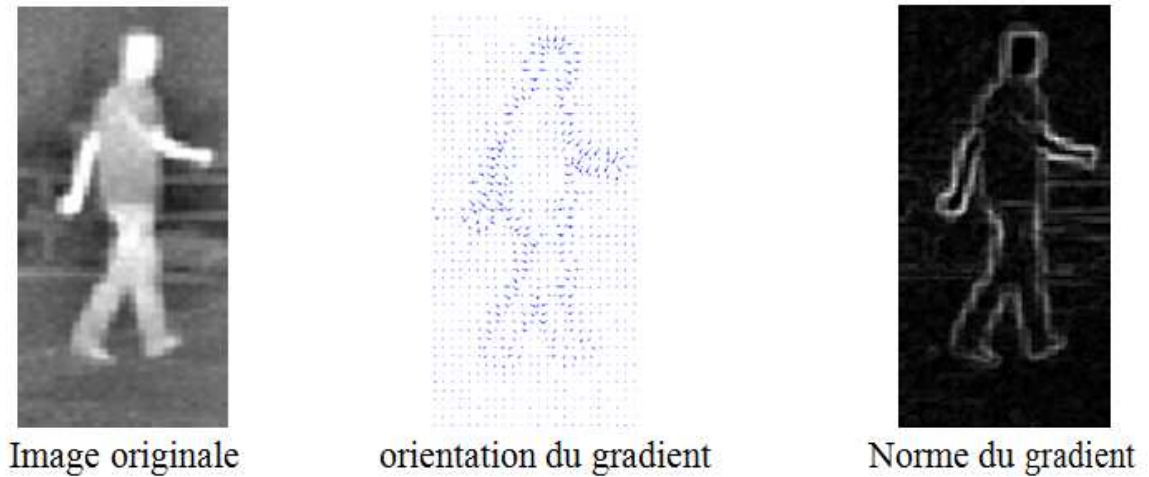


Figure 10. Exemple d'image de piéton en image infrarouge (gauche), l'orientation du gradient associée à cette image (milieu) et la norme du gradient (droite).

II .2.2.2 Découpage de l'image :

L'image est ensuite découpée en cellules (**Cf. Figure 11**). Le découpage est exhaustif et toutes les régions de l'image sont ainsi couvertes. La taille des cellules est fixée au préalable selon les besoins et les performances recherchées.

Pour des raisons pratiques, la taille d'une région est définie en pixels. Ainsi, pour obtenir des descripteurs de même taille pour toutes les images, celles-ci doivent être impérativement de même taille. Avant d'être traitée, chaque image doit donc être redimensionnée selon une taille définie. Dans notre cas, nous traitons des images de taille 40×40 pixels. Il serait cependant possible de définir la taille des cellules en fonction de la taille des images à traiter, sans avoir besoin de redimensionner ces dernières, la finalité étant d'obtenir le même nombre de cellules quel que soit la taille de l'image.

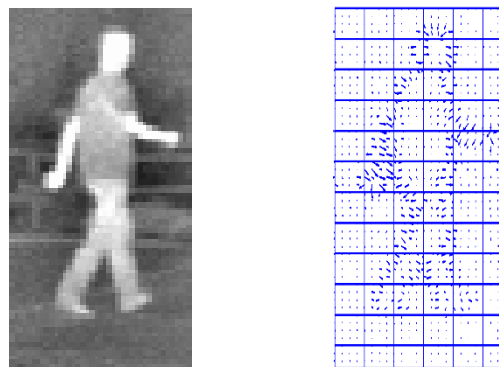


Figure 11. Découpage d'une image infrarouge (gauche) en 12×6 Cellules (droite).

II .2.2.3 Calcul des histogrammes :

Lorsque le découpage de l'image est effectué, nous calculons ainsi un histogramme d'orientation de gradient pour chaque cellule (Cf. **Figure 12**). Chaque pixel des cellules participe au vote. Celui-ci peut être pondéré par la magnitude du gradient à l'emplacement du pixel. Pour chaque pixel de coordonnées (x, y) , la valeur associée dans l'histogramme H sera :

$$H(a) = H(a) + \begin{cases} NG(x, y) & \text{si vote pondéré} \\ 1 & \text{sinon} \end{cases} \quad (3)$$

Avec a une orientation tel que $OG(x, y) \in [\theta_a, \theta_{a+1} [$.

Cette pondération permet ainsi d'accorder davantage d'importance au vote d'un pixel appartenant à un contour, qui générera donc une magnitude importante, par rapport au vote d'un pixel appartenant à une région homogène. Le vote permet ainsi de tenir compte de la forme de l'objet contenu dans l'image.

Le nombre de niveaux de l'histogramme est également définissable. En fonction de la finesse de l'histogramme, nous pourrions ainsi considérer avec plus ou moins de précision l'orientation des gradients.

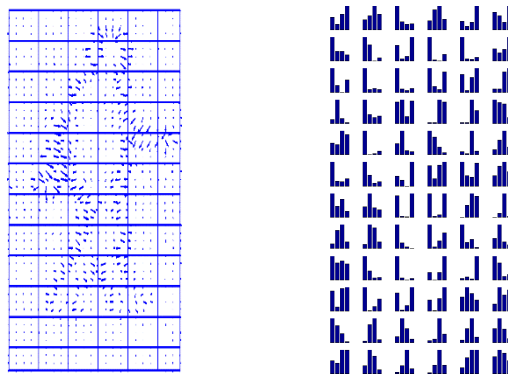


Figure 12. Calcul des histogrammes définis pour 4 niveaux d'orientation (droite) pour chaque cellule (gauche).

Pour un gradient non signé, nous avons ainsi $\theta_a = a\pi/n$ avec n le nombre de niveaux de l'histogramme.

De plus, nous pouvons choisir de prendre en compte le signe du gradient ou non. Cette particularité permet ainsi de caractériser indifféremment la forme de l'objet par rapport à sa couleur ou sa luminosité. La forme de l'objet sera prépondérante vis-à-vis de son apparence.

II .2.2.4 Normalisation des histogrammes :

La dernière étape concerne la normalisation des histogrammes. Cette normalisation est justifiée par la réduction de la variabilité dans le cas de changements d'illumination. L'idée consiste ainsi à normaliser chaque cellule au sein d'un bloc de cellules voisines. Le nombre de cellules par bloc est un paramètre supplémentaire à fixer.

Les blocs peuvent se recouvrir partiellement, ainsi les cellules sont normalisées en tenant compte de différents voisinages. Les histogrammes associés à ces cellules appartenant à différents blocs sont ainsi normalisés selon différents facteurs de normalisation, et ajoutés plusieurs fois dans le descripteur final. Le descripteur peut donc contenir de l'information redondante.

Au final, la concaténation des histogrammes de tous les blocs, forme le vecteur de « **HOG** ». La décomposition en blocs et cellules conserve l'information spatiale utile à la reconnaissance d'objets [10].

Configuration de HOG

Décrire une image à l'aide d'histogrammes fait appel à de nombreux paramètres pour gérer le découpage de l'image en cellules, le regroupement des cellules en blocs, le type de normalisation des histogrammes et les caractéristiques des histogrammes. Voici la liste des paramètres à régler :

- Cellules
 - taille des cellules : nombre de pixels définissant une cellule.
- Blocs
 - taille : nombre de cellules par bloc.
 - chevauchement : nombre de cellules entre chaque bloc.
 - facteur de normalisation.
- Histogrammes
 - vote pondéré ou non,
 - gradient signé ou non,
 - nombre de niveaux.

II .3. Les différents types d'apprentissage :

Mais avant de parler de la classification, il faut parler de l'apprentissage dans ses différents types [14].

II .3.1 Apprentissage supervisé

Dans le domaine de l'intelligence artificielle il existe différents types d'apprentissage artificiel. L'apprentissage supervisé utilise une base d'apprentissage contenant des exemples annotés [15]. A partir de cette base de départ et de ces annotations, le système est capable de classer un nouvel exemple qui n'appartient pas à la base. Dans ce cas, le classifieur prend des décisions binaires, c'est-à-dire « l'objet appartient à la classe ou l'objet n'appartient pas à la

classe ». Les bases de données seront donc constituées d'exemples positifs (un panneau) et d'exemples négatifs (autre chose).

L'objectif est de définir une règle permettant d'assigner tout motif inconnu x d'un certain espace X à une classe donnée l à partir de connaissances *à priori*. Ces dernières sont fournies sous forme d'une base d'apprentissage, représentative de la tâche de classification. Elle représente la mémoire, ou l'expérience, du système. Elle doit notamment contenir un ensemble significatif de N motifs x_i , $i \in [1, N]$ de chacune des catégories à distinguer. Si chaque motif x_i a été au préalable associé à classe, étiquetée l_i , l'algorithme de classification vise à séparer au mieux l'espace entre les L classes. L'apprentissage est dit *supervisé*.

En revanche, si cette distribution est inconnue, il est dit *non-supervisé* et l'objectif est de trouver le nombre optimal de classes permettant la meilleure séparation au sens de critères donnés.

II .3.1 Apprentissage non-supervisé

Pour l'apprentissage non-supervisé, l'ensemble d'apprentissage est seulement composé d'exemples, sans aucune étiquette de classe. C'est à l'algorithme de trouver des dépendances, des structures entre les différents exemples. Le « *clustering* » ou partitionnement de données regroupe un ensemble de méthodes d'apprentissage non-supervisé, comme l'algorithme des *K-means* [16]. Les classes (ou « *clusters* » en anglais) sont créées par l'algorithme qui regroupe dans une même classe des objets ayant des caractéristiques communes entre elles et différentes avec les objets n'appartenant pas aux mêmes classes. Prenons un exemple de classification par les *K-means*. L'algorithme est initialisé aléatoirement avec un certain nombre de clusters pour lesquels un point moyen, appelé barycentre, est évalué. A chaque itération, la distance entre chaque exemple aux différents centroïdes est calculée ; chaque exemple est alors associé au cluster dont la distance au centroïde est la plus proche. Puis les centroïdes sont réévalués. L'algorithme se termine lorsqu'il n'y a plus aucun changement.

II .3.1 Apprentissage semi-supervisé :

Ce terme regroupe des méthodes qui se situent entre l'apprentissage non-supervisé et l'apprentissage supervisé. Ce type de méthodes est utilisé quand un grand nombre de données est disponible mais sans qu'elles soient toutes étiquetées. L'initialisation de la méthode est faite à partir d'un petit jeu de données correctement étiquetées. Puis l'algorithme doit lui-même étiqueter les exemples suivants et construire son propre modèle.

Les algorithmes d'apprentissage non-supervisé et semi-supervisé sont beaucoup utilisés pour la recherche d'informations sur internet notamment. Ils permettent de traiter ainsi une grande quantité de donnée.

II .3.1 Apprentissage par renforcement :

Ce type de méthode est un apprentissage interactif. A chaque décision que l'algorithme prend, il reçoit en retour des réponses de l'environnement appelées signaux de renforcement. C'est un processus adaptatif qui améliore la solution en fonction des réponses

qu'il reçoit. L'algorithme de « *Q-learning* » [17] réalise un apprentissage par renforcement. Il produit une matrice Q dont chaque élément $Q(s, a)$ est une mesure de l'intérêt d'effectuer l'action a lorsque le système se trouve dans l'état s . Par ailleurs, des résultats théoriques garantissent, dans des cas précis, la convergence de l'algorithme vers des valeurs optimales de Q .

II .4. Classifieur de panneaux routiers

Après l'extraction des caractéristiques, différents classifieurs sont utilisés dans ce domaine pour affecter le panneau à sa classe.

Pour les TSR, plusieurs classifieurs ont été utilisés tels que les réseaux de neurones, les séparateurs à vaste marge, les K-D tree, les forêts aléatoires.

On va présenter quelques uns, en détaillons leur principe.

II .4.1 Les réseaux de neurones

Introduits par **McCulloch et Pitts**, ces modèles de calcul s'inspirent du fonctionnement des neurones biologiques [18]. Le mécanisme de décision s'appuie sur la perception et la mémoire. La structure est composée d'un ensemble d'unités de traitement interconnectées, les neurones, qui communiquent entre elles au moyen de signaux. Un réseau de neurones est constitué de cellules de trois types : celles de la couche d'entrée N_i reçoivent les informations provenant de l'extérieur, celles de la couche de sortie N_o envoient les données hors du réseau, et entre les deux les cellules de la couche cachée N_h (Cf. **Figure 13**). Plus précisément, une unité de traitement k reçoit des données y_j de N cellules en amont (Cf. **Figure 14**).

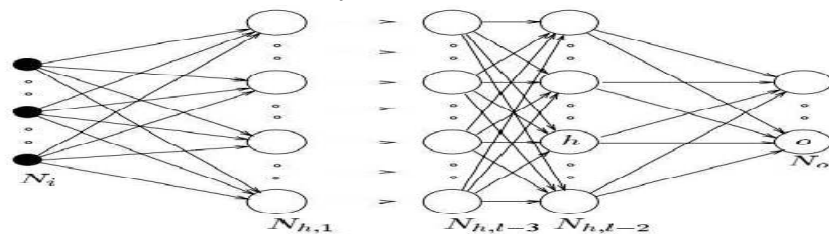


Figure 13. Réseau de neurones à $l - 2$ couches cachées [19].

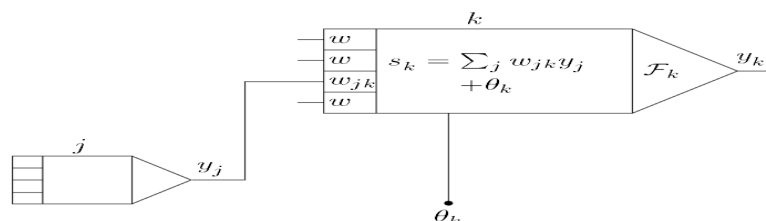


Figure 14. Unité de traitement k d'un réseau de neurones. Les données reçues des cellules en amont y_j et le biais θ_k sont pondérés par les coefficients w_{jk} . Le niveau de sortie y_k est fonction de cette somme s_k et d'une fonction d'activation F_k [19]

Les connexions sont pondérées d'un facteur w_{jk} déterminant l'effet du signal de l'unité j sur l'unité k . Le potentiel s_k de k à un instant t est ainsi calculé comme une somme pondérée de ses entrées $y_j(t)$ et d'un biais $\theta_k(t)$, définie comme étant la règle de propagation du réseau.

$$S_k(t) = \sum_j w_{jk} y_j(t) + \theta_k(t) \quad (4)$$

La sortie de l'unité est ensuite déterminée à l'aide d'une fonction d'activation F_k .

$$y_k(t+1) = F_k(S_k(t)) \quad (5)$$

La plupart du temps, les fonctions d'activation choisies sont des sigmoïdes (comme la tangente hyperbolique).

L'apprentissage consiste à adapter les poids des connexions en fonction des données et de la topologie du réseau. Si l'information entrant dans une unité ne provient que des neurones situés en amont, le réseau est non bouclé. Le modèle du perceptron introduit par [20] appartient à cette catégorie. Pour adapter les poids, l'apprentissage des réseaux neuronaux utilise une méthode de gradient visant à minimiser l'écart entre les sorties obtenues et celles correspondant aux données. Cette descente de gradient se fait généralement à l'aide d'une **rétropropagation du gradient** [21]. Le principe est de propager en arrière l'erreur commise en sortie sur toutes les connexions des unités ayant pris part à cette décision. L'importance de la correction dépend de l'influence de l'élément dans la prise de décision.

Dans le cadre de la reconnaissance de panneaux, de nombreuses études mettent en œuvre des réseaux de neurones pour la phase de classification. Dans [22] un réseau spécifique est utilisé pour chaque catégorie de panneaux de résolution 50*50. Dans [23] les auteurs comparent différents types de réseaux sur des panneaux routiers circulaires et triangulaires de résolution 30*30. Les auteurs dans [24] ont utilisé ce classifieur sur des panneaux français et allemands dans divers contextes routiers (autoroute, ville, routes de campagne) sous différentes conditions d'illuminations. Les résultats obtenus atteignaient les 94% de bonne reconnaissance.

II .4.2 Les k plus proches voisins

La méthode des K plus proches voisins (**K-PPV**) (K Nearest Neighbour (**K-NN**)). Aucun apprentissage ne prend réellement place, c'est-à-dire qu'il n'y a pas de phase consistant en la construction d'un modèle à partir d'un échantillon d'apprentissage [25].

Le principe est le suivant :

Etant donnée une base d'apprentissage d'images étiquetées correctement et un entier k, le classifieur **K-NN** détermine la classe d'un nouvel objet en lui attribuant la classe majoritaire des k objets lui ressemblant le plus dans la base d'apprentissage. La figure 15 illustre le principe de fonctionnement de ce classifieur qui se base sur le calcul de la distance entre l'exemple en question, et les exemples de l'échantillon, puis on lui attribue la classe majoritaire des 3 objets qui rassemble le plus.

K-NN est utilisé dans [26] en exploitant les caractéristiques de **HOG**. Evaluée sur la base de données GTSRB, la méthode **KNN** procure un taux de reconnaissance de **96,97%**.

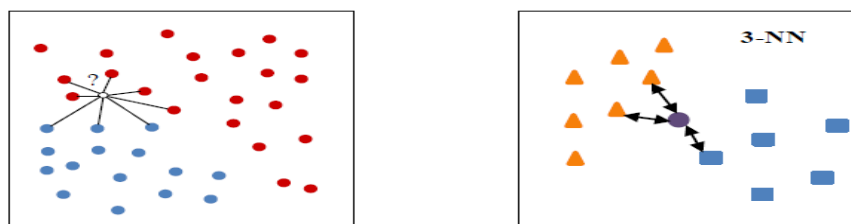


Figure.15. Principe de **K-PPV** [25]

II .4.3 Séparateurs à Vastes Marges (SVM)

II .4.3.1 Introduction

Les machines à vecteurs de support (Support Vector Machine, SVM) appelés aussi séparateurs à vastes marges sont des techniques d'apprentissage supervisées destinées à résoudre des problèmes de classification [27]. Les SVM exploitent les concepts relatifs à la théorie de l'apprentissage statistique et à la théorie des bornes de Vapnik et Chervonenkis. La justification intuitive de cette méthode d'apprentissage est la suivante : si l'échantillon d'apprentissage est linéairement séparable, il semble naturel de séparer parfaitement les éléments des deux classes de telle sorte qu'ils soient le plus loin possibles de la frontière choisie. Ces fameuses machines ont été inventées en 1992 par Boser et al, mais leur dénomination par SVM n'est apparue qu'en 1995 avec Cortes et al. Depuis lors, de nombreux développements ont été réalisés pour proposer des variantes traitant le cas non-linéaire. Le succès de cette méthode est justifié par les solides bases théoriques qui la soutiennent. Elles permettent d'aborder des problèmes très divers dont la classification. SVM est une méthode particulièrement bien adaptée pour traiter des données de très haute dimension.

Plusieurs travaux utilisent les **SVM** pour la reconnaissance des panneaux routiers comme [13] en exploitant les caractéristiques de **HOG**. Les SVM sont appliqués aussi sur les images couleurs comme dans [28] où un taux de 95,71% de reconnaissance a été atteint.

II .4.3.2 Principe de la technique SVM

Cette technique est une méthode de classification binaire qui tente de séparer les exemples positifs des exemples négatifs dans l'ensemble des exemples. La méthode cherche alors l'hyperplan qui sépare les exemples positifs des exemples négatifs, en garantissant que la marge entre le plus proche des positifs et des négatifs soit maximale (**Cf. Figure 16**). Cela garantit une généralisation du principe car de nouveaux exemples pourront ne pas être trop similaires à ceux utilisés pour trouver l'hyperplan mais être situés d'un côté ou l'autre de la frontière. L'intérêt de cette méthode est la sélection de vecteurs supports qui représentent les vecteurs discriminant grâce auxquels est déterminé l'hyperplan. Les exemples utilisés lors de la recherche de l'hyperplan ne sont alors plus utiles et seuls ces vecteurs supports sont utilisés pour classer un nouveau cas, ce qui peut être considéré comme un avantage pour cette méthode.

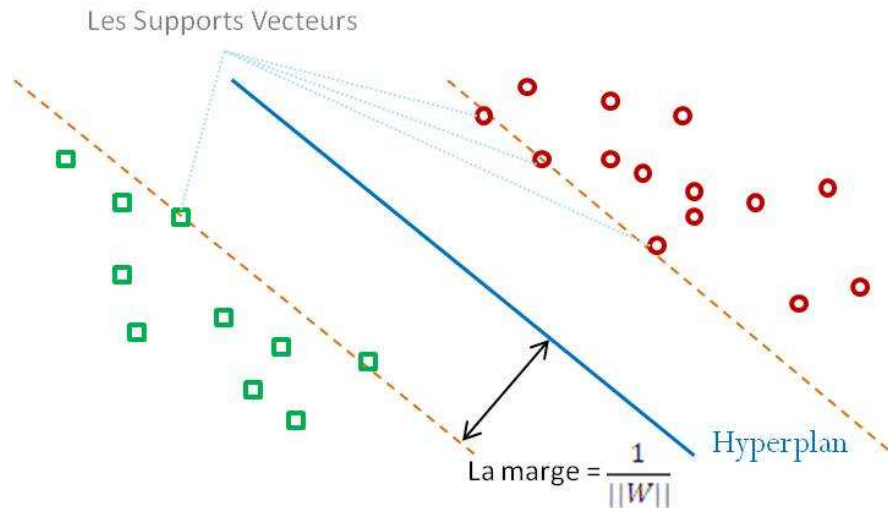


Figure 16. Principe de K-PPV [25]

Considérons un ensemble de N exemples linéairement séparables $x_i \in \mathbb{R}^k$ étiquetés $l_i \in \{-1; 1\}$. La séparation entre les deux classes est modélisée par un hyperplan H :

$$H : \omega x + b = 0 \quad (6)$$

La distance minimale entre H et l'origine vaut donc $\frac{|b|}{\|\omega\|}$ (Cf. Figure 17). Les points les plus proches de l'hyperplan pour chaque classe sont appelés les **vecteurs supports** et sont situés à une distance d^- (resp. d^+) pour la classe -1 (resp. $+1$). Ils forment deux hyperplans H^+ et H^- choisis de sorte que leur distance $\frac{2}{\|\omega\|}$ soit maximale. Les performances de la classification sont optimales lorsque la marge est maximale, ce qui revient à minimiser $\frac{\|\omega^2\|}{2}$. Ce problème peut être résolu via les multiplicateurs de Lagrange $\alpha_i \geq 0$.

$$\frac{\|\omega^2\|}{2} - \sum_{i=1}^m \alpha_i l_i (\omega x_i + b) - 1 \quad (7)$$

La fonction de classification pour un exemple inconnu x est alors :

$$f(x) = \text{sgn}(\omega x + b) = \text{sgn}(\sum_{i=1}^m \alpha_i l_i (x_i \cdot x) + b) \quad (8)$$

Les exemples de la base d'apprentissage vérifient alors :

$$\begin{cases} \omega x_i + b \geq 1 & \text{si } l_i = 1 \\ \omega x_i + b \leq -1 & \text{si } l_i = -1 \end{cases} \implies l_i (\omega x_i + b) - 1 \geq 0 \quad (9)$$

$$(10)$$

En présence de bruit, le SVM peut être rendu plus tolérant aux exemples mal classifiés par l'ajout de variables "molles" ζ_i .

$$\begin{cases} \omega x_i + b \geq 1 - \xi_i & \text{si } l_i = 1 \\ \omega x_i + b \leq -1 + \xi_i & \text{si } l_i = -1 \end{cases} \quad (11)$$

$$\begin{cases} \omega x_i + b \geq 1 - \xi_i & \text{si } l_i = 1 \\ \omega x_i + b \leq -1 + \xi_i & \text{si } l_i = -1 \end{cases} \quad (12)$$

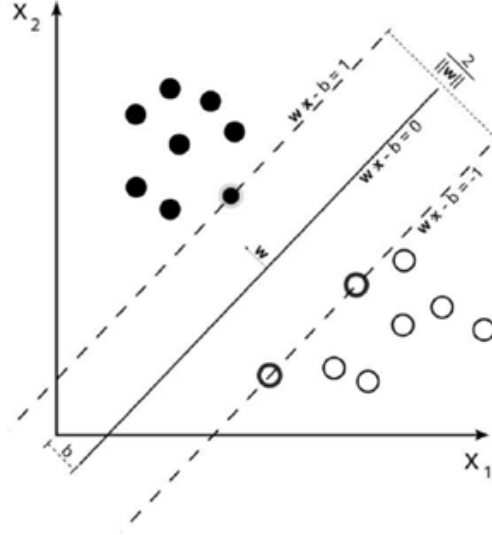


Figure 17. Séparation linéaire des données de la base d'apprentissage par l'hyperplan $H : wx + b = 0$. Les vecteurs de support sont les points ayant une distance minimale avec H .

Toutefois, les données ne sont pas nécessairement séparables de façon linéaire. Elles sont alors transformées dans un espace de dimension supérieure grâce à une fonction non linéaire $z = \Phi(\mathbf{x})$.

La séparation linéaire sera effectuée dans ce nouvel espace. Pour éviter le calcul explicite de Φ , le produit scalaire $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ est remplacé par un noyau K . Pour déterminer l'étiquette d'un exemple x , nous étudions le signe de $w\Phi(\mathbf{x}) + b$.

$$f(x) = \text{sgn}(w\Phi(x) + b) \quad (13)$$

$$= \text{sgn}\left(\sum_{i=1}^m \alpha_i l_i (\Phi(x_i) \cdot \Phi(x)) + b\right) \quad (14)$$

$$= \text{sgn}\left(\sum_{i=1}^m \alpha_i l_i K(x_i, x) + b\right) \quad (15)$$

Parmi les noyaux les plus utilisés, nous retrouvons :

– le noyau linéaire $K(x_i, x_j) = x_i x_j$. (16)

– le noyau polynomial $K(x_i, x_j) = (\gamma x_i \cdot x_j + r)^d$, $\gamma \in \mathbb{R}^+$, $r \in \mathbb{R}$, $d \in \mathbb{N}$. (17)

– le noyau gaussien $K(x_i, x_j) = e^{(-\gamma \|x_i - x_j\|^2)}$ (18)

Dans notre cas, on va utiliser un noyau linéaire.

Pour gérer les problèmes multi classes, il existe deux solutions, utiliser des SVM de type

un-contre-un ou *un-contre-tous*. Dans le premier cas, chaque classe est modélisée contre une autre, c'est-à-dire que tous les éléments de l_i sont considérés comme positifs alors que tous ceux de la classe l_j sont vus comme négatifs. Pour N classes, il y aura au final $N(N - 1)/2$ classifieurs. Un exemple inconnu sera associé à la classe présentant la confiance la plus forte. Dans le second cas, une classe est cette fois séparée de toutes les autres.

Le principal avantage des SVMs est leur capacité de généralisation, ainsi que la relative rapidité de leur apprentissage, et le déterminisme de ce dernier. Ils cherchent à minimiser la limite sur l'erreur de généralisation du modèle plutôt que minimiser l'erreur sur la base d'apprentissage.

Ainsi, les exemples d'apprentissage situés loin de l'hyperplan ne changeront pas les vecteurs de support, indiquant par là une meilleure classification d'exemples non rencontrés. Ils sont de plus peu sensibles à la dimension de l'espace des caractéristiques, contrairement aux réseaux de neurones et la classification est possible même avec des bases de données de petite dimension. Ils sont utilisés dans de nombreuses applications, reconnaissance d'écriture manuscrite, classification de texte, détection d'objet, etc.

II .4.4 K-D tree et la forêt aléatoire

Un arbre **K-d** est un arbre de recherche binaire permettant de séparer des motifs représentés par un vecteur de caractéristiques de dimension K [29]. Chaque nœud, hormis les feuilles, sépare les données en deux selon la i -ème caractéristique f_i dont la variance pour le sous-ensemble concerné est maximale. Pour s'assurer d'obtenir un arbre équilibré, la médiane mi est utilisée telle que le sous-arbre gauche vérifie la propriété $f_i < mi$ et inversement à droite (Cf. **Figure 18**). L'arbre **K-d** est un algorithme de recherche du plus proche voisin. Un exemple est en effet classifié en descendant l'arbre jusqu'à atteindre une feuille, c'est-à-dire une classe donnée.

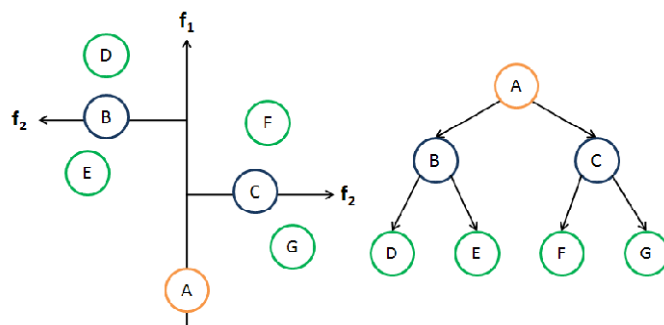


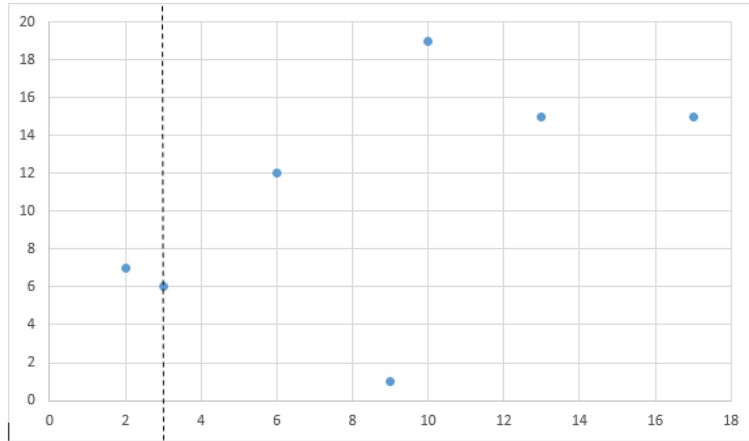
Figure 18. Illustration de la construction d'un arbre K-d. le nœud A est séparé en 2 nœuds en utilisant la caractéristique f_1 , puis B et C sont séparés chacune en deux nœuds aussi en utilisant les caractéristiques f_2 et f_3 en $\{D,E\}$ et $\{F,G\}$ [4]

Exemple de la construction de l'arbre 2-D :

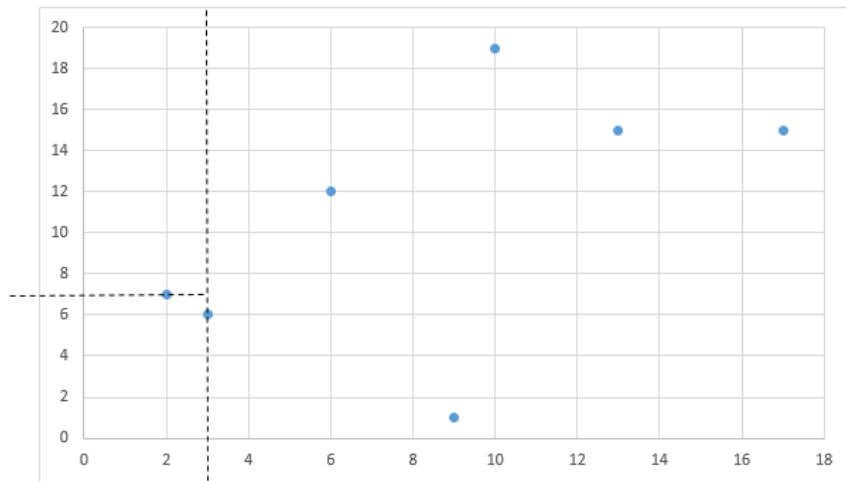
Considérant les points suivant : (3,6) ;(17,15) ;(13,15) ;(6,12) ;(9,1) ;(2,7) ;(10,19)

1- le point (3,6) va diviser l'espace en deux parties selon la caractéristique « x » :

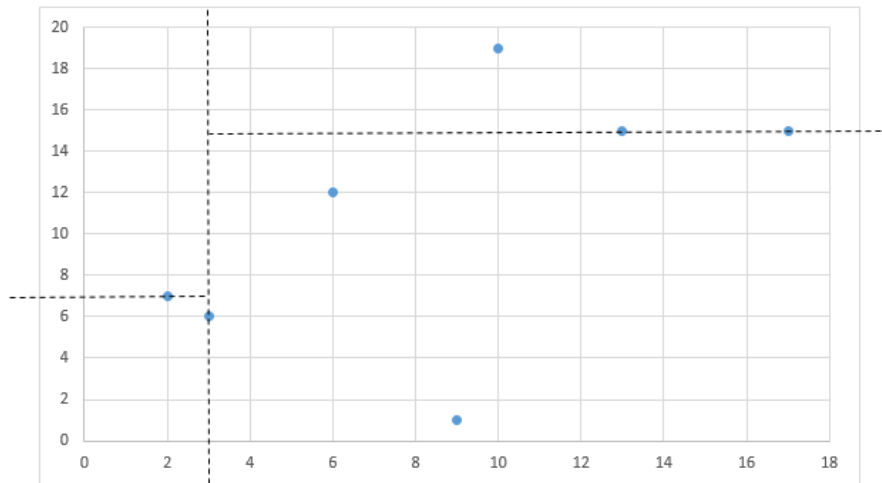
Tracer la ligne $X=3$.



2- le point (2,7) va diviser la partie gauche en deux parties horizontalement :
Tracer la ligne $Y=7$ à gauche de la ligne $X=3$.

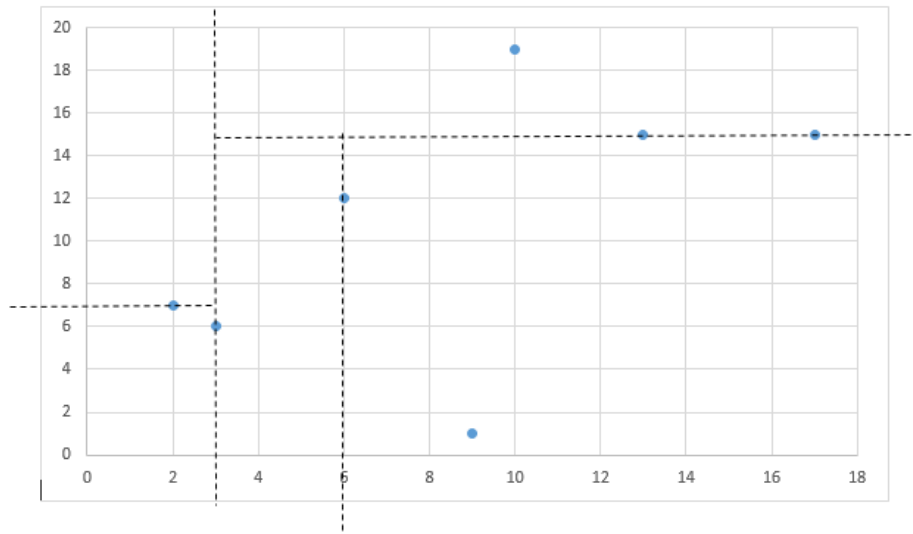


3- le point (17,15) va diviser l'espace à gauche de la ligne $X=3$ en deux parties horizontalement:
Tracer la ligne $Y=15$ à gauche de la ligne $X=3$.



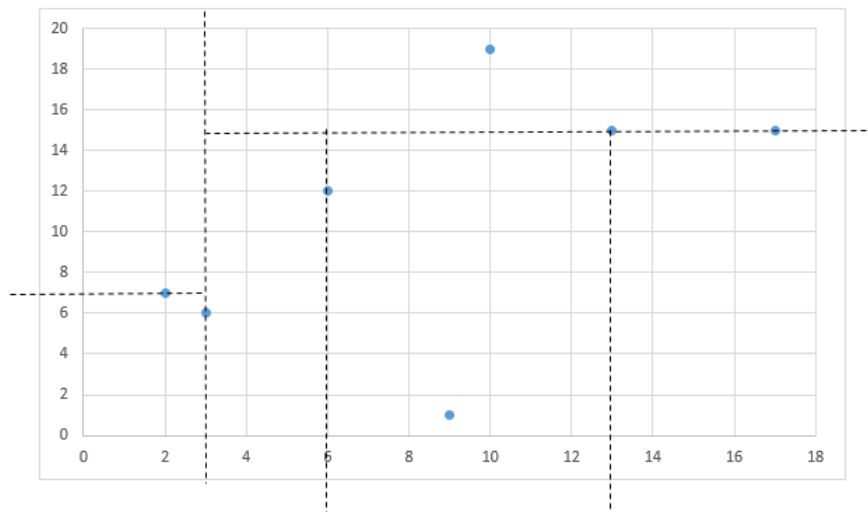
4-le point (6,12) va diviser l'espace au dessous de la ligne $Y=15$ et à gauche de la ligne $X=3$ en deux parties verticalement :

Tracer la ligne $X=6$ à droite de la ligne $X=3$ et au dessous de la ligne $Y=15$.



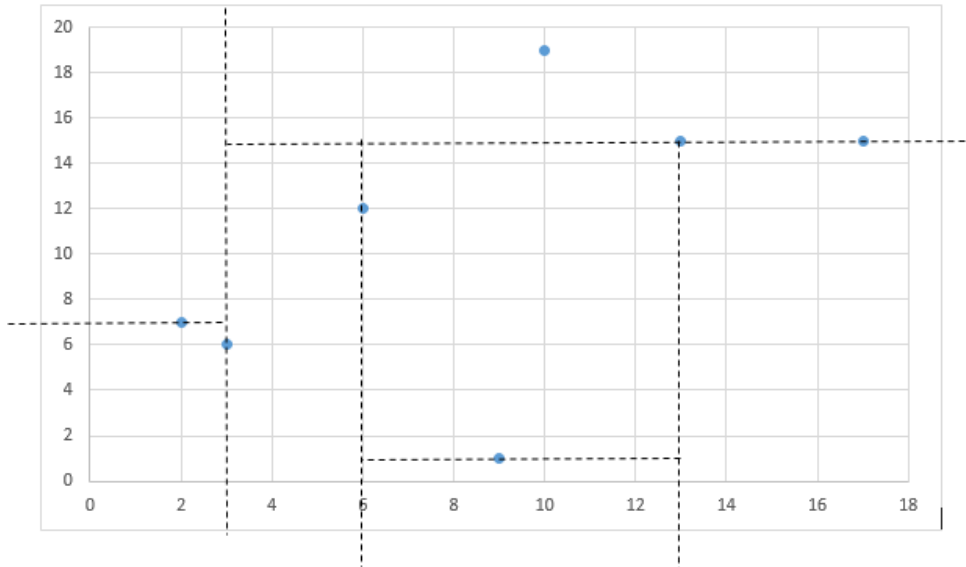
5-le point (13,15) va diviser l'espace au dessous de la ligne $Y=15$ et à droite de la ligne $X=6$ en deux parties :

Tracer la ligne $X=13$ à droite de la ligne $X=6$ et au dessous de la ligne $Y=15$.



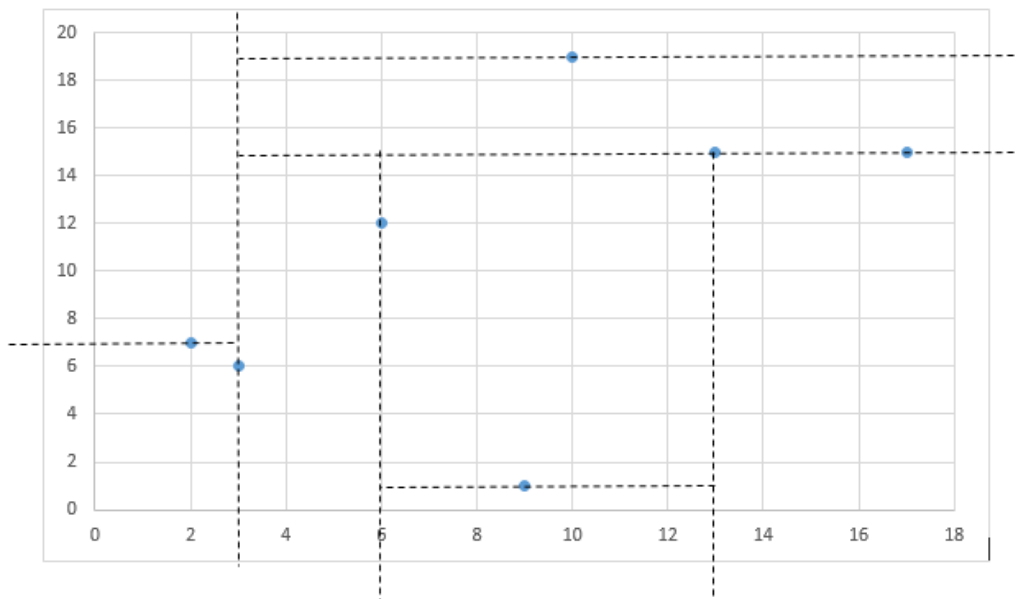
6-le point (9,1) va diviser l'espace entre les lignes $X=3$ et $X=6$ en deux parties horizontalement

Tracer la ligne $Y=1$ entre les lignes $X=3$ et $X=6$



7-le point (10,9) va diviser l'espace à droite de la ligne $X=3$ et au dessus de la ligne en deux parties horizontalement :

Tracer la ligne $Y=19$ à droite de la ligne $X=3$ et au dessus de la ligne $Y=15$



Lors de l'utilisation des vecteurs HOG. La recherche est effectuée en utilisant une file de priorité contenant les candidats classés en fonction de leur distance par rapport à la requête. Le classement détermine l'ordre dans lequel les nœuds sont examinés. Pendant la recherche, les nœuds examinés sont ajoutés de manière itérative à la file de priorité. La recherche est terminée lorsque l'algorithme scanne un nombre maximum prédéfini de nœuds E_{max} .

Pendant la phase de test, les k plus proches voisins (**KNN**) sont récupérés pour chaque candidat \mathbf{X} . Le vote de la classe l_i est pondéré par l'inverse de la distance d à \mathbf{X} (généralement la distance euclidienne). Le vote maximal détermine la classe de \mathbf{X} .

$$vote_{l_i} = \sum_1^L \frac{1}{d(k_{NN}, \mathbf{X})}, k_{NN} \in l_i \quad (19)$$

K-d tree a été utilisé par **Zaklouta et Stanculescu** en exploitant les caractéristiques de HOG sur des images de résolution 40×40 pixels de la base **GTSRB** [12]. Cette dernière contient 43 classes, dont 26640 images d'apprentissage, et 12569 images de test, **Zaklouta et Stanculescu** ont obtenu un taux de reconnaissance de 92,9%.

Une des premières limitations de ce type d'arbre est sa sensibilité au bruit, une simple erreur à un embranchement et la classification est fautive.

Les forêts aléatoires [30] proposent une solution à ce problème en générant un ensemble d'arbres aléatoires. Aléatoires dans le sens où :

- un sous-ensemble de la base d'apprentissage est choisi aléatoirement.
- à chaque nœud, un sous-ensemble de caractéristiques est choisi. Un compromis sur sa taille doit être trouvé entre un nombre trop élevé qui conduirait à une sur-segmentation et un trop faible qui augmenterait la vitesse d'apprentissage mais aussi le risque de sous-segmentation.

La figure 19 illustre la classification d'un exemple x dans une forêt aléatoire. Chaque arbre aléatoire $\mathbf{T}_j \in \{\mathbf{T}_1; \mathbf{T}_2; \dots; \mathbf{T}_T\}$ dans la forêt est traversé. A la feuille atteinte en \mathbf{T}_j , la probabilité à postériori que x appartienne à la classe $L \in \{1, 2, \dots, L\}$ est désigné par $P_j(l/x)$. La classe l^* d'un exemple inconnu est déterminé en combinant les probabilités *a posteriori* obtenues pour chaque arbre.

$$p(l/x) = \frac{1}{T} \sum_{j=1}^T P_j(l/x) \quad (20)$$

$$l^* = \max_L P(l/x) \quad (21)$$

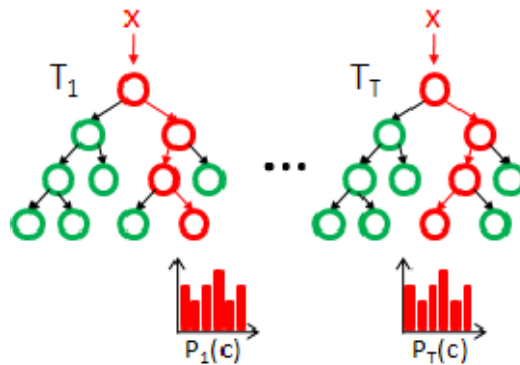


Figure 19. Classification d'un exemple en utilisant les forêts aléatoires [6]

De par leur facilité d'utilisation, les forets aléatoires sont employés dans bon nombre d'applications. Elles sont bien adaptées dans le cas de bases de données déséquilibrées et permettent même un classement des caractéristiques selon un critère d'importance.

Ce classifieur a été utilisé dans [6] en exploitant les caractéristiques de HOG sur la base **GTSRB**, et a atteint un taux de classification de 97%.

II .5. Conclusion

Nous avons vu dans ce chapitre quelques descripteurs et classifieurs utilisés pour la reconnaissance des panneaux routiers, et la partie principale pour le classifieur qui est l'apprentissage, ainsi que ses différents types.

Nous discuterons dans le chapitre suivant notre approche, et les différents outils utilisés ainsi que les résultats obtenus.

Chapitre 3 : Conception et implémentation

III.1. Introduction

Après avoir pris connaissance dans les chapitres précédents des différents descripteurs et classifieurs utilisés pour les TSR, nous présentons dans ce chapitre les outils utilisés pour le développement de notre système, et son mise en pratique, ainsi que les différents paramètres utilisés pour l'extraction des caractéristiques de HOG, et la stratégie utilisée pour la classification. Nous validons notre travail par la description de différentes expérimentations réalisées et les résultats obtenus.

III.2. Ressources utilisées

Les ressources physiques exploitées :

- Processeur Intel(R) Core(TM) i3-2350M CPU @ 2.30GHZ.
- Mémoire vive d'une capacité de 2Go.

Et comme ressource logicielle, nous avons utilisé :

- Système d'exploitation : Windows7.
- Langage de programmation : C++.
- L'EDI : Visual Studio Express 2013.
- La bibliothèque OpenCV 3.0.

III.3. Pourquoi Visual Studio Express 2013

Microsoft Visual Studio Express est un ensemble d'environnements de développement intégrés gratuits, développé par Microsoft. Il s'agit d'une version allégée de Microsoft Visual Studio. L'idée de ces éditions « express » est, selon Microsoft, de fournir un environnement de développement facile à utiliser et à apprendre pour des jeunes ou des passionnés. Visual Studio n'est évidemment pas uniquement un éditeur de code destiné aux développeurs, c'est une plateforme où les différents intervenants (développeurs, chefs de projets, testeurs, etc.) en charge d'un projet logiciel vont pouvoir collaborer tout au long du développement d'un projet.

III.4. Pourquoi C++

Le C++ est un langage de programmation très répandu et très utilisé : on s'en sert pour réaliser des programmes embarqués dans des appareils mobiles comme pour des jeux vidéo, des logiciels de gestion de données comme des compilateurs pour d'autres langages de programmation. Le C++ est un langage où l'on peut plus ou moins tout faire, et de différentes façons, ce qui fait sa puissance mais en le rendant de fait plus difficile à apprendre.

C'est au début des années 80 que Bjarne Stroustrup a inventé le "C with classes" : à ce moment-là, il ne s'agissait que d'un C auquel on avait rajouté des classes et de l'héritage. Depuis plus de 20 ans, le langage C++ a énormément évolué et il a été normalisé. Aujourd'hui, C et C++ sont deux langages différents bien qu'ils partagent un peu d'histoire.

Le C++ est un langage multi-paradigmes : objet, fonctionnel, générique sont très communs en C++.

C++ est actuellement le 2^{ème} langage le plus utilisé dans le monde. Le langage C++ n'appartient à personne et par conséquent n'importe qui peut l'utiliser sans avoir besoin d'une autorisation ou obligation de payer pour avoir le droit d'utilisation. Nous avons choisi ce langage parce qu'il est très rapide en exécution (assez bas niveau) et portable.

III.5. Pourquoi OpenCV :

OpenCV est une bibliothèque libre de vision par ordinateur. Cette bibliothèque est écrite en C et C++ et peut être utilisée sous Linux, Windows et Mac OS X. Des interfaces ont été développées pour Python, Ruby, Matlab et autre langage. OpenCV est orientée vers des applications en temps réel. Un des buts d'OpenCV est d'aider les gens à construire rapidement des applications sophistiquées de vision à l'aide d'infrastructure simple de vision par ordinateur. La bibliothèque d'OpenCV contient près de 500 fonctions. Il est possible grâce à la « licence de code ouvert » de réaliser un produit commercial en utilisant tout ou partie d'OpenCV.

III.6. La base de données :

Les panneaux de signalisation sont caractérisés par une grande variabilité dans leur apparence visuelle dans des environnements du monde réel. Par exemple, les changements d'éclairage, des conditions météorologiques variables et occlusions partielles impactent sur la perception de la signalisation routière (Cf. **Figure 20**). Dans la pratique, un grand nombre de différentes classes de panneaux doit être reconnu avec une très grande précision. Les panneaux de signalisation ont été conçus pour être facilement lisibles pour les humains, qui exécutent très bien cette tâche. Cependant, Pour les systèmes informatiques, la classification des panneaux de signalisation pose un problème de reconnaissance des formes difficiles. Le benchmark allemand de la reconnaissance des panneaux routiers GTSRB (German Traffic Sign Recognition Benchmark) présente plus de 50.000 images de panneaux routiers allemands dans 43 classes.

L'ensemble de données de GTSRB est extrait de 10 heures de vidéo qui a été enregistrée sur différents types de routes en Allemagne pendant la journée.

La base d'apprentissage est représentée par 39 209 images, et La base de données de test est constituée de 12 630 images de panneaux appartenant à 43 classes distinctes.



Figure 20. Exemples des panneaux utilisés dans notre application

Pour notre projet, nous nous sommes intéressés qu'à 5 classes de panneaux uniquement, les panneaux de limitations de vitesse de 50, 60, 80, 100 et 120 Km/h.

III.7. Diagramme de l'application

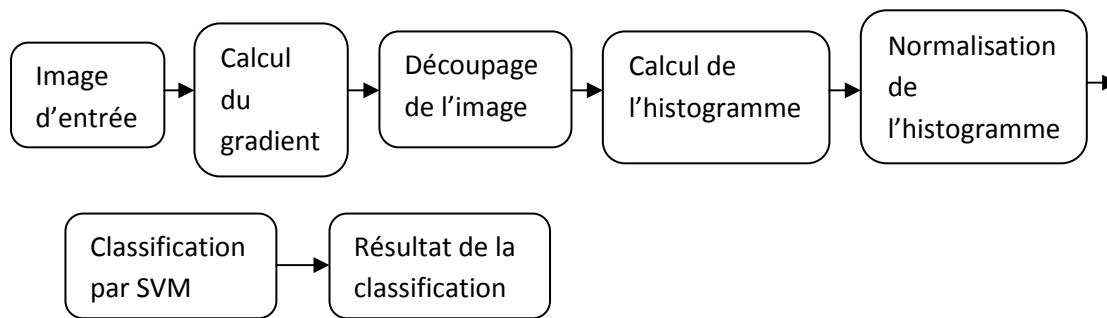


Figure 21. Diagramme de l'application

III.8. L'extraction des descripteurs

Pour l'extraction de descripteur, les histogrammes de gradients orientés (HoG) sont utilisés vu leurs bonnes performances pour la reconnaissance d'objets dans plusieurs applications telles que la détection de visages, détection de piétons, détection de véhicules, reconnaissance de panneaux, etc.

Nous avons utilisé trois configurations différentes de HOG : HOG1, HOG2, et HOG3, comme le montre le **Tableau 2**.

Nom	Dimension	Cellule	Bloc	nombre de niveaux dans l'histogramme
HOG1	576	8 x 8	16 x 16	9 bins
HOG2	2592	4 x 4	8 x 8	8 bins
HOG3	1568	5 x 5	10 x 10	8 bins

Tableau.2. Les paramètres de descripteur de **HOG de notre système**

III.9. Classification des descripteurs

Pour la classification nous avons opté pour les SVMs. En effet ces derniers peuvent aborder des problèmes très divers dont la classification. SVM est une méthode particulièrement bien adaptée pour traiter des données de très haute dimension.

Le plus grand intérêt de cette méthode est d'apporter une solution automatique au difficile compromis à faire entre la fidélité aux données d'apprentissage et la création d'un bon modèle discriminant.

Pour implémenter notre reconnaissance de panneaux, nous avons adopté la stratégie un-contre-tous. Dans ce cas, on a obtenu cinq classifieurs, un pour chacune des classes : 50, 60, 80, 100 et 120 km/h.

Chaque classe est modélisée contre les autres, c'est-à-dire que lors de l'apprentissage tous les éléments de li (exemples représentant la classe i) sont considérés comme positifs alors que le reste des exemples est vu comme négatif.

III.10. Résultats obtenus

Comme mentionné avant, la reconnaissance des panneaux routiers passe par deux phases, l'extraction des caractéristiques et leur classification.

Nous avons exploité trois configurations différentes de HOG: (HOG1, HOG2, et HOG3).

Pour la classification on a utilisé une classification multi-classe, pour cela on a utilisé Cinq SVM linéaires avec 1000 itérations, où chaque SVM classe les panneaux avec la stratégie un-contre-tout.

Le tableau 3 nous montre les différents taux d'évaluation obtenu pour chaque configuration de HOG.

Dans la première configuration, le taux de bonne reconnaissance été de 0.97.

La deuxième et la troisième configuration ont presque donné le même taux de bonne reconnaissance, 0.98.

Nom	Taux d'évaluation
HOG1	0.89
HOG2	0.87
HOG3	0.88

Tableau.3. Le taux d'évaluation pour chaque configuration de **HOG**.

III.10. Validation

Pour valider notre application nous avons calculé la matrice de confusion pour chacune des configurations de HOG (HOG1, HOG2, HOG3).

La matrice de confusion, dans la terminologie de l'apprentissage supervisé, est un outil servant à mesurer la qualité d'un système de classification.

Chaque ligne de la matrice représente le nombre d'occurrences d'une classe estimée, tandis que chaque colonne représente le nombre d'occurrences d'une classe réelle (ou de référence).

Dans notre système, les lignes 1, 2, 3, 4 et 5 (successivement) représentent le nombre d'occurrences des classes estimée, c-à-d; 50, 60, 80, 100 et 120 (successivement), et les colonnes 1, 2, 3, 4 et 5 (successivement) représentent le nombre d'occurrences des classes réelles, c-à-d; 50, 60, 80, 100 et 120 (successivement),

Un des intérêts de la matrice de confusion est qu'elle montre rapidement si le système parvient à classer correctement ou pas.

Me figures suivantes (22, 23, 24) montre la matrice de confusion, et le **CRR** obtenu pour chaque configuration de **HOG**.

Matrice de confusion pour HOG 1 :

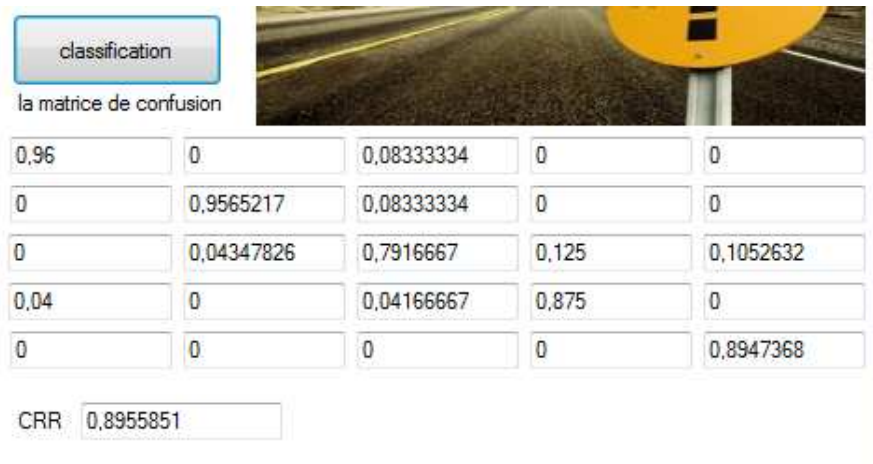


Figure 22. La matrice de confusion et le CRR obtenus avec la configuration HOG1

Matrice de confusion pour HOG2 :

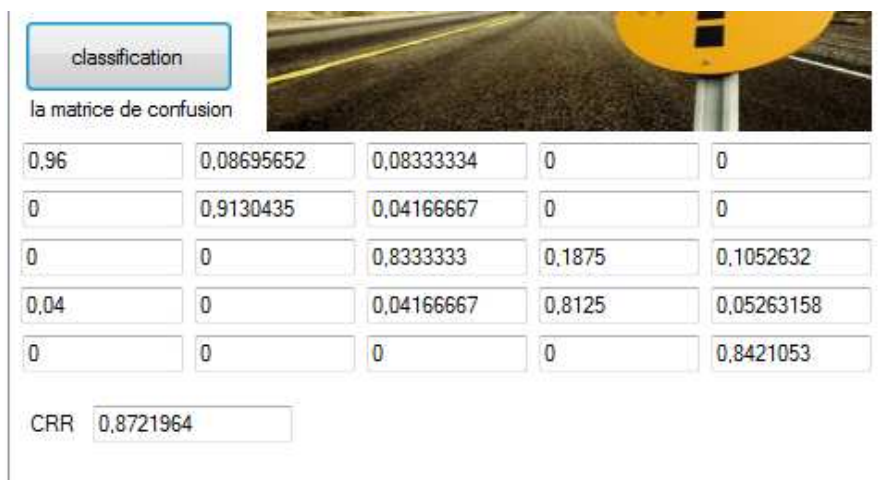


Figure 23. La matrice de confusion et le CRR obtenus avec la configuration HOG2

Matrice de confusion pour HOG 3 :

0,96	0	0,08333334	0	0,05263158
0	1	0,04166667	0	0,05263158
0	0	0,83333333	0,1875	0,05263158
0,04	0	0,04166667	0,8125	0
0	0	0	0	0,8421053

CRR 0,8895877

Figure 24. La matrice de confusion et le CRR obtenus avec la configuration HOG3

Pour le **CRR (Correct Rate Recognition)** on a divisé la somme de la diagonale de la matrice de confusion par le nombre de classe. Les résultats obtenus pour chaque configuration de **HOG** sont montrés sur le tableau 4.

Nom	CRR
HOG1	0.89
HOG2	0.87
HOG3	0.88

Tableau.4. Le CRR pour chaque configuration de HOG.**III.11. Présentation de l'application**

Notre application comporte deux fonctionnalités principales : l'apprentissage et la classification comme le montre la figure 25.

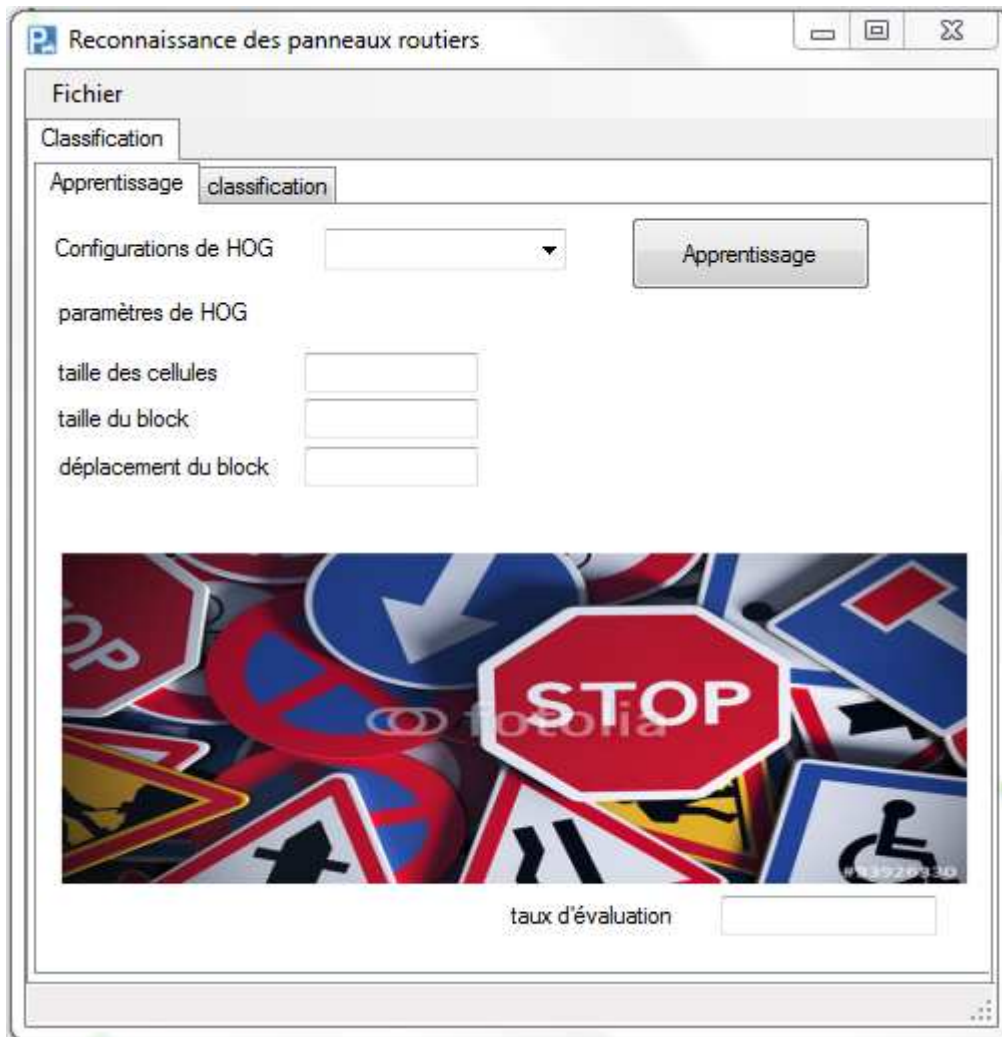


Figure 25. La première fonctionnalité de l'application : l'apprentissage

Les fonctionnalités de l'application :

III.11.1. l'apprentissage

Cette partie permet d'entraîner et d'évaluer le classifieur.

La partie apprentissage est basée sur l'extraction des descripteurs et leur classification. Pour l'extraction des caractéristiques, trois configurations de HOG sont utilisées selon le choix de l'utilisateur : HOG 1, HOG 2, HOG 3.

Donc comme le montre la figure 26, l'utilisateur peut choisir la configuration de HOG qu'il souhaite utiliser pour la reconnaissance de son panneau.

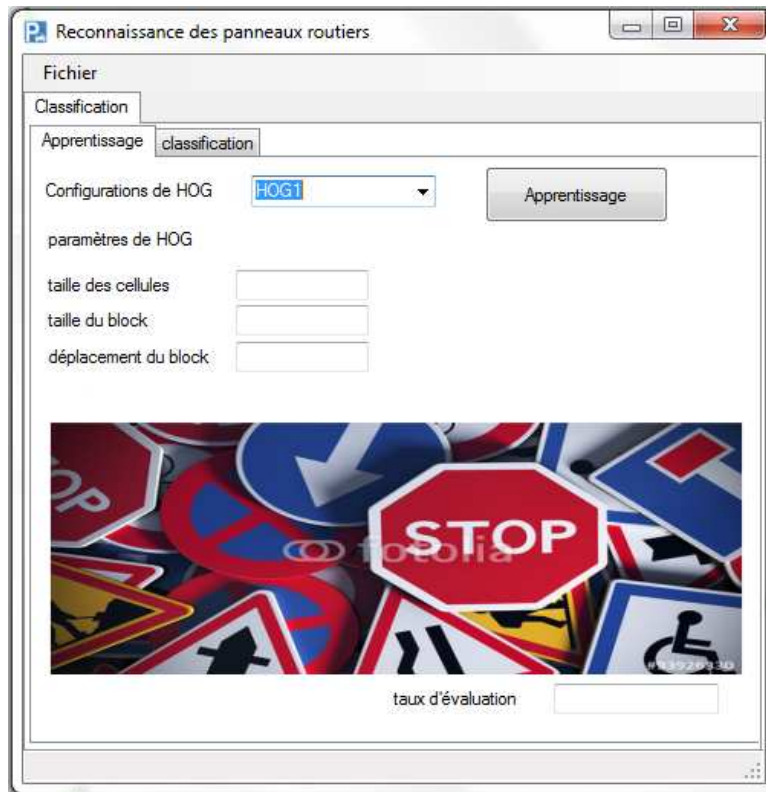


Figure 26. Le choix de la première configuration de HOG

Après avoir sélectionné la configuration, les paramètres de HOG sont affichés. Comme pour HOG1, la taille des cellules été de 8 x 8, la taille du block été de 16 x 16, déplacement est de 8 x 8 (Cf. **Figure 27**).



Figure 27. L'affichage des paramètres de HOG1

Pour lancer l'apprentissage, en appui sur le bouton « apprentissage », alors cinq fenêtres s'affichent, l'une après l'autre pour le choix des données d'apprentissage qui sont des images représentant des panneaux routiers, organisées dans des répertoires (Cf. Figure 28).

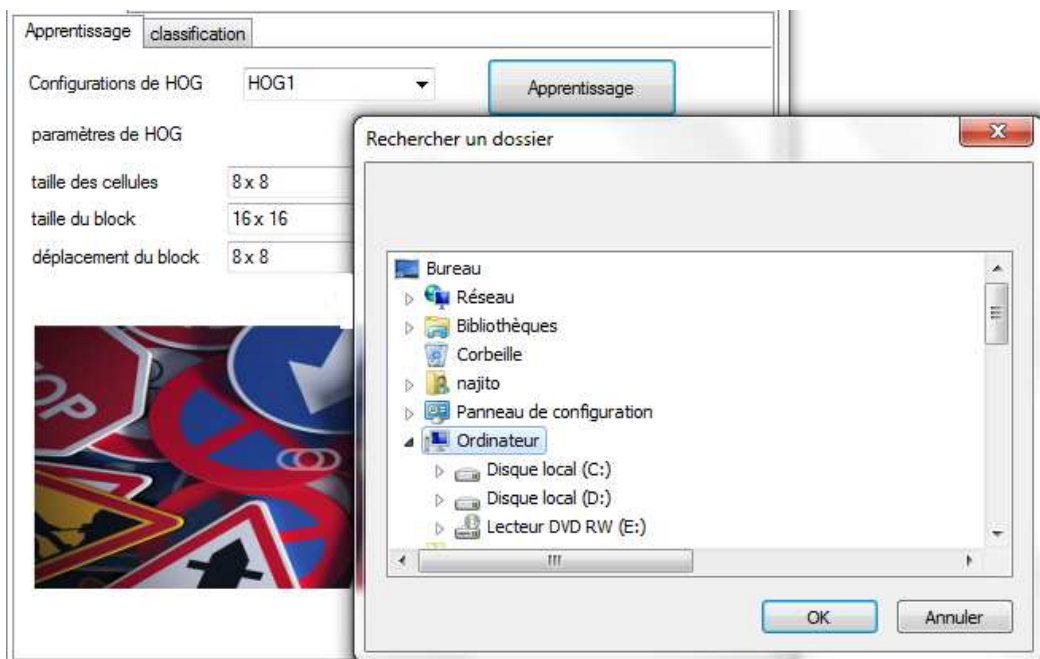


Figure 28. Le choix des images de l'apprentissage

Le répertoire « 00002 » contient des panneaux de 50 Km/h (Cf.figure 29).

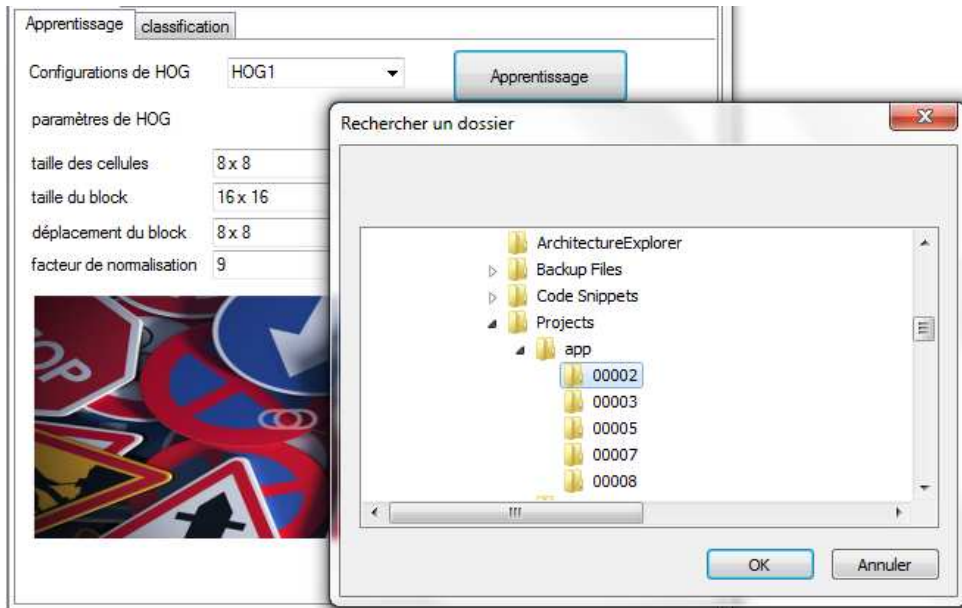


Figure 29. La sélection du répertoire des panneaux de 50 Km/h

Le répertoire « 00003 » contient des panneaux de 60 Km/h (Cf.figure 30).

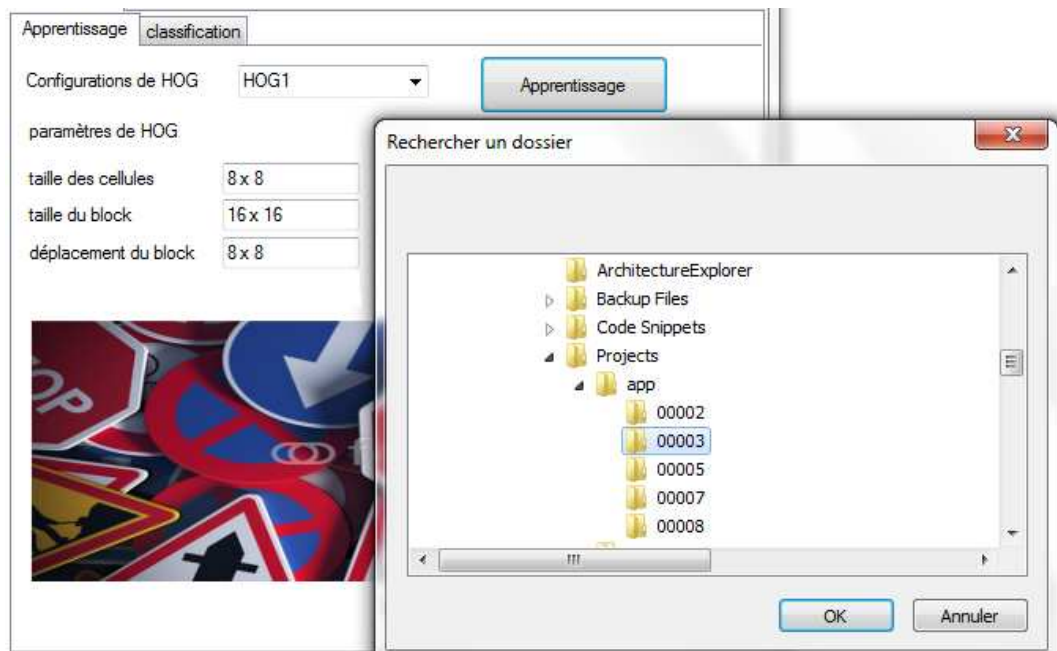


Figure 30. La sélection du répertoire des panneaux de 60 Km/h

Le répertoire « 00005 » contient des panneaux de 80 Km/h (Cf.figure 31).

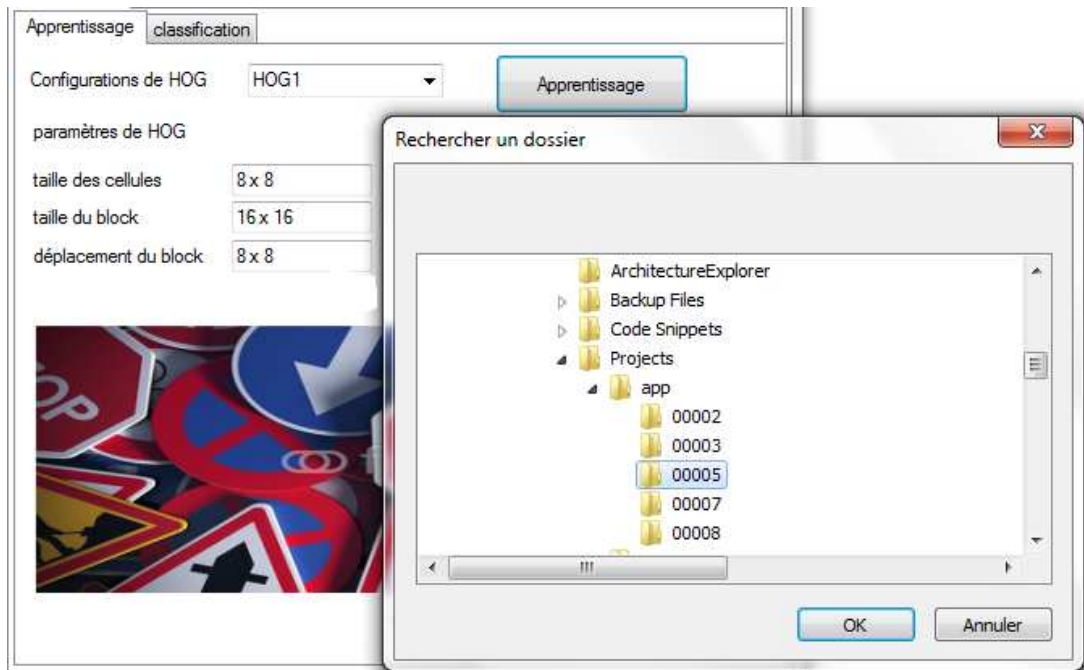


Figure 31. La sélection du répertoire des panneaux de 80 Km/h

Le répertoire « 00007 » contient des panneaux de 100 Km/h (Cf.figure 32).

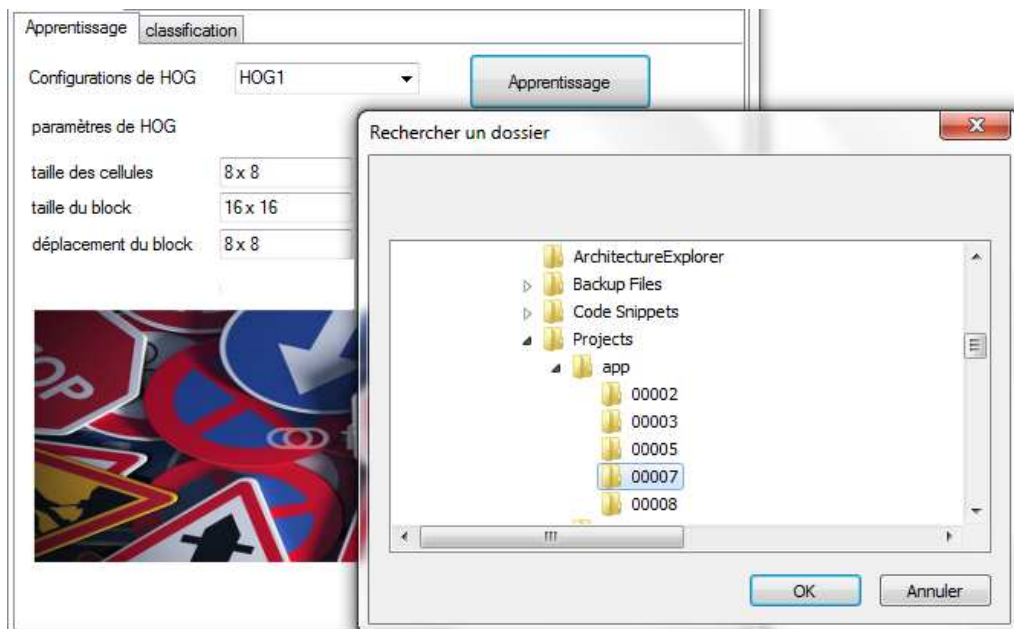


Figure 32. La sélection du répertoire des panneaux de 100 Km/h

Le répertoire « 00008 » contient des panneaux de 120 Km/h (Cf.figure 33).

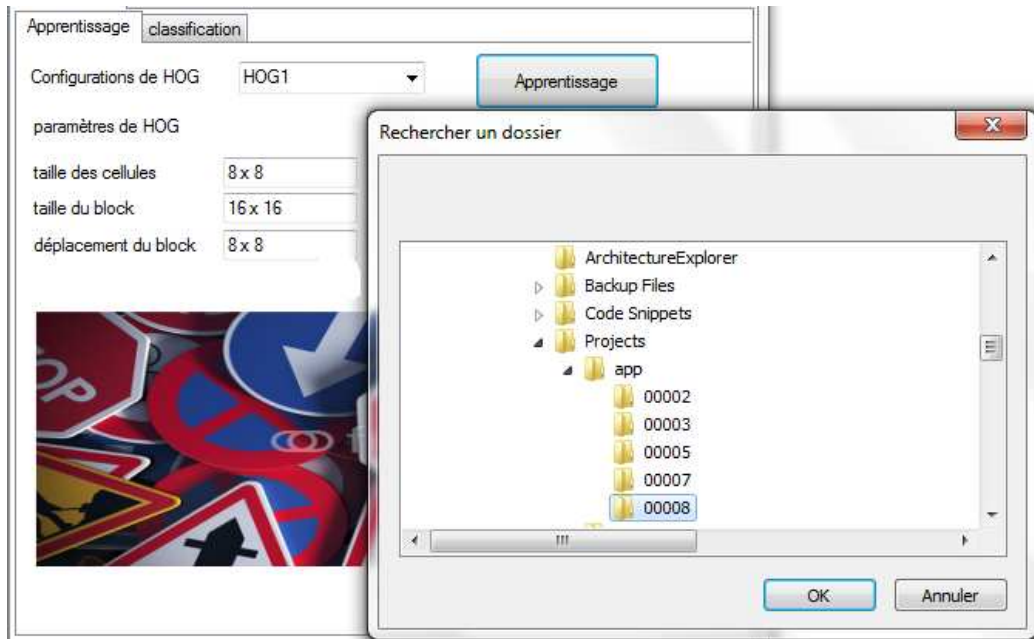


Figure 33. La sélection du répertoire des panneaux de 120 Km/h

Après le choix des répertoires, et la lecture des images, on extrait les caractéristiques de HOG selon les paramètres fixés. Une partie de ces descripteurs sera utilisée pour l'apprentissage des classifieurs et le reste pour leur évaluation (Cf. **Figure 34**).

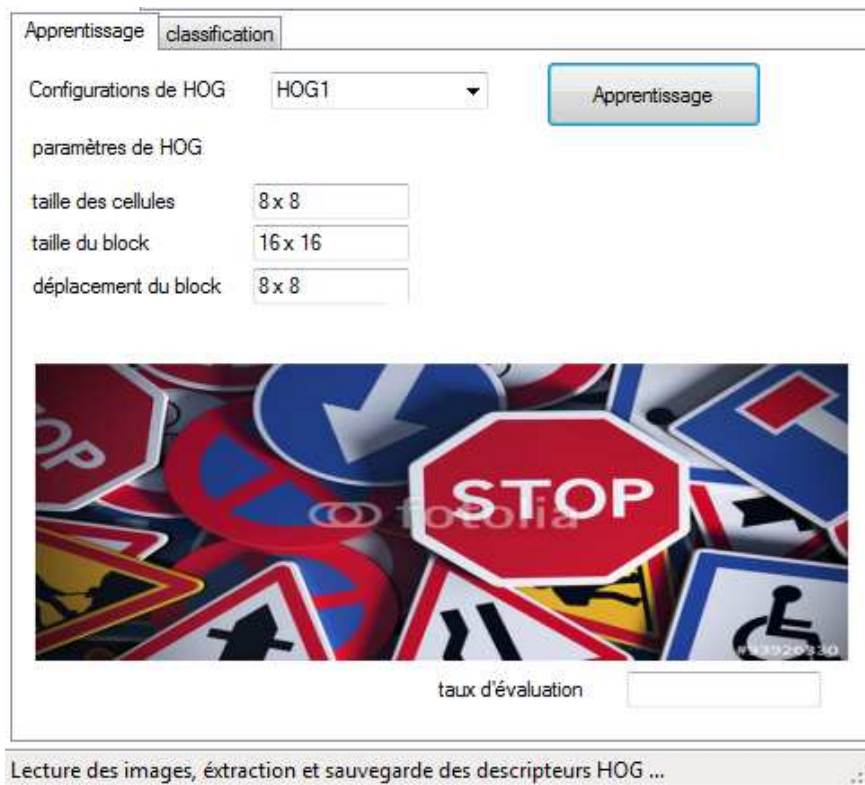


Figure 34. Lecture des images, extraction et sauvegarde des descripteurs de HOG

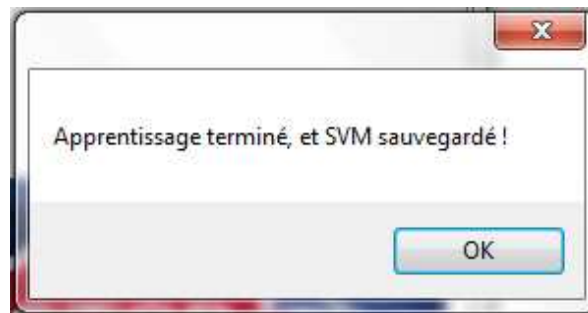


Figure 35. Fenêtre d'information : SVM sauvegardé

Dans le but de préciser la bonne configuration de HOG et de vérifier si les paramètres choisis pour SVM sont acceptables ou pas, on a calculé le taux d'évaluation pour chaque configuration.

Tel que la première configuration de HOG donne un taux d'évaluation de 0.97 (Cf. **figure 36**).

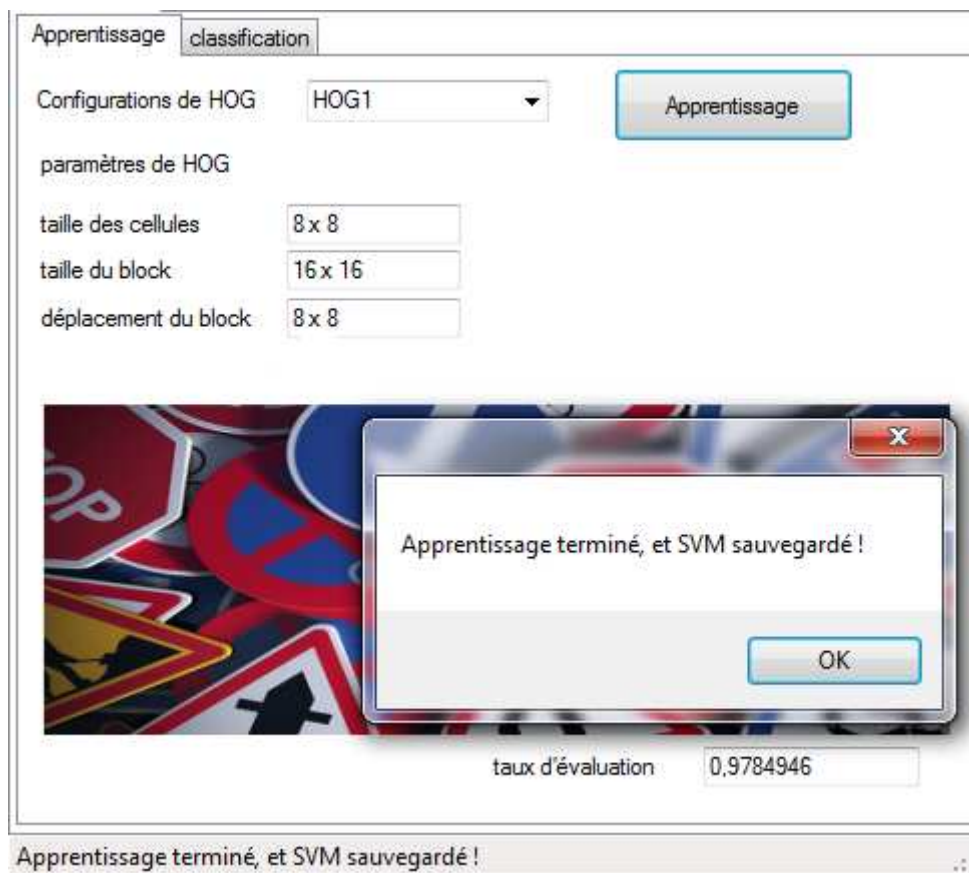


Figure 36. Affichage du taux d'évaluation pour HOG1

Le taux d'évaluation de la deuxième et la troisième configuration été de 0.98, tel que les paramètres sont montré dans les figures (Cf. **Figure 37,38**).

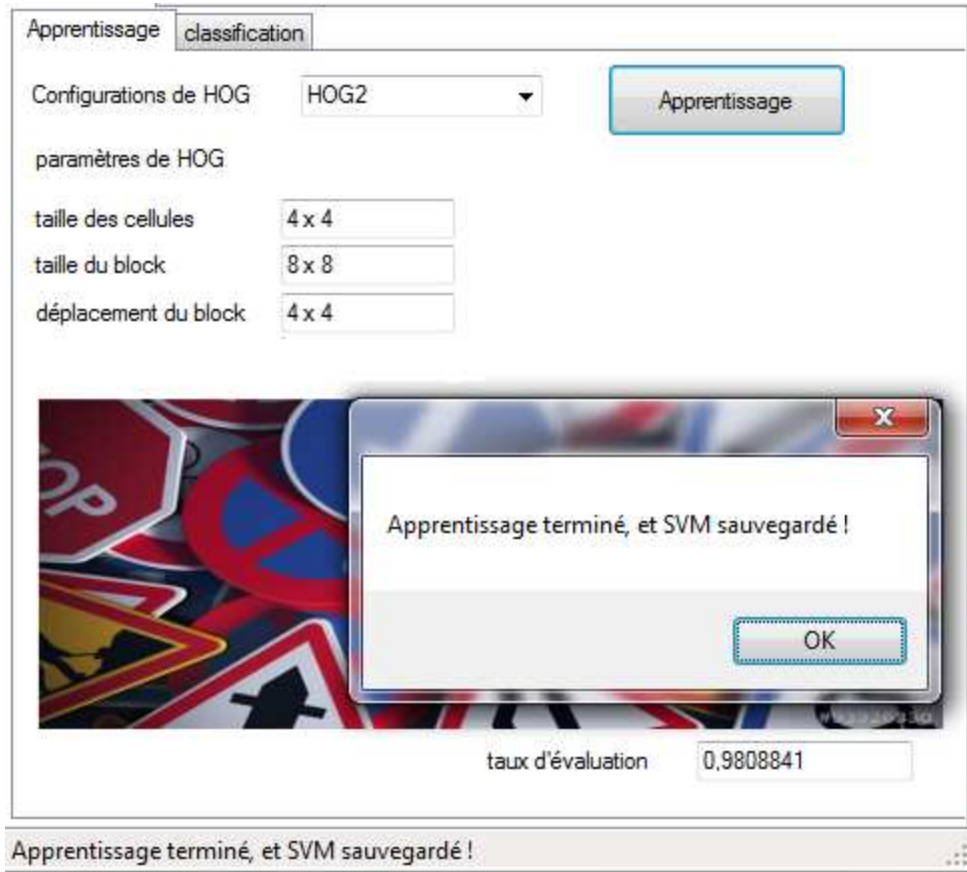


Figure 37. Affichage du taux d'évaluation pour HOG2

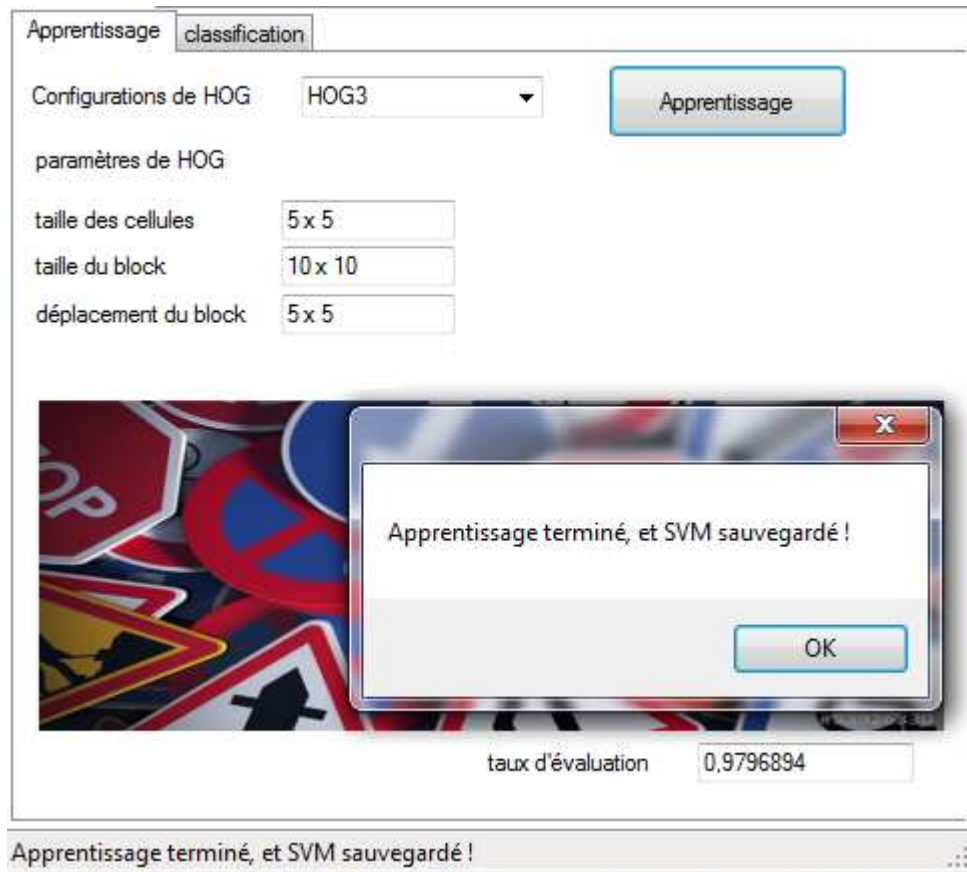


Figure 38. Affichage du taux d'évaluation pour HOG3

III.11.2. la classification

Après l'apprentissage et l'évaluation, vient la validation.

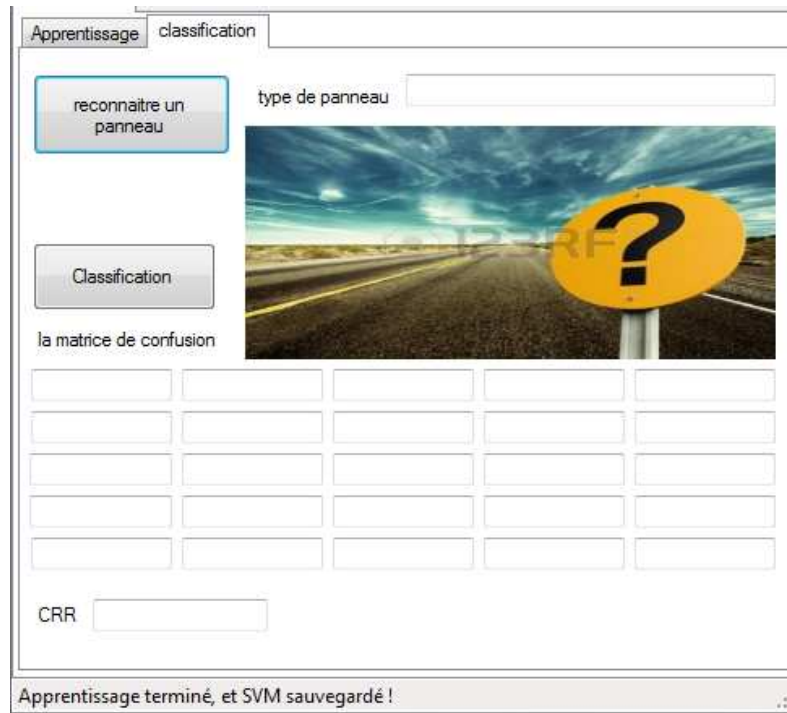


Figure 39. La deuxième fonctionnalité de l'application : la classification

Dans cette partie, on a deux choix :

Reconnaitre une image : en appuyant sur ce bouton on peut choisir une image pour que le classifieur nous donne sa classe (50/60/80/100/120 Km/h) ou (inconnu) (Cf. **Figure 40**).

Et ça, après l'extraction de ses caractéristiques en utilisant les paramètres de HOG choisis dans la phase d'apprentissage.

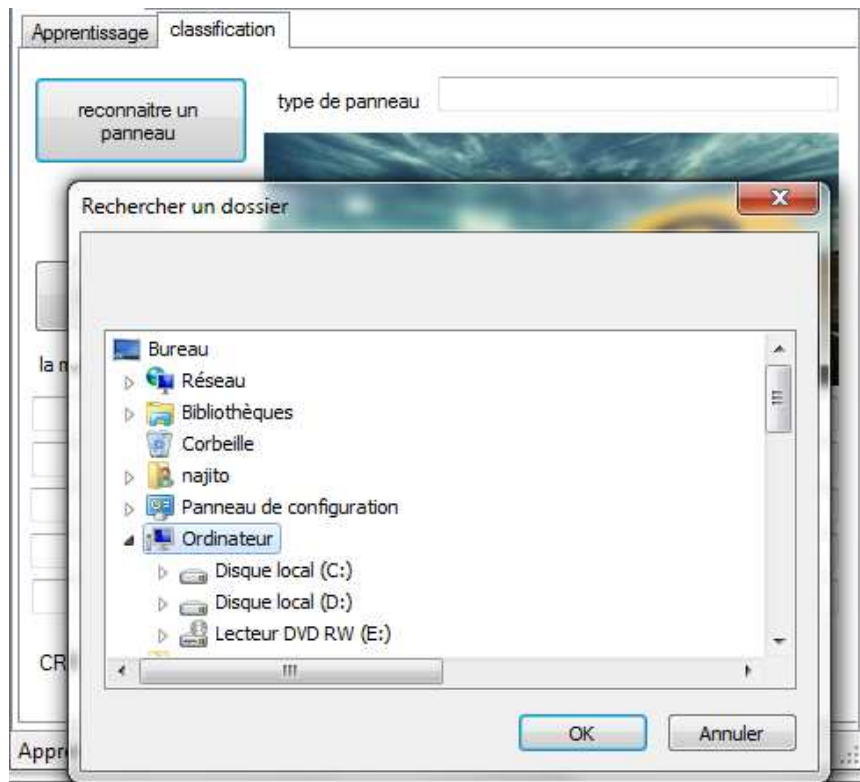


Figure 40. Choix de l'image du panneau à classifier

Des panneaux bien prédits :



Figure 41. Un panneau de 120 Km/h reconnu

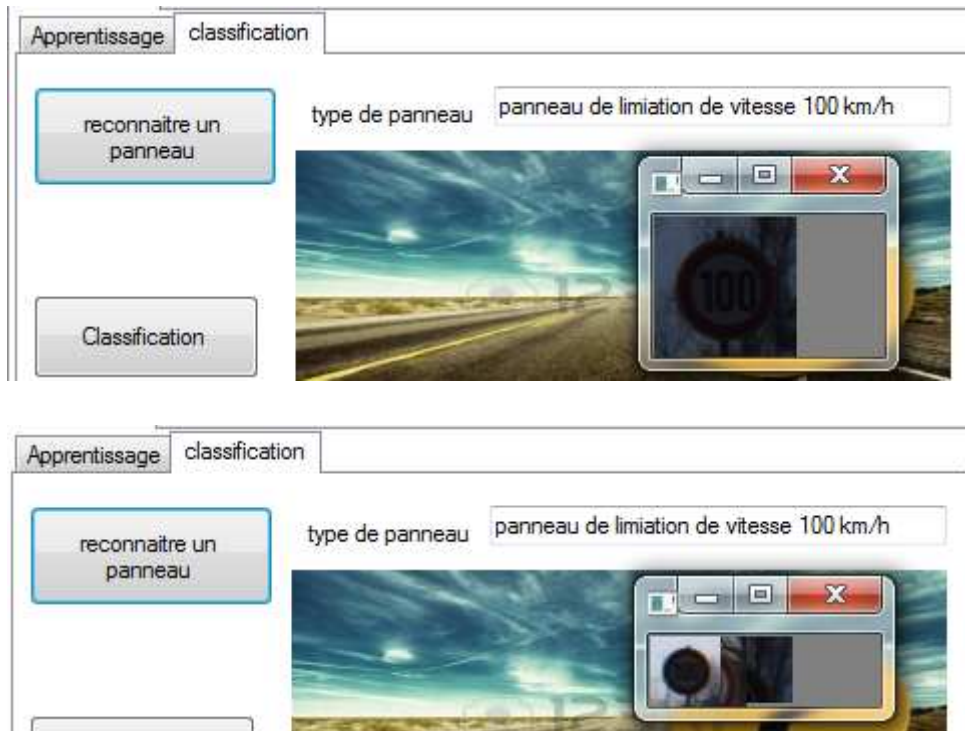


Figure 42. Un panneau de 100 Km/h reconnu



Figure 43. Un panneau de 50 Km/h reconnu



Figure 44. Un panneau de 60 Km/h reconnu





Figure 45. Un panneau de 80 Km/h reconnu

Des panneaux non reconnu :



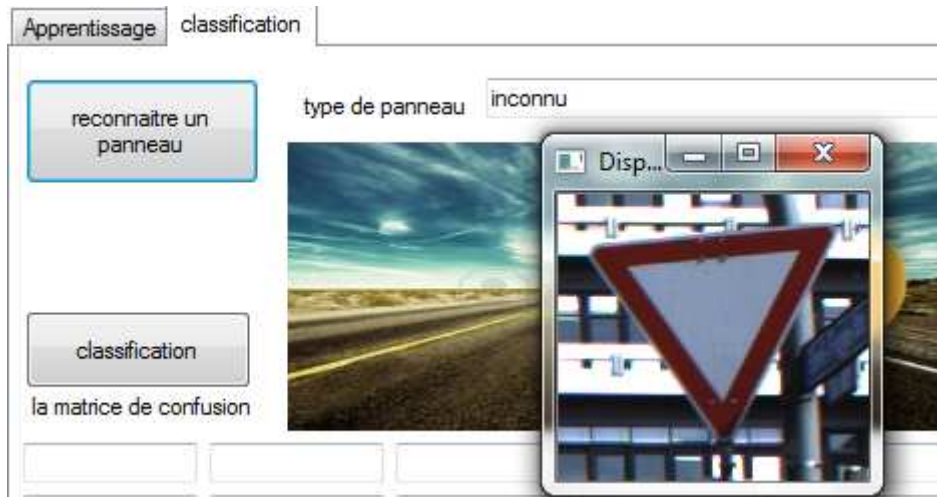




Figure 46. Exemple des panneaux non reconnu par notre système

Des panneaux mal classé :

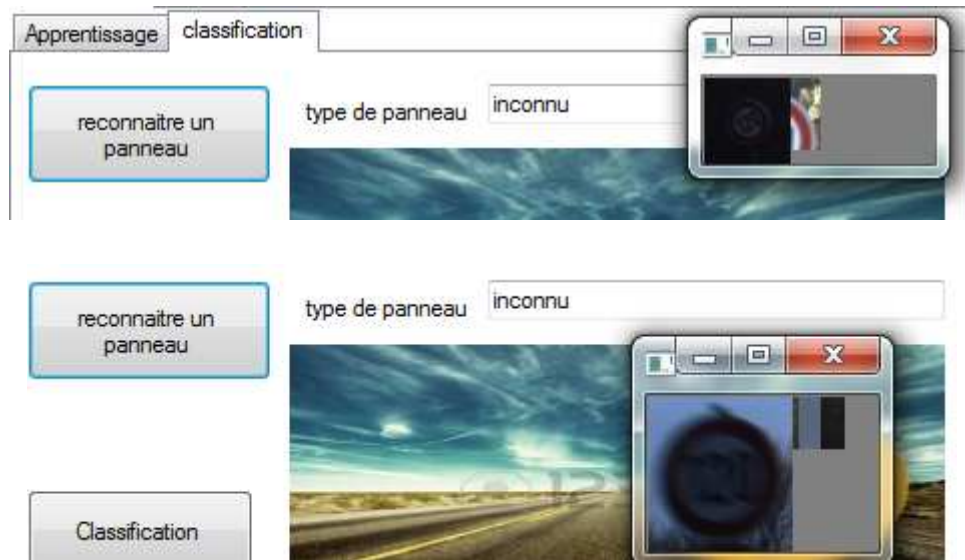


Figure 47. Exemple d'un panneau de 120 Km/h non reconnu par notre système

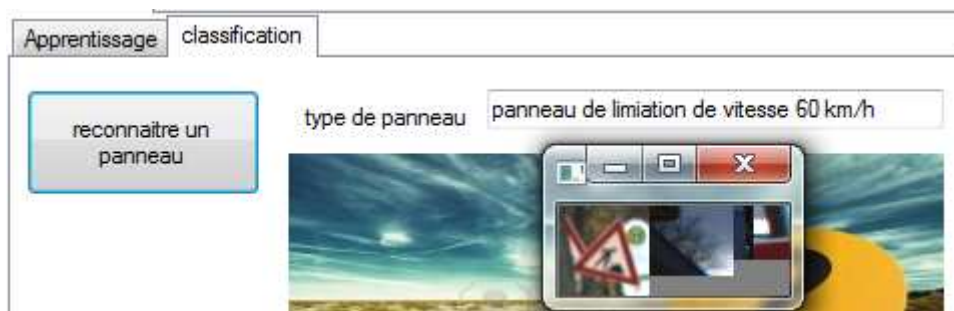


Figure 48. Exemple d'un panneau inconnu, classifié comme panneau de 60 Km/h



Figure 49. Exemple d'un panneau inconnu, classifié comme panneau de 50 Km/h



Figure 50. Exemple d'un panneau inconnu, classifié comme panneau de 80 Km/h

La validation : nous avons calculé la matrice de confusion pour qu'on puisse calculer le CRR pour chaque configuration de HOG.

Donc en appuyant sur le bouton « **classification** » une fenêtre va être affichée pour le choix de la base de validation.

Les résultats obtenus sont présentés dans les figures **22**, **23** et **24** (dans la section **III.10**).

III.12. Conclusion

Ce chapitre a été consacré à la description de notre conception et la mise en œuvre de notre système de reconnaissance des panneaux routiers.

Notre application consiste à montrer aux utilisateurs la manière de faciliter la tâche de reconnaître un panneau, tout en tenant compte de la qualité des résultats obtenus.

Cette application est développée en C++ sous Visual Studio 2013.

Conclusion générale

Dans ce mémoire nous avons présenté la deuxième étape d'un système de reconnaissance de panneaux routiers, qui est basé principalement sur deux étapes principales, l'extraction des descripteurs et leur classification.

Pour l'extraction des descripteurs des panneaux de la base GTSRB, on a utilisé les descripteurs de HOG, ce dernier a été attribué à sa classe en utilisant SVM.

L'utilisation de HOG avec SVM nous a permis d'obtenir des bons résultats vu le taux d'évaluation obtenu, et cela confirme l'efficacité de HOG dans ce domaine.

Finalement, la reconnaissance des panneaux routiers est un domaine vaste, et pour cela beaucoup d'applications sont développés.

Bibliographie

- [1] N Salameh, « Conception d'un système d'alerte embarqué basé sur les communications entre véhicules », Thèse de doctorat, Institut National des Sciences Appliquées de Rouen 2011.
- [2] R Bishop et Ebooks Corporation, « Intelligent vehicle technology and trends » Artech House, 2005.
- [3] M Boumediene, Jean-Philippe Laufenburger, J Daniel, C Cudel, « Détection, association et suivi de pistes pour la reconnaissance de panneaux routiers », Rencontres francophones sur la Logique Floues et ses Applications, Oct 2014, Cargèse, France. <hal-01123472>
- [4] Mohammed Boumediene, « Contributions aux systèmes d'aide à la conduite automobile », thèse en vue de l'obtention du diplôme du doctorat en science, Université des Sciences et de la Technologie d'Oran « Mohamed Boudiaf », 2015.
- [5] S.K Berkaya et al, « On circular traffic sign detection and recognition, Expert Systems With Applications » 2015, <http://dx.doi.org/10.1016/j.eswa.2015.11.018>
- [6] F Zaklouta, « Multiclass Object Recognition for Driving Assistance Systems and Video Surveillance », Thèse de doctorat, Ecole des Mines de Paris 2011.
- [7] A Cord, « Reconnaissance d'objets routiers – Application. Perception pour la conduite automatisée », support du cour, 10/10/2012.
- [8] D.G Lowe, « Distinctive Image Features from Scale-Invariant Keypoints », International Journal of Computer Vision 60(2), pp. 91–110, 2004.
- [9] B Soheilian, A Arlicot, N Paparoditis, « Extraction de panneaux de signalisation routière dans des images couleurs », Institut Géographique National, Laboratoire MATIS IGN, MATIS, 73, avenue de Paris, 94165 Saint-Mandé cedex, France.
- [10] N Dalal et N Triggs, « Histograms of Oriented Gradients for Human Detection ». In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 886-893, San Diego, CA, USA, 2015.
- [11] F Zaklouta, B Stanciulescu, O Hamdoun, « Traffic Sign Classification using K-d trees and Random Forests », In International Joint Conference on Neural Networks, 2011.
- [12] F Zaklouta et B Stanciulescu, « Real-time traffic sign recognition in three stages ». Robotics and Autonomous Systems, 62(1):16-24, 2014.
- [13] A. Adam, C. Ioannidis, « Automatic road-sign detection and classification based on support vector machines and HOG descriptors ». ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume II-5, 2014. ISPRS Technical Commission V Symposium, 23 – 25 June 2014, Riva del Garda, Italy
- [14] <http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>
(consulté le 21/05/2016 à 17 :00)
- [15] Ricco Rakotomalala, Apprentissage supervisé, support du cour, université lyon2, http://eric.univ-lyon2.fr/~ricco/cours/slides/Apprentissage_Supervise.pdf
- [16] Anil K, Jain and Richard C, Dubes, Algorithms for clustering data, prectice-Hall PTR, 1998.
- [17] Robert E, Schapire, The strength of weak learnability, Machine learning, 5(2): 197-227, june 1990.

- [18] W. S McCulloch, and Pitts, «A Logical Calculus of the Ideas Immanent in Nervous Activity». Bulletin of Mathematical Biophysics 5(4), pp. 115–133. 1943.
- [19] B Krose and P van der Smagt, «Introduction to Neural Networks». 1996.
- [20] F Rosenblatt, «Principles on Neuro dynamics». Spartan Books. 1959.
- [21] D. E Rumelhart, G. E Hinton, and R. J Williams. «Learning Representations by Back-Propagating Errors». Nature 323(6088), pp. 533–536. 1986.
- [22] Alberto Broggi, Pietro Cerri, Paolo Medici, Pier Paolo Porta, VisLab – Dipartimento di Ingegneria dell’Informazione Università degli Studi di Parma, ITALY. Proceedings of the 2007 IEEE Intelligent Vehicles Symposium.
- [23] De la escalera et al, «road traffic sign detection and classification». IEEE transactions on industrial electronics, vol 44, n°6, decembre 1997.
- [24] Fabien MOUTARDE, «apprentissage statistiques et fouille de données pour l’analyse temps réel d’images et les systèmes de transport intelligents ». Mémoire présenté à l’université Pierre et Marie Curie (Paris 6) pour obtenir le diplôme d’habilitation à diriger des recherches. 2013.
- [25] Weighted k-nearest-neighbor techniques and ordinal classification. Sonderforschungsbereich 386, paper 399.
- [26] M Mathias, R Timofte, R Benenson, and L.V Gool, «Traffic Sign Recognition-How far are we from the solution?»
- [27] C. J. C Burges, A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2(2):121-167. 1998.
- [28] B.Wali Safat, A. Hannan Mahammad, Aini Hussain, and A. Samad Salina, «An Automatic Traffic Sign Detection and Recognition System Based on Colour Segmentation, Shape Matching, and SVM». Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2015, Article ID 250461, 11 pages. <http://dx.doi.org/10.1155/2015/250461>
- [29] J. R Quinlan, «Induction of Decision Trees». Machine Learning 1(1), pp. 81–106. 1986
- [30] L Breiman, «Random Forests». Machine Learning 45(1), pp. 5–32. 2001.