
RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEURE ET DE LA RECHERCHE SCIENTIFIQUE
CENTRE UNIVERSITAIRE BELHADJ BOUCHAIB D'AÏN-TÉMOUCHENT



Institut des Sciences
Département de Mathématiques et de l'Informatique

Mémoire

Pour l'obtention du Diplôme de Master en Informatique
Option : Réseaux et Ingénierie des Données (RID)

Présenté par :

Mme. Nourria KEDDAR
Mme. Leila AHMED BELKACEM

DÉTECTION ET RECONNAISSANCE DES PANNEAUX DE SIGNALISATION ROUTIÈRE

Encadrant :

Mr. Djalal MERAD BOUDIA
Maitre Assistant "A" au C.U.B.B.A.T.

Soutenu en 2019

Devant le jury composé de :

Président : Mme. Fatima Zohra BELGRANA (M.C.B) C.U.B.B.A.T.

Examineur : Mme. Karima BERRAMLA (M.A.A) C.U.B.B.A.T.

Encadrant : Mr. Djalal MERAD BOUDIA (M.A.A) C.U.B.B.A.T.

Remerciements

Nous adressons en premier lieu notre reconnaissance à notre "DIEU" tout puissant, de nous avoir permis d'arriver là, car sans lui rien n'est possible.

Nous tenons tout d'abord à remercier Mr Djalal MERAD BOUDIA notre encadreur de mémoire, pour son encadrement, son suivi et ses conseils lors de la réalisation de notre mémoire.

Nous tenons ensuite à remercier tous nos professeurs et enseignants de département d'informatique qui ont contribué à notre formation.

Nous souhaitons aussi remercier mes dames :Fatima Zohra BELGRANA,Karima BERRAMLA,les membres du jury pour leur précieux temps accordé à l'étude de notre mémoire.

Enfin nous remercions tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail.

Résumé

Afin de faciliter le processus de conduite et d'éviter les accidents de la route, qui sont devenus incriminables ces derniers temps, et qui sont devenus dangereux pour la vie humaine en général et pour la vie des enfants en particulier, donc un être humain fait une grande attention dans la recherche d'une solution appropriée pour sortir de ce problème ou au moins de le minimiser. Alors la solution est de franchir le domaine de la vision par ordinateur que nous étudions et utilisons pour identifier ce type d'images.

Notre travail consiste à implémenter une application bureau permettant d'extraire des panneaux de signalisation à partir d'une image. Après une étape de prétraitement, expliquée en détail dans le chapitre "Conception et Développement", nous utilisons l'approche top-down qui exige de sélectionner, en premier lieu, des zones susceptibles d'être des zones d'intérêt, ensuite, de focaliser notre traitement à ces zones extraites afin de valider s'il s'agit vraiment d'un panneau de signalisation. L'approche top-down nous évite de traiter une large zone de l'image, et par conséquent un gain de temps d'exécution. Ce qui implique des meilleures performances de l'application. Une autre phase prend place après ces traitements pour identifier un panneau en considérant une base de panneaux de signalisation. L'identification d'un panneau de signalisation est traduite par l'affichage de l'image correspondante de la base, ou bien d'afficher un texte expliquant la plaque ou même encore d'émettre un son indiquant de quel panneau il s'agit.

Mot clé

Traitement d'image, top-down, Pixel, OpenCV, SURF, SIFT, RVB, HSV, RSR, ROI, Panneaux de Signalisation Routière.

Abstract

In order to facilitate the driving process and to avoid road accidents, which have become incremental in recent times, and which have become dangerous for human life in general and for the lives of children in particular, so a human being does great attention in finding an appropriate solution to get out of this problem or at least to minimize it. So the solution is to cross the field of computer vision that we study and use to identify this type of images.

Our job is to implement a desktop application to extract signs from an image. After a pretreatment step, explained in detail in the "Conception and development" chapter, we use the top-down approach which requires to select, first, areas that may be areas of interest, then to focus our treatment to these extracted areas to validate if it is really a road sign. The top-down approach avoids us to treat a large area of the image, and consequently a saving of time of execution. Which implies better performance of the application. Another phase takes place after these treatments to identify a panel by considering a base of road signs. The identification of a sign is translated by the display of the corresponding image of the base, or to display a text explaining the plate or even to emit a sound indicating which panel it is .

Keywords

Image processing, top-down, Pixel, OpenCV, SURF, SIFT, RGB, HSV, RSR, ROI, Road Signs.

Table des matières

Remerciements

Résumé

Abstract

Introduction générale **viii**

1 Généralités sur le traitement d'images	1
1.1 Introduction	3
1.2 Définition de l'image	3
1.3 Image numérique :(numérisée)	4
1.4 Caractéristiques d'une image numérique	4
1.4.1 Pixel	4
1.4.2 La résolution	5
1.4.3 Dimension	6
1.4.4 La texture	6
1.4.5 Bruit	7
1.4.6 La luminance	7
1.4.7 Histogramme	8
1.4.8 Le contraste	9
1.5 Types d'images	9
1.5.1 Images matricielles	9
1.5.2 Images vectorielles	9
1.6 Codages des couleurs	10
1.6.1 Image noir et blanc	10
1.6.2 Niveaux de gris	11
1.6.3 Image couleur	11

1.7	Formats d'image	13
1.8	Le système de traitement d'images	13
1.8.1	Acquisition	14
1.8.2	Filtrage	14
1.8.3	Segmentation	16
1.9	Domaines D'application	17
1.9.1	Imagerie aérienne et spatiale	18
1.9.2	Technologies biomédicales	18
1.9.3	La robotique	18
1.9.4	la télésurveillance	18
1.9.5	Simulation et contrôle de processus	18
1.10	Conclusion	18
2	Outil de développement	19
2.1	Introduction	20
2.2	Présentation de la librairie OpenCV	20
2.2.1	Historique	20
2.2.2	Qu'est -ce que OpenCV	21
2.2.3	Fonctionnalités	21
2.2.4	Modules OpenCV	22
2.3	Installation et configuration OpenCV2.4.13 avec Netbeans 8.2	23
2.3.1	Téléchargement et extraction OpenCV	23
2.3.2	Netbeans	24
2.3.3	Configuration du OpenCV dans Netbeans 8.2	24
2.3.4	Mise en place variables d'environnement	30
2.4	Conclusion	32
3	Conception et Développement	33
3.1	Introduction	34
3.2	Etat de l'art	34
3.2.1	Système RSR intégré dans les voitures	35
3.2.2	Applications mobiles avec RSR	36
3.2.3	Travaux de recherche	36
3.3	Les panneaux de signalisation routière	38
3.3.1	Les types des panneaux de signalisation routière	39
3.4	Mise en œuvre (Conception et développement du projet)	42
3.4.1	Description de l'approche	42
3.4.2	Le travaille avec OpenCV	43

TABLE DES MATIÈRES

3.4.3	Etape de prétraitement de l'image et de détection	43
3.4.4	Etape de reconnaissance	55
3.5	Conclusion	65
	Conclusion et perspectives	66
	Abréviation	68
	Bibliographie	68

Table des figures

1.1	Centre Universitaire - BELHADJ BOUCHAIB - Ain Témouchent . . .	3
1.2	Représentation d'image numérique[1]	4
1.3	Représentation en groupe de Pixel d'une image du C.U.B.B.A.T . . .	5
1.4	Représentation de Résolution d'une image.	5
1.5	Représentation de dimension d'une image	6
1.6	Image sans bruit	7
1.7	Image avec bruit	7
1.8	Image représente l'effet de luminance	8
1.9	Image du "C.U.B.B.A.T" avec histogramme	8
1.10	Comparaison du volume d'une image "C.U.B.B.A.T" pour différents codages	10
1.11	Image"C.U.B.B.A.T" niveaux de gris	11
1.12	Principe codage de la couleur[2]	12
1.13	Valeurs RVB d'un pixel "C.U.B.B.A.T"	12
1.14	Principaux formats d'image[2]	13
1.15	Schéma d'un système de traitement d'images[3]	13
1.16	approche région et approche contour[4]	17
2.1	Choix le chemin d'extraction d'OpenCV	23
2.2	Extraction d'OpenCV	24
2.3	Création d'un nouveau projet	25
2.4	Création d'un nouveau projet	25
2.5	Création d'un nouveau projet	26
2.6	L'ajoute de la Bibliothèque OpenCV	26
2.7	L'ajoute de la Bibliothèque OpenCV	27
2.8	L'ajoute de la Bibliothèque OpenCV	27
2.9	L'ajoute de la Bibliothèque OpenCV	28

2.10	L'ajoute de la Bibliothèque OpenCV	28
2.11	L'ajoute de la Bibliothèque OpenCV	29
2.12	L'ajoute du chemin OpenCV	29
2.13	L'ajoute du chemin OpenCV	30
2.14	Propriétés du système	30
2.15	Propriétés du système	31
2.16	Propriétés du système	31
3.1	Exemple d'application RSR dans une voiture de la marque Opel . . .	35
3.2	Application mobile myDriveAssist	36
3.3	Exemples des dangers pouvant être signalés grâce aux signaux routière	38
3.4	Exemples des indications fournies par les panneaux sur les directions à suivre	38
3.5	Exemples des informations liées à l'usage de la route que les panneaux peuvent indiquer	39
3.6	Signaux d'interdiction ou de restriction	39
3.7	Signaux d'avertissement de danger	40
3.8	Signaux de priorité	40
3.9	Signaux de fin d'interdiction ou de restriction	41
3.10	Signaux d'obligations	41
3.11	Schéma de description de la proche	42
3.12	Exemple de filtre médian	44
3.13	Exemple de filtre gaussien	44
3.14	Espace colorimétrique HSV[5]	45
3.15	Image entrée en RVB	46
3.16	Image sortie en HSV	46
3.17	Exemple de segmentation bleu et rouge	48
3.18	Détection des objets rouges qui ne présente pas forcément des panneaux	49
3.19	Exemple des cercles bleu et rouge détecté	50
3.20	Image binarisée	51
3.21	Image binarisée sans et avec filtrage gaussien	52
3.22	Exemple de détection des contours de formes déférent	53
3.23	Exemple de détection des faux contours bleu et rouge	54
3.24	Exemple de points d'intérêts détecté	56
3.25	Organigramme de détection, d'extraction et de comparaison des ca- ractéristiques	58
3.26	Comparaison des caractéristiques	59
3.27	Diagramme UML de notre Application	59

TABLE DES FIGURES

3.28	Panneaux similaire	61
3.29	Cas des obstacles	61
3.30	Page de garde	62
3.31	Page d'authentification	62
3.32	Page d'accueil	63
3.33	Roconnaissance Rouge	63
3.34	Roconnaissance Bleu	64
3.35	Roconnaissance avec texte	64

Liste des tableaux

3.1	Description du matériel utilisé pour les tests	60
3.2	Résultats des tests avec SURF	60
3.3	Résultats des tests avec SIFT	60
3.4	Temps d'exécution avec SURF et SIFT	61

Introduction Générale

Ces derniers temps, le domaine de traitement d'image est un domaine très vaste puisque il est intégré à plusieurs disciplines par exemple la sécurité d'information, caméras de surveillance, et le domaine de RSR (Reconnaissance de la Signalisation Routière) qui est un nouvel outil performant qui détecte, localise et reconnaît automatiquement les panneaux de signalisation routière à partir d'images. Ce dernier constitue le cadre de travail de notre projet de fin d'étude.

Alors, Un système de reconnaissance automatique des panneaux de signalisation peut être mis en œuvre sur l'automobile dans le but de détecter et de reconnaître tous les panneaux de signalisation confrontés.

Donc les moyens de transport actuel sont très développés puisque ils facilitent la vie des personnes et c'est ça le rôle de l'informatique, alors les dernières marques des voitures intègrent cette technologie donc on peut nommer ces voitures par des voitures intelligente puisque ils ont reconnus les plaques des routes automatiquement.

Ensuite depuis longtemps, les personnes handicapés sont démunis et ont de la difficulté à conduire les voitures, maintenant ils peuvent facilement conduire les voitures et le grands merci au développement continu dans ce domaine, ce dernier utilisent les caméras pour reconnaître les panneaux de signalisation routière et aussi pour le stationnement.

Le choix de notre thème à une grande relation avec cette technologie, il a pour but de concevoir et développer une application qui peut reconnaître les différents panneaux de signalisation routière.

Contexte et motivation du projet

L'application en question s'agit d'une application de traitement d'image de reconnaissance des panneaux de route pour avoir une solution au voiture afin de faciliter la conduite au personnes et augmenter la sécurité aux vie. Aussi, la tendance actuelle de notre économie va vers la fabrication des voitures après une période de montage. Ainsi, notre application peut être exploitée dans la partie soft d'une voiture Algérienne.

Approche de Solution

En tenant compte des critiques et des besoins d'informatiser les services cités ci-dessus, la solution est de concevoir et développer une application de reconnaissance des panneaux de route permettant de satisfaire au maximum possible le client.

Pour cela l'application doit répondre aux besoins suivants :

- Avoir un logiciel qui respecte les principes des Interfaces Homme/Machine (IHM) tels que l'ergonomie et la fiabilité.
- Réduire les tâches manuelles qui nous permettraient de gagner en spatio-temporel.
- Gagner en spatio-temporel.

Organisation du rapport

L'objet du mémoire est de fournir les différentes informations liées au contexte et au déroulement du projet .Il est organisé et composé de deux parties :

Partie I :

La première partie contient presque toutes les informations essentielles pour faire un démarrage dans le domaine du traitement d'image, donc cette partie est composée de deux chapitres.

— Chapitre I : Généralités sur le traitement d'images.

— Chapitre II : Outils de développements.

Partie II :

La deuxième partie composée d'un seul chapitre Mise en œuvre du projet qui est composé d'une seule partie :

— Chapitre III : Conception et Développement.

Généralités sur le traitement d'images

Sommaire

1.1	Introduction	3
1.2	Définition de l'image	3
1.3	Image numérique :(numérisée)	4
1.4	Caractéristiques d'une image numérique	4
1.4.1	Pixel	4
1.4.2	La résolution	5
1.4.3	Dimension	6
1.4.4	La texture	6
1.4.5	Bruit	7
1.4.6	La luminance	7
1.4.7	Histogramme	8
1.4.8	Le contraste	9
1.5	Types d'images	9
1.5.1	Images matricielles	9
1.5.2	Images vectorielles	9
1.6	Codages des couleurs	10
1.6.1	Image noir et blanc	10
1.6.2	Niveaux de gris	11
1.6.3	Image couleur	11
1.7	Formats d'image	13
1.8	Le système de traitement d'images	13
1.8.1	Acquisition	14
1.8.2	Filtrage	14
1.8.3	Segmentation	16

1.9 Domaines D'application	17
1.9.1 Imagerie aérienne et spatiale	18
1.9.2 Technologies biomédicales	18
1.9.3 La robotique	18
1.9.4 la télésurveillance	18
1.9.5 Simulation et contrôle de processus	18
1.10 Conclusion	18

1.1 Introduction

Le traitement d'image est un domaine très large qui a connu, et qui a assisté un développement important depuis quelques dizaines d'années[3].

Alors, le traitement d'image est l'ensemble des techniques qui sont appliqué à l'image numérique pour améliorer ou d'en extraire des informations[3].

Donc, dans ce chapitre, nous abordons les notions de base nécessaires et des techniques de traitement d'images. Tels que : la définition d'image, les types et caractéristiques d'image, les opérations ou les traitements possibles appliquer sur l'image et enfin le domaine d'application.

1.2 Définition de l'image

Une image est une représentation visuelle[6] de description d'un objet ou personne par dessin, peinture, photographie, films,... etc. C'est aussi derrière l'affichage sur écran c'est un ensemble organisé d'informations, acquise, créée, traitée ou stockée sous forme binaire (suite de 0 et de 1). Ce qui nécessite sa numérisation [7][2].



FIGURE 1.1 – Centre Universitaire - BELHADJ BOUCHAIB - Ain Témouchent

1.3 Image numérique :(numérisée)

L'image numérique est l'image dont la surface est divisée en éléments de taille fixe appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs.

La numérisation d'une image est la conversion de celle-ci de son état analogique en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques $f(x,y)$, comme la montre la figure 1.2 où :

x,y : coordonnées cartésiennes d'un point de l'image. $f(x, y)$: niveau d'intensité.

La valeur en chaque point exprime la mesure d'intensité lumineuse perçue par le capteur[1].

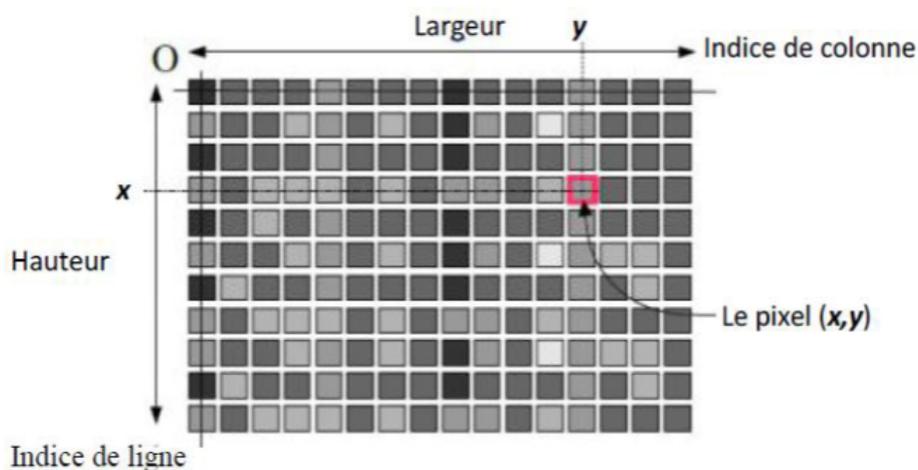


FIGURE 1.2 – Représentation d'image numérique[1]

1.4 Caractéristiques d'une image numérique

Comme nous l'avons déjà dit, l'image est un ensemble organisé d'informations qui contient les caractéristiques suivantes :

1.4.1 Pixel

Une image numérique est constituée d'un ensemble de points appelés pixels (abréviation de Picture Élément) pour former une image. Le pixel représente ainsi le plus petit élément constitutif d'une image numérique. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image[8]. Par exemple, peut être affichée comme un groupe de pixels dans la figure 1.3 ci-dessous.



FIGURE 1.3 – Représentation en groupe de Pixel d'une image du C.U.B.B.A.T

1.4.2 La résolution

La résolution est un terme souvent confondu avec la "définition", détermine par contre le nombre de points ou pixels par unité de surface, exprimé en points par pouce (PPP, en anglais DPI pour Dots Per Inch), un pouce représentant 2.54 cm[9].



FIGURE 1.4 – Représentation de Résolution d'une image.

Formule de conversion cm, Pixel, Pouce [10]

$$\left. \begin{array}{l} 1 \text{ cm} = 37.79527559055 \text{ pixel} \\ 100 \text{ pixel} = 2.646 \text{ cm} \\ 1 \text{ pouce} = 2.54 \text{ cm} \\ 1 \text{ cm} = 0.3937 \text{ pouce} \end{array} \right\} \text{ alors } 1 \text{ pouce} = 2.54 \text{ cm} = 72 \text{ pixel} \quad (1)$$

1.4.3 Dimension

On appelle définition, le nombre de points (pixel) constituant l'image, c.à.d. sa (dimension Informatique). Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image[11].



FIGURE 1.5 – Représentation de dimension d'une image

1.4.4 La texture

Une texture est une région dans une image numérique qui a des caractéristiques homogènes. Ces caractéristiques sont, par exemple, un motif basique qui se répète. La texture est composée de Texel, l'équivalent des pixels[7].

1.4.5 Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur[12].



FIGURE 1.6 – Image sans bruit



FIGURE 1.7 – Image avec bruit

1.4.6 La luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet.

Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes).
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir ; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites.[3]



FIGURE 1.8 – Image représente l'effet de luminance

1.4.7 Histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image. Il permet de donner un grand nombre d'informations sur la distribution des niveaux de gris (Couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans les cas d'une image trop claire ou d'une image trop foncée [7].

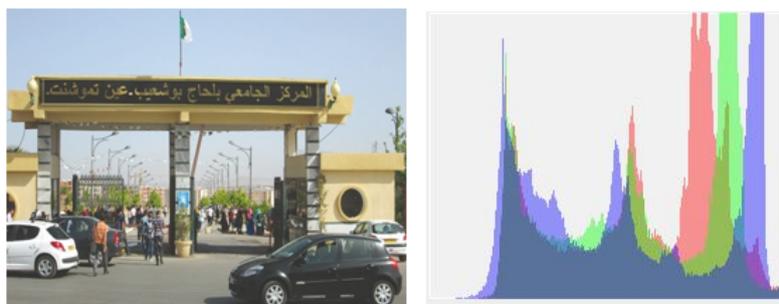


FIGURE 1.9 – Image du "C.U.B.B.A.T" avec histogramme

1.4.8 Le contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images.

Si $L1$ et $L2$ sont les degrés de luminosité respectivement de deux zones voisines $A1$ et $A2$ d'une image, le contraste C est défini par le rapport :[7]

$$C = \frac{L1 - L2}{L1 + L2} \quad (2)$$

1.5 Types d'images

On distingue deux types d'images :

1.5.1 Images matricielles

Dans la description que nous avons faite jusqu'à présent des images nous avons utilisé une matrice. On dit alors que l'image est matricielle ou en anglais bitmap. Ce type d'image est adapté à l'affichage sur écran mais peu adapté pour l'impression car bien souvent la résolution est faible (couramment de 72 à 150 ppp pour les images sur Internet)[2].

1.5.2 Images vectorielles

Le principe des images vectorielles est de représenter les données de l'image à l'aide de formules mathématiques. Cela permet alors d'agrandir l'image indéfiniment sans perte de qualité et d'obtenir un faible encombrement[2].

Par exemple, pour décrire un cercle dans une image, il suffit de noter la position de son centre et la valeur de son rayon plutôt que l'ensemble des points de son contour. Ce type est généralement obtenu à partir d'une image de synthèse créée par logiciel (exemple : Autocad) et non pas à partir d'un objet réel. Ce type est donc particulièrement adapté pour le travail de redimensionnement d'images, la cartographie ou l'infographie [2].

1.6 Codages des couleurs

Nous avons vu qu'une image apparaît comme une matrice où chaque case contient des nombres associés à une couleur. Usuellement on distingue 3 grands types de couleurs pour une image numérique :

- Le noir et blanc.
- Les niveaux de gris.
- La couleur.

Ces types sont généralement à choisir lors d'une numérisation par scanner ou lors de la configuration d'un appareil photographique [2].

1.6.1 Image noir et blanc

Le noir et blanc est le plus simple. Le contenu de chaque case de la matrice est soit un 0 (noir) soit 1 (blanc). Le nombre de couleurs n'est que de 2 et le rendu de l'image le moins performant mais parfois suffisant dans le cadre par exemple de documents scripturaux[2].

16 millions de couleurs	niveaux de gris	noir et blanc
		
$303 \times 452 \times 3 = 410.868$ octets $303 \times 452 \times 3 \times 8 = 3.286.944$ bits	$303 \times 452 = 136.956$ octets $303 \times 452 \times 8 = 1.095.648$ bits	$(303 \times 452) / 8 = 15.604$ octets $303 \times 452 = 136.956$ bits

FIGURE 1.10 – Comparaison du volume d'une image "C.U.B.B.A.T" pour différents codages

1.6.2 Niveaux de gris

Le codage dit en niveaux de gris permet d'obtenir plus de nuances que le simple noir et blanc. Il offre des possibilités supplémentaires pour coder le niveau de l'intensité lumineuse. La couleur est codée souvent sur un octet soit 8 bits ce qui offre la possibilité d'obtenir 256 niveau de gris (0 pour le noir et 255 pour le blanc). On peut aussi le faire avec 16 niveaux de gris (4 bits)[2].



FIGURE 1.11 – Image "C.U.B.B.A.T" niveaux de gris

1.6.3 Image couleur

1.6.3.1- Principe

La couleur d'un pixel est obtenue, comme le ferait un peintre, par le mélange de couleurs fondamentales. Il ne s'agit pas ici de décrire toutes les techniques utilisées. Nous allons décrire un des principes les plus couramment utilisé qui est celui de la synthèse additive[2].

1.6.3.2- Codage RVB

Le principe consiste à mélanger les 3 couleurs : rouge, vert et bleu (noté RVB ou RGB en anglais). A l'aide de ces 3 couleurs, on obtient toute une palette de nuances allant du noir au blanc. A chaque couleur est associé un octet (donc 256 niveaux de luminosité) de chacune des couleurs fondamentales [2].

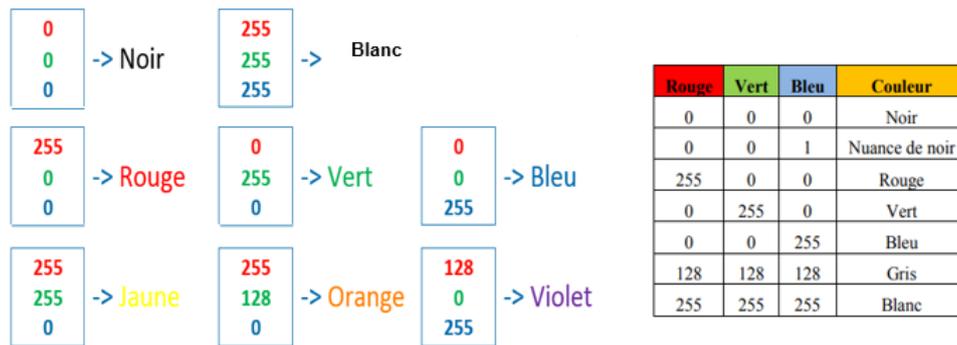


FIGURE 1.12 – Principe codage de la couleur[2]

Un pixel "couleur" est alors codé avec 3 octets et on a alors la possibilité d'obtenir 2^{24} possibilités de couleurs soit de l'ordre de 16 millions de couleurs différentes[2].



FIGURE 1.13 – Valeurs RVB d'un pixel "C.U.B.B.A.T"

Question

comment convertit-on une image couleur en une image en niveau de gris[13] ?

$$NdG = \frac{R + G + B}{3}$$

$$NdG = \frac{\log(R) + \log(G) + \log(B)}{3} \quad (3)$$

$$NdG = 0.30R + 0.59G + 0.11B$$

1.7 Formats d'image

Lors de son enregistrement, une image est stockée suivant un format d'image précis. Ce format doit permettre de stocker l'information de l'image avec un minimum de perte d'informations. Il existe ainsi différents formats qui pourront favoriser soit la conservation de la qualité soit la diminution de la taille du fichier informatique[2].

Le tableau suivant donne les principales caractéristiques des principaux standards utilisés [2] :

Format	Type	Compression données	Nombre couleurs	Affichage progressif	Usage
BMP	matriciel	non	de 2 à 16 millions	non	Image non dégradée ; Taille fichier importante.
JPG	Matriciel	oui	16 millions	oui	Taux de compression réglable ; Perte de qualité.
GIF	Matriciel	oui	De 2 à 256 couleurs	oui	Pas de perte de qualité ; Usage pour Internet.
TIFF	Matriciel	oui	16 millions	non	Pas d'usage Internet
PNG	Matriciel	oui	de 2 à 16 millions	Oui	Recommandé pour Internet
SVG	Vectoriel	oui	16 millions	non	Usage cartographie, animations

FIGURE 1.14 – Principaux formats d'image[2]

1.8 Le système de traitement d'images

On peut appliquer plusieurs traitements sur l'image numérique, la figure suivante résume l'ensemble de ces traitements [3].

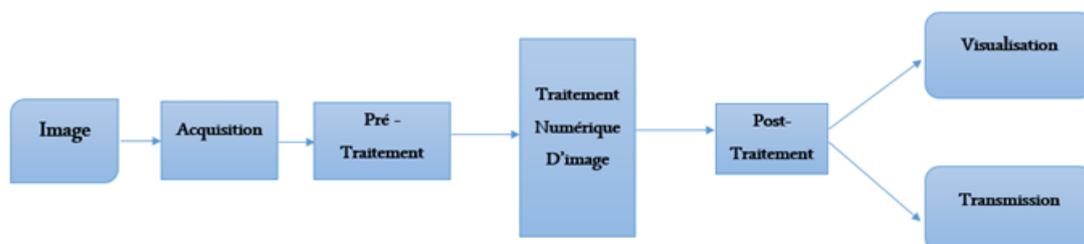


FIGURE 1.15 – Schéma d'un système de traitement d'images[3]

1.8.1 Acquisition

pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation (échantillonnage, quantification). On utilise plus couramment des caméras vidéo, des appareils photos numériques. En médecine, on utilise des imageurs IRM, TEP, scanner X, écho doppler, échographie, scintigraphie etc...[14].

1.8.2 Filtrage

est une opération qui consiste à réduire le bruit contenu dans une image au moyen d'algorithmes provenant des mathématiques par l'utilisation de méthodes d'interpolation ou de la morphologie mathématique [14].

On peut diviser les filtres en deux grandes catégories :

1- Filtres linéaires

Le filtre est dit linéaire si la valeur du nouveau pixel est une combinaison linéaire des valeurs des pixels du voisinage[15].

Il y a plusieurs types de filtrage linéaire nous distinguons :

a - Filtre passe-bas (lissage)

ce filtre n'affecte pas les composantes de basse fréquence dans les données d'une image, mais doit atténuer les composantes de haute fréquence. L'opération de lissage est souvent utilisée pour atténuer le bruit et les irrégularités de l'image [14].

b - Filtre passe-haut (accentuation)

le renforcement des contours et leur extraction s'obtiennent dans le domaine fréquentiel par l'application d'un filtre passe-haut. Le filtre digital passe-haut a des caractéristiques inverses du filtre passe-bas. Ce filtre n'affecte pas les composantes de haute fréquence d'un signal, mais doit atténuer les composantes de basse fréquence [14].

c - Filtre passe-bande (différentiation)

cette opération est une dérivée du filtre passe-bas. Elle consiste à éliminer la redondance d'information entre l'image originale et l'image obtenue par filtrage passe-bas. Seule la différence entre l'image source et l'image traitée est conservée [14].

d - Filtre directionnel

dans certains cas, on cherche à faire apparaître des détails de l'image dans une direction bien déterminée. Pour cela, on utilise des filtres qui opèrent suivant des directions (horizontales, verticales et diagonales) [14].

* Quelques méthodes de filtrage linéaire [8] :

1. Filtre moyenneur.
2. Filtre gaussien.

Le principal inconvénient des filtres linéaires est que la réduction de bruit s'accompagne d'un étalement des transitions entre régions. Ce problème peut être surmonté par l'utilisation des filtres non linéaires [3].

2 - Filtres non linéaires

Le filtrage non-linéaire est une opération qui remplace la valeur de chaque pixel par une combinaison non-linéaire des valeurs de ses pixels voisins, ce type de filtre pallie les inconvénients majeurs des filtres linéaires dont la présence de valeurs aberrantes même après filtrage et la mauvaise conservation des transitions [15]. On trouve aussi dans les filtres non linéaires les deux types de filtrages : le Filtre passe-bas et le Filtre passe-haut.

* Quelques méthodes de filtrage non linéaire [8][15] :

1. Filtre médian.
2. Le filtre de Canny.
3. Le filtre de Laplacien.
4. Le filtre de gradient.
5. Les filtres de PREWITT, SOBEL, FREEMAN, ET KIRSCH.
6. Symmetric Nearest Neighbor .
7. Nagao.

1.8.3 Segmentation

A partir d'une image numérique, il convient de d'extraire les informations pertinentes en regard de l'application concernée, les traiter puis les interpréter. Le terme générique d'analyse d'images désigne l'ensemble de ces opérations.

En analyse, d'images on distingue les traitements de bas-niveau et ceux de haut-niveau. Cette distinction est liée au contenu sémantique des entités traitées et extraites de l'image. Les traitements de bas-niveau opèrent, en général, sur les grandeurs calculées à partir des valeurs attachées à chaque point de l'image sans faire nécessairement la liaison avec la réalité qu'elles représentent. A l'opposé, les traitements de haut-niveau s'appliquent à des entités de nature symboliques associées à une représentation de la réalité extraite de l'image ; ils sont relatifs à l'interprétation et à la compréhension de l'image. La segmentation d'image est un traitement de bas-niveau qui consiste à créer une partition d'une image A en sous-ensembles R_i appelés régions, tel que :

$$\begin{aligned} \forall i \quad R_i &\neq \emptyset \\ \forall i, j \quad i \neq j \quad R_i \cap R_j &= \emptyset \quad (4) \\ A &= \bigcup R_i \end{aligned}$$

Une région est un ensemble connexe de pixels ayant des caractéristiques communes (intensité, texture, ...) qui les différencient des pixels des régions voisines. Dans le cas des images couleur ces caractéristiques sont déterminées à partir des composantes colorimétriques des pixels. Les connaissances utilisées sont les plus souvent du domaine de l'image numérique et du traitement du signal, donc sémantiquement assez pauvres[4].

1.8.3.1- Les différentes méthodes de segmentation

La segmentation définie par l'ensemble des pixels representent en deux méthodes, soit par : approche région de la segmentation ou bien par les contours de la région approche contour de la segmentation. Ces deux approches sont duales du fait que chaque région possède un contour et qu'un contour délimite forcément une région[4].

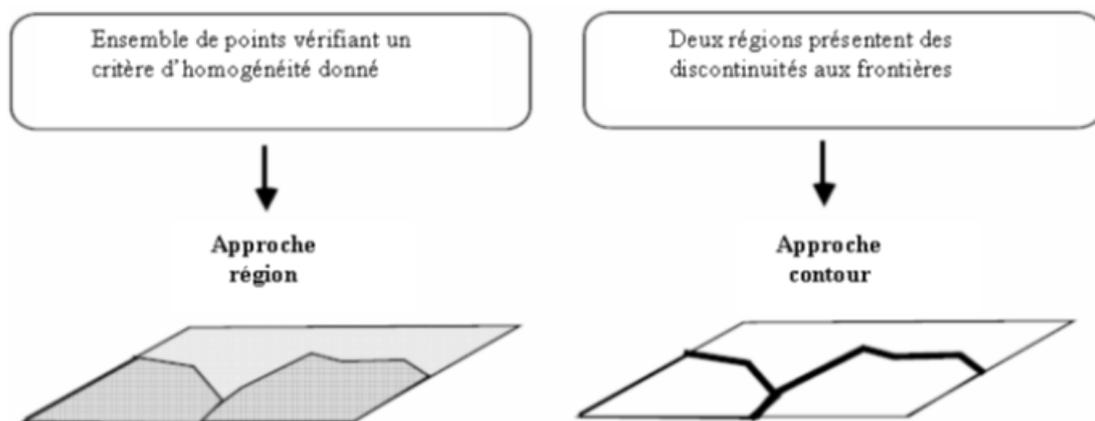


FIGURE 1.16 – approche région et approche contour[4]

a - Approche « régions »

En approche région, l'affinité des points connexes est favorisée. Cela peut être vu comme une technique contextuelle. Les points connexes ayant des propriétés semblables (attributs) : intensité de gris, couleur, texture, vont être réunis dans le même ensemble [16].

b - Approche « contours »

Cette approche, est une technique non contextuelle qui ignore les rapports pouvant exister entre les régions de l'image. On regroupe les pixels suivant un attribut global. Elle comprend les techniques de détection de contours, mais les contours obtenus ne conduisent pas toujours directement à la partition recherchée. En effet, les pixels contours mis en évidence pour une forme, généralement ne sont pas connexes. Il faut alors appliquer des algorithmes de fermeture de contours. Ce n'est qu'après fermeture que les régions apparaissent, déterminées par l'intérieur des contours [16].

1.9 Domaines D'application

Le traitement d'images possède l'aspect multidisciplinaire. On trouve ses applications dans des domaines très variés tels que[14] :

1.9.1 Imagerie aérienne et spatiale

Dans laquelle les traitements concernent l'étude des images satellites, l'analyse des ressources terrestres, la cartographie automatique, les analyses météorologiques[14].

1.9.2 Technologies biomédicales

On trouve des utilisations de cette technique dans l'échographie, la résonance magnétique nucléaire, ainsi que dans le domaine de la reconnaissance automatique des cellules ou de chromosomes[14].

1.9.3 La robotique

Qui connaît actuellement le plus grand développement et dont les tâches usant de l'imagerie sont principalement l'assemblage (pièce mécanique, composants électroniques,...), le contrôle de qualité, ainsi que la robotique mobile[14].

1.9.4 la télésurveillance

Exemple, radar automatique : recherche en temps réel d'un véhicule par reconnaissance de son immatriculation parmi un flot de véhicules circulant sur le boulevard périphérique par caméra fixe[14].

1.9.5 Simulation et contrôle de processus

On trouve des utilisations de cette technique dans les cours de pilotage et le contrôle des panneaux [14].

Aussi : L'astronomie, la chimie, la physique nucléaire (identification de trajectoires de particules), l'armement (guidage de missiles, la détection des couleurs + reconnaissance des formes pour les panneaux de signalisation routière c'est le but de notre sujet de projet fin d'étude), métiers du spectacle, les télécommunications (TV, radio, vidéo, publicité,...), l'architecture, l'imprimerie,... [14].

1.10 Conclusion

Dans ce chapitre, nous avons présenté une masse importante d'informations concernant les méthodes utilisées dans le domaine de traitement d'image. Dans ce qui suit,nous allons parler sur les outils utilisés pour réaliser notre projet.

Chapitre 2

Outil de développement

Sommaire

2.1	Introduction	20
2.2	Présentation de la librairie OpenCV	20
2.2.1	Historique	20
2.2.2	Qu'est -ce que OpenCV	21
2.2.3	Fonctionnalités	21
2.2.4	Modules OpenCV	22
2.3	Installation et configuration OpenCV2.4.13 avec Netbeans 8.2	23
2.3.1	Téléchargement et extraction OpenCV	23
2.3.2	Netbeans	24
2.3.3	Configuration du OpenCV dans Netbeans 8.2	24
2.3.4	Mise en place variables d'environnement	30
2.4	Conclusion	32

2.1 Introduction

Dans ce chapitre nous allons présenter la bibliothèque OpenCV spécialisée dans le traitement d'images et vidéo en temps réel fournit un ensemble de packages ainsi qu'une large gamme de filtres et des algorithmes de traitement des images. Accessibles au travers des applications pour les langages C, C++, Python et Java ce dernier est le langage choisit pour développer notre application en utilisant un environnement de développement (IDE Netbeans), open source, permet de prise en charge plusieurs langage en plus de java :C, C++,..., qui offre un éditeur couleur, JavaDoc, éditeur graphique d'interface, disponible pour plusieurs systèmes d'exploitations et qui gère les bibliothèques nécessaires au projet, générer les documentations. en suite la configuration du OpenCV avec Netbeans sous Windows pas à pas.

2.2 Présentation de la librairie OpenCV

2.2.1 Historique

Officiellement lancé en 1999, le projet OpenCV est développé initialement par Intel pour optimiser les applications gourmandes en temps processeur. Cela faisait partie d'une série de projets tels que l'affichage d'un mur en 3 dimensions. Cette bibliothèque est distribuée sous licence BSD[17]. Les principaux acteurs du projet sont l'équipe de développement de bibliothèque de chez Intel ainsi qu'un certain nombre d'experts dans l'optimisation de chez Intel Russie[17].

Les objectifs de base du projet étaient :

- Faire des recherches sur la vision par ordinateur en vue de fournir un logiciel libre et optimisé.
- Établir une infrastructure commune s'appuyant sur les développeurs pour obtenir un code plus lisible et transférable.
- Continuer à développer en rendant le code portable et permettre des performances optimisées gratuites avec une licence qui est libre de toutes contraintes commerciales.

La première version alpha d'OpenCV fut présentée lors de la conférence IEEE sur la vision par ordinateur et la reconnaissance de formes en 2000. Après cela, cinq versions bêta ont été publiées entre 2001 et 2005 et la première version 1.0 a été publiée en 2006[17]. Au milieu de l'année 2008, OpenCV obtient l'appui de la société de robotique Willow Garage et la bibliothèque est encore développée à ce jour. Une version 1.1 est sortie en Octobre 2008 et un livre écrit par deux auteurs d'OpenCV, publié par O'Reilly Media est sorti sur le marché ce même mois.

La deuxième version majeure d'OpenCV née en octobre 2009. Il s'agit d'OpenCV 2 incluant des changements majeurs au niveau du langage C++ servant à faciliter le développement de nouvelles fonctions et améliorant les performances[17].

2.2.2 Qu'est -ce que OpenCV

OpenCV (Open Source Computer Vision) 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 Java, Python, Ruby, Matlab et autre langage. Open CV est orienté 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 plus de 500 fonctions[17].

2.2.3 Fonctionnalités

La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques. Elle propose des opérations de traitement appliqué sur les images et vidéo[17].

1 - Traitement d'images

Elle propose la plupart des opérations classiques en traitement bas niveau des images :

- Lecture, écriture et affichage d'une image.
- Calcul de l'histogramme des niveaux de gris ou d'histogrammes couleurs.
- Lissage, filtrage.
- Seuillage d'image (méthode d'Otsu, seuillage adaptatif).
- Segmentation (composantes connexes, GrabCut).
- Morphologie mathématique[17].

2 - Traitement vidéo

Cette bibliothèque s'est imposée comme un standard dans le domaine de la recherche parce qu'elle propose un nombre important d'outils issus de l'état de l'art en vision des ordinateurs tels que :

- Lecture, écriture et affichage d'une vidéo (depuis un fichier ou une caméra).
- Détection de droites, de segment et de cercles par Transformée de Hough.

- Détection de visages par la méthode de Viola et Jones.
- Cascade de classifieurs boostés.
- Détection de mouvement, historique du mouvement.
- Poursuite d’objets par mean-shift ou Camshift.
- Détection de points d’intérêts.
- Estimation de flux optique (Méthode de Lucas–Kanade).
- Triangulation de Delaunay.
- Diagramme de Voronoi.
- Enveloppe convexe.
- Ajustement d’une ellipse à un ensemble de points par la méthode des moindres carrés[17].

2.2.4 Modules OpenCV

OpenCV a une structure modulaire. Les principaux modules d’OpenCV sont énumérés :

1 - CxCore (Coeur d’OpenCV)

Contient les structures de donnée et les opérations mathématiques de base. Elle permet de réaliser les fonctions d’algèbre matriciel, la transformation de données, la gestion de mémoire, la manipulation d’erreurs, le chargement dynamique de code et permet aussi la réalisation des graphiques[17].

2 - HighGui (bibliothèque d’interfaçage graphique)

OpenCV intègre sa propre bibliothèque haut-niveau pour ouvrir, enregistrer et afficher des images et des flux vidéo. Celle-ci contient aussi un certain nombre de fonctions permettant de réaliser des interfaces graphiques très simples, mais largement suffisantes pour tester nos programmes[17].

3- ImgProc (traitement d’image)

Ce module comprend des algorithmes de traitement d’image de base, y compris le filtrage d’image, transformations d’image, les conversions d’espace couleur et etc[17].

4 - Vidéo (traitement de flux vidéo)

Ceci est une vidéo module d'analyse qui inclut des algorithmes de suivi (tracking) d'objets, des algorithmes de soustraction de fond et etc[17].

5 - ObjDetect (détection d'objets)

Cela comprend l'objet de détection et de reconnaissance des algorithmes pour les objets standard[17].

6 - ML (bibliothèque d'apprentissage automatique)

Comporte les fonctions de classification, d'analyse de donnée et des outils de clustering[17].

7 - Calib3D

Calibrage caméra et reconstruction 3D [17].

2.3 Installation et configuration OpenCV2.4.13 avec Netbeans 8.2

2.3.1 Téléchargement et extraction OpenCV

1. Télécharger OpenCV pour Windows
2. Extraire le programme dans C:\

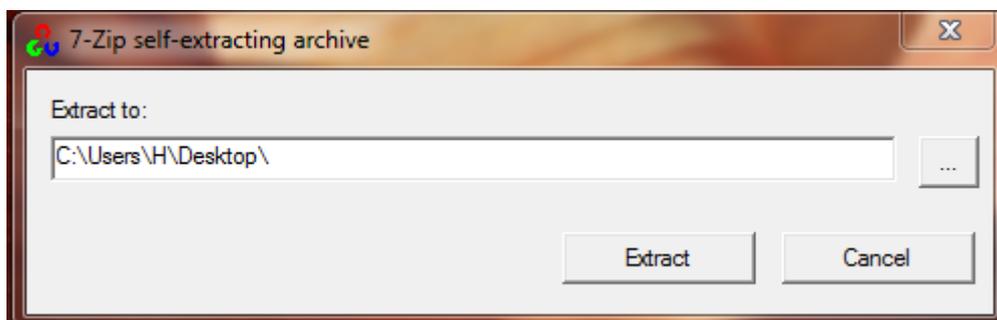


FIGURE 2.1 – Choix le chemin d'extraction d'OpenCV

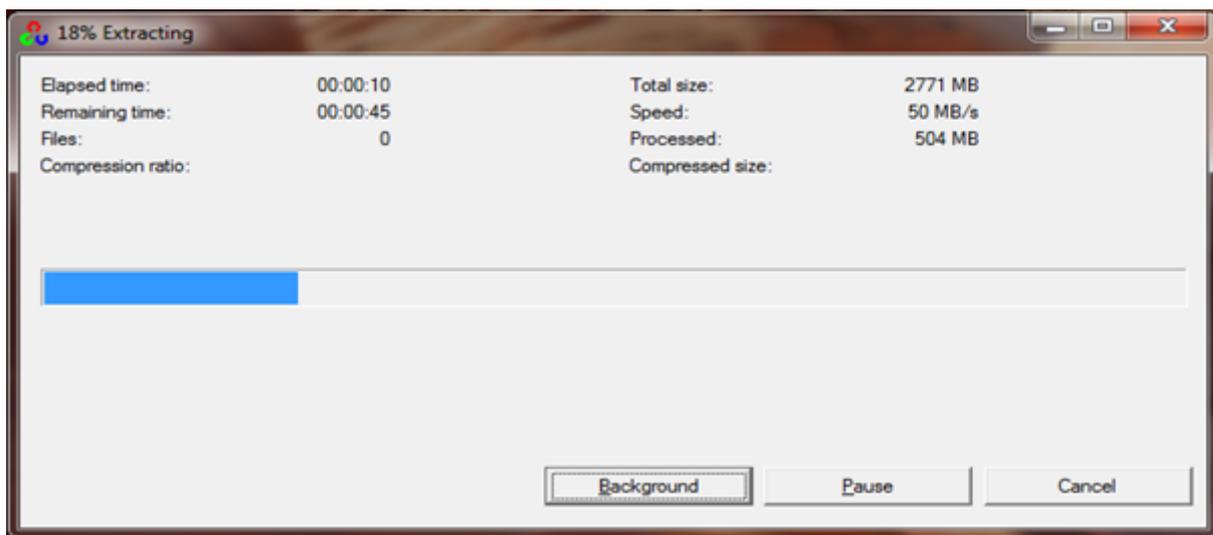


FIGURE 2.2 – Extraction d'OpenCV

2.3.2 Netbeans

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML, ou d'autres (dont Python et Ruby) par l'ajout de greffons. Il offre toutes les facilités d'un IDE moderne (éditeur en couleurs, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web)[18].

Compilé en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java. NetBeans constitue par ailleurs une plate forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate forme. L'IDE Netbeans s'enrichit à l'aide de greffons[18].

2.3.3 Configuration du OpenCV dans Netbeans 8.2

1. Créer un nouveau projet dans Netbeans 8.2.
2. Aller sur File → New Project → Projet (Voir Figure 2.3).

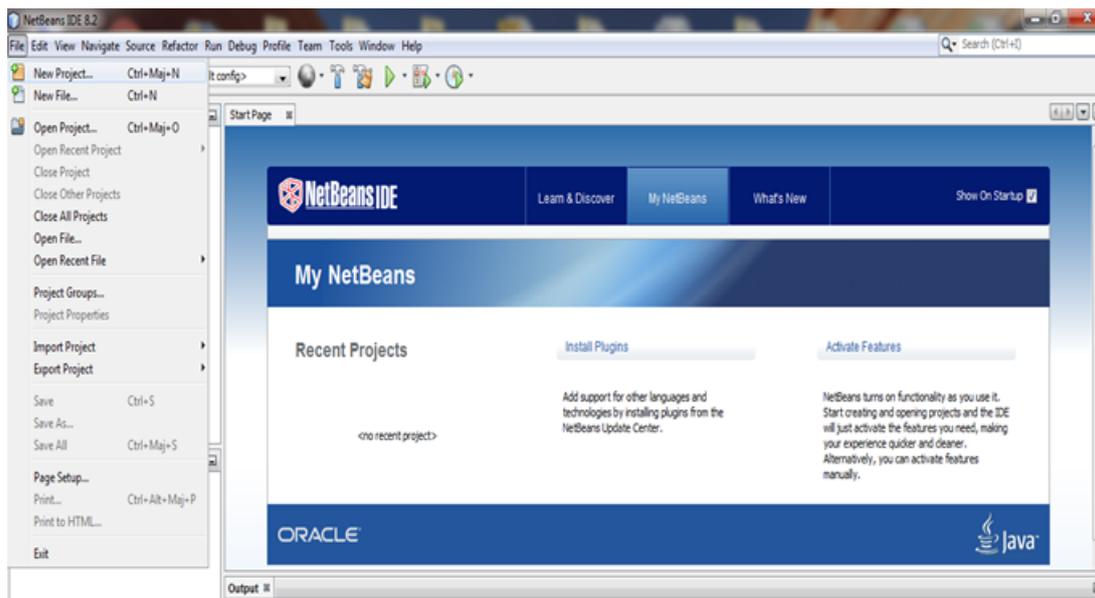


FIGURE 2.3 – Création d'un nouveau projet

3. Choisie Java → Java Application (Voir Figure 2.4).

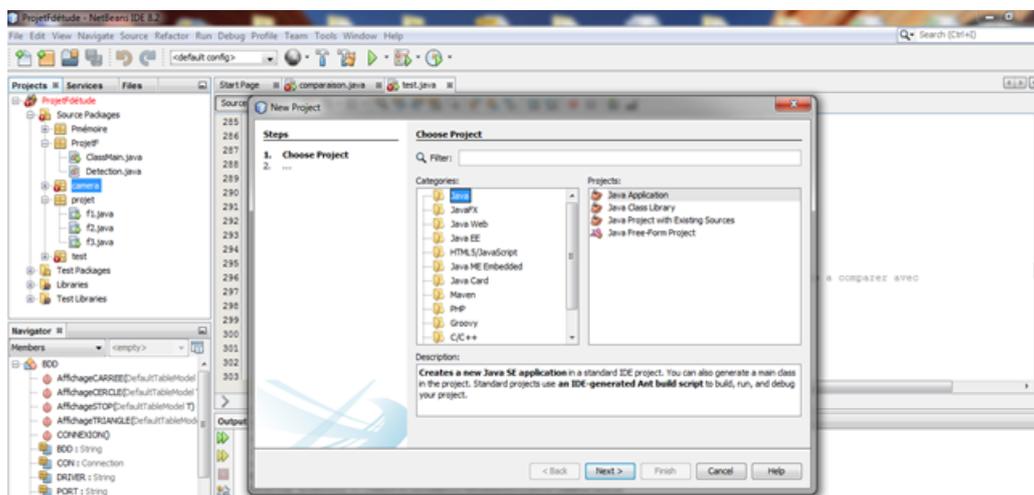


FIGURE 2.4 – Création d'un nouveau projet

4. Donner un nom pour votre Projet (Voir Figure 2.5).

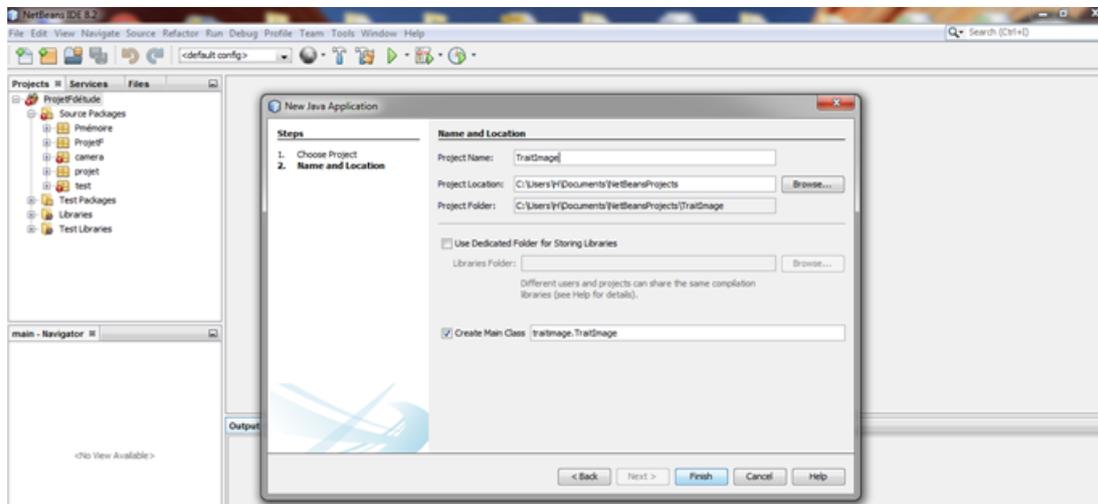


FIGURE 2.5 – Création d'un nouveau projet

5. Pour ajouter la bibliothèque OpenCV sur votre Projet aller vers le projet qui vous avez créé → Clic droit sur Libraries → Add Library (Voir Figure 2.6).

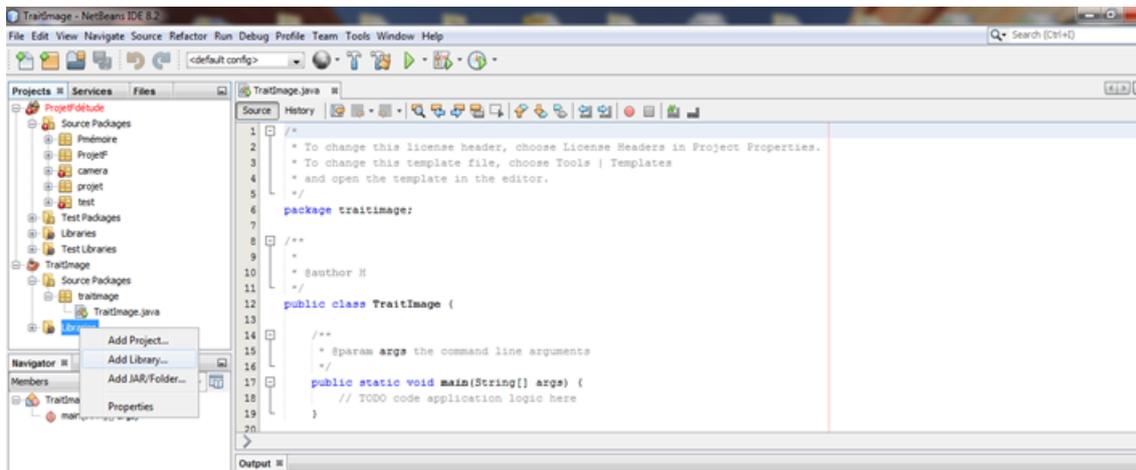


FIGURE 2.6 – L'ajoute de la Bibliothèque OpenCV

6. Une fenêtre s'affiche → Clic droit sur Create (Voir la figure 2.7).

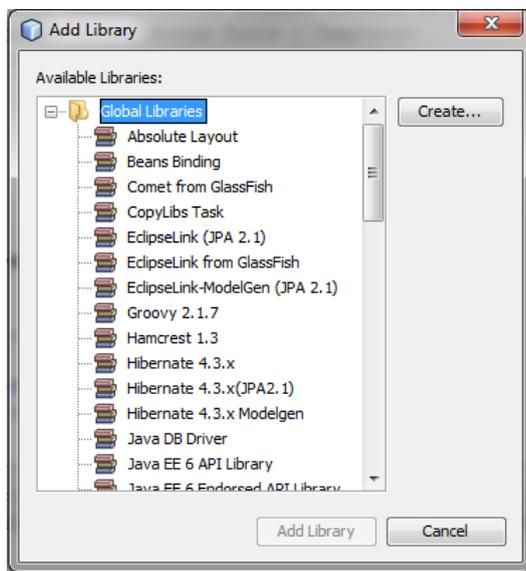


FIGURE 2.7 – L'ajoute de la Bibliothèque OpenCV

7. Tapez le Nom de Bibliothèque OpenCV 2.4.13 (voir la Figure 2.8).

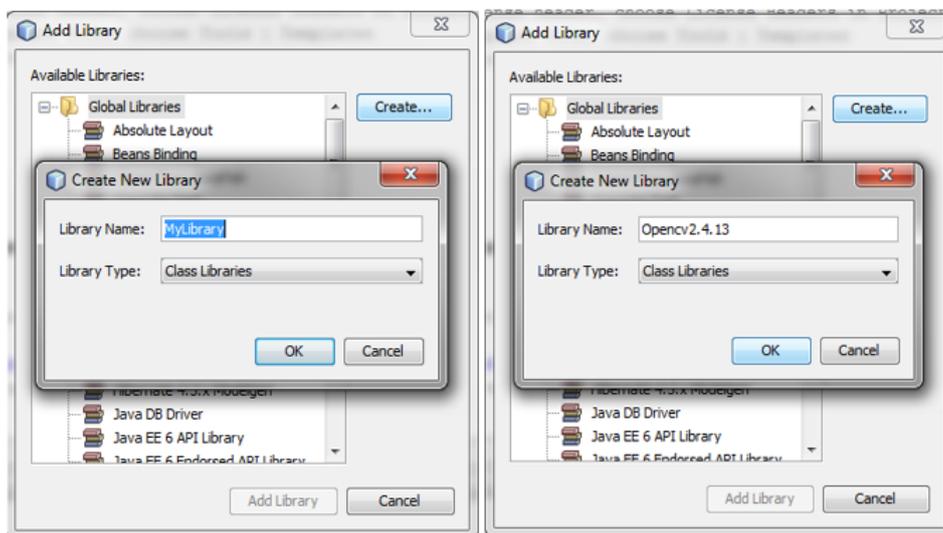


FIGURE 2.8 – L'ajoute de la Bibliothèque OpenCV

8. En suite cliquer sur OK et aller vers le chemin d'installation du OpenCV (Voir Figure 2.9).

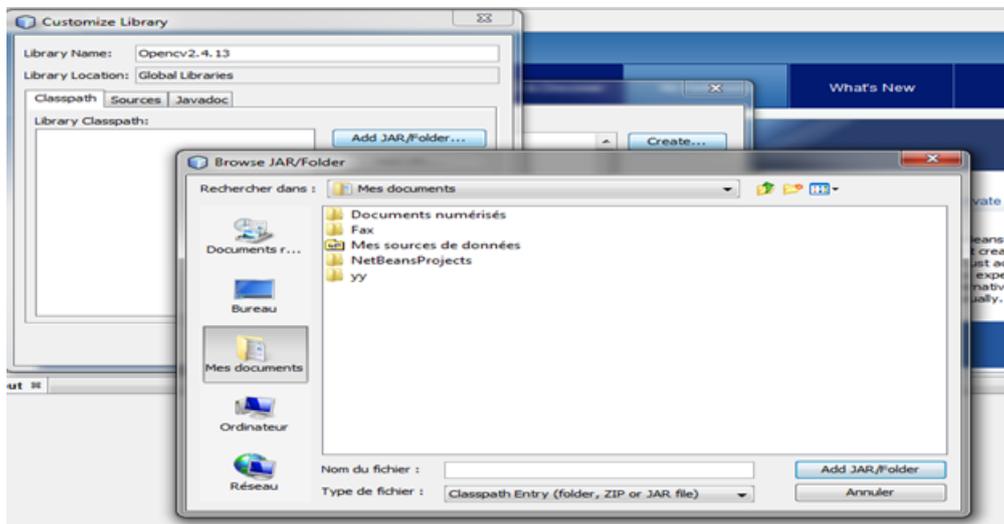


FIGURE 2.9 – L'ajoute de la Bibliothèque OpenCV

9. Ajouter opencv-2413.jar de la bibliothèque OpenCV (Voir Figure 2.10).

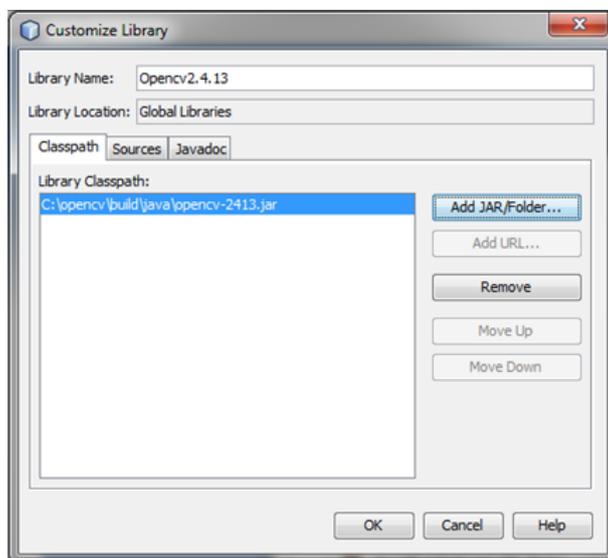


FIGURE 2.10 – L'ajoute de la Bibliothèque OpenCV

10. Voilà la bibliothèque OpenCV et ajouter dans votre Projet.(Voir Figure 2.11).

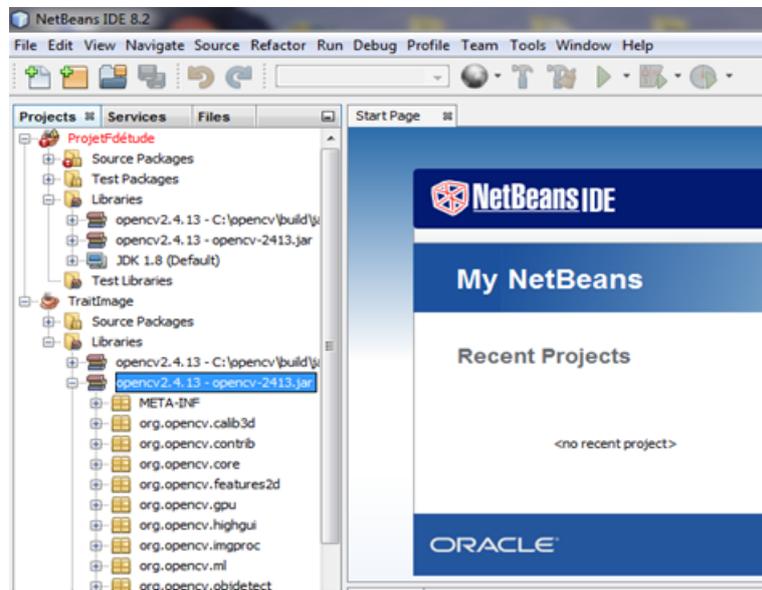


FIGURE 2.11 – L'ajoute de la Bibliothèque OpenCV

11. Il faut ajouter le chemin de la bibliothèque OpenCV clic droit sur le projet → Propriétés (Voir Figure 2.12).

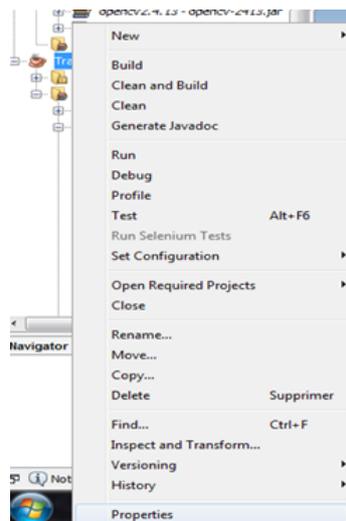


FIGURE 2.12 – L'ajoute du chemin OpenCV

12. Clic sur Run et Copier le chemin D'OpenCV `C:\Users\Lenovo\Desktop\opencv\build\java\x64` dans la case Working Directory puis OK (Voir Figure 2.13).

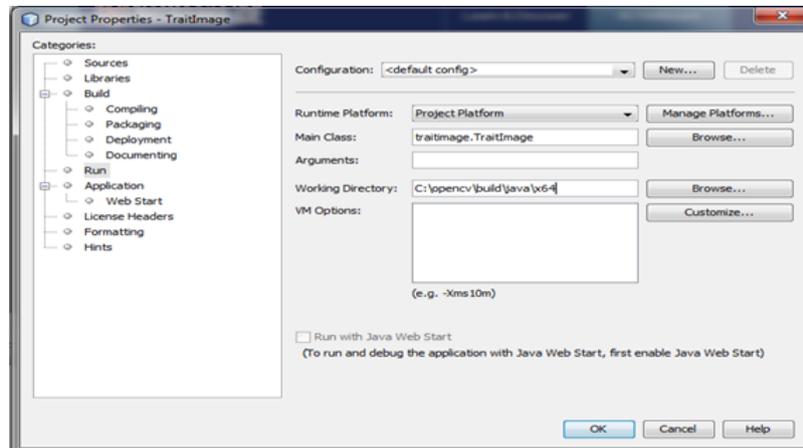


FIGURE 2.13 – L'ajoute du chemin OpenCV

2.3.4 Mise en place variables d'environnement

1. Pour définir le variable d'environnement aller sur :
2. Menu démarrer → Panneau de configuration → Système et sécurité → Système → clic droit sur Paramètres système avancés (Voir Figure 2.14).

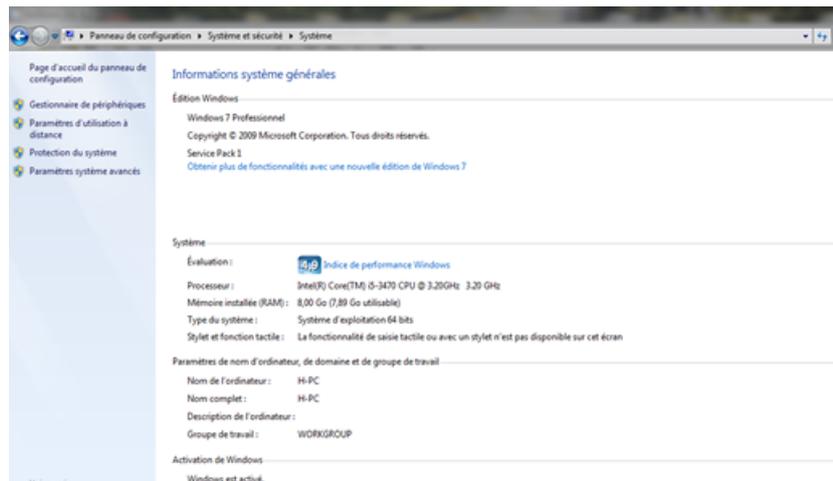


FIGURE 2.14 – Propriétés du système

3. Clic droit sur variables d'environnement (Voir Figure 2.15).

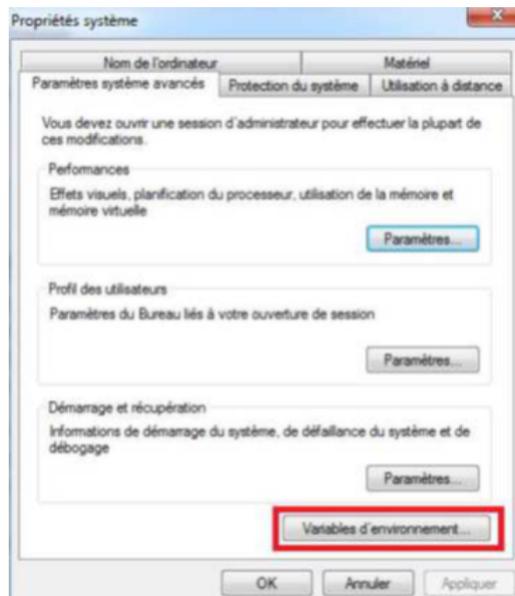


FIGURE 2.15 – Propriétés du système

4. On trouve la variable PATH sous Variables système, aller vers l'emplacement d'extraction du Opencv et sélectionner le chemin :
`C:\Users\Lenovo\Desktop\opencv\build\java\x64` et copier à la fin Valeur de la variable et cliquer sur OK (Voir Figure 2.16).

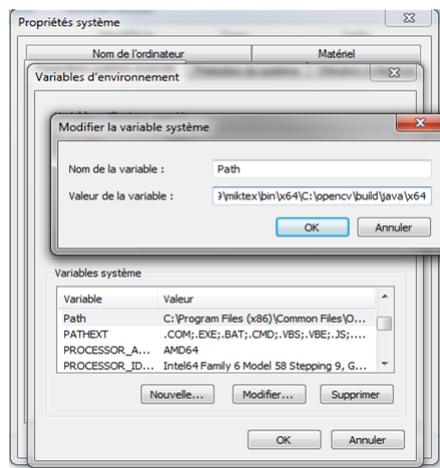


FIGURE 2.16 – Propriétés du système

5. Alors le processus de configuration du OpenCV est terminé.

2.4 Conclusion

La bibliothèque OpenCV est une bibliothèque large et simple à configurer avec plusieurs environnement de programmation par exemple Netbeans, Visual Studio, Python, Code blocs, c++..., on trouve des exemples ou des codes simple sont disponible en ligne pour les débutons du domaine de traitement d'images, donc cette bibliothèque permet aux utilisateurs de créer des applications aux traitements d'image et vidéo complexe après l'inclus de la bibliothèque OpenCV dans l'environnement de travail.

Chapitre 3

Conception et Développement

Sommaire

3.1	Introduction	34
3.2	Etat de l'art	34
3.2.1	Système RSR intégré dans les voitures	35
3.2.2	Applications mobiles avec RSR	36
3.2.3	Travaux de recherche	36
3.3	Les panneaux de signalisation routière	38
3.3.1	Les types des panneaux de signalisation routière	39
3.4	Mise en œuvre (Conception et développement du projet)	42
3.4.1	Description de l'approche	42
3.4.2	Le travail avec OpenCV	43
3.4.3	Etape de prétraitement de l'image et de détection	43
3.4.4	Etape de reconnaissance	55
3.5	Conclusion	65

3.1 Introduction

Dans ce chapitre nous commencerons par une discussion sur les applications existantes et les travaux de recherche dans le cadre de travail des systèmes de détection et de reconnaissance des panneaux de signalisation, après ceci nous allons entamer les panneaux de signalisation et leurs caractéristiques, ensuite nous présenterons l'implémentation détaillée de l'application en expliquant les étapes de notre système.

3.2 Etat de l'art

Ces dernières années, de nombreux systèmes d'assistance avancés aide à la conduite (ADAS) ont été développés afin de réduire le nombre d'accidents sur les routes[19]. Cette technologie est mise en œuvre dans la plupart des nouveaux modèles de voitures de toutes les marques et permet au conducteur de connaître l'état de la route d'une manière précise. Ces systèmes permettent aussi d'avertir le conducteur de l'état des routes, signaler les collisions possibles, détecter les obstacles sur la route et départs de voie ou détecter et communiquer au conducteur les panneaux de signalisation qui se trouvent sur la route[19].

En ce qui concerne les systèmes de détection et de reconnaissance du signal (RSR), c'est un système qui agit comme un observateur infatigable en combinaison avec d'autres systèmes de voiture pour assurer la sécurité des passagers du véhicule. C'est généralement un avertissement automatique qui va avertir le conducteur en émettant un bip et en montrant directement la limitation du chemin via un ordinateur de bord. Ce type de technologie n'est pas un standard puisque chaque fabricant lui donne des caractéristiques[19].

Cette détection et reconnaissance de panneaux de signalisation est possible grâce à l'utilisation de caméras vidéo qui sont normalement situés sur le pare-chocs avant ou sur le pare-brise du véhicule afin d'enregistrer les images de la route.

Une fois le signal vidéo capturé, c'est l'ordinateur qui est responsable de la lecture du signal et de son interprétation. Le système doit être suffisamment robuste et rapide pour avertir le conducteur en temps réel. Avec cela, le conducteur peut consulter en cas de dépassement de la limitation de la route pour augmenter la sécurité et oubliez les amendes éventuelles si vous suivez les instructions affichées par le système[19].

Actuellement, il y a des voitures qui ont déjà été munies par ces systèmes comme nous l'avons dit auparavant et il y a aussi des applications qui peuvent être utilisées via le mobile.

3.2.1 Système RSR intégré dans les voitures

Les fabricants qui ont actuellement intégré ces systèmes dans leur véhicules sont nombreux tels que : Audi, BMW, Citroën, Ford, Honda, Infiniti, Jaguar, JEEP, Land Rover, Lexus, Mercedes, Nissan, Opel, Peugeot, Porsche, Renault, Toyota, Volkswagen et Volvo, ci-dessous un exemple d'application TSR dans une voiture de la marque Opel (Figure 3.1)[19].



FIGURE 3.1 – Exemple d'application RSR dans une voiture de la marque Opel

En ce qui concerne les systèmes RSR intégrés dans les véhicules, il existe deux types d'applications de la technologie RSR[19] :

1. **Application passive** : consiste à informer le conducteur par le biais de pictogrammes ou des sons qu'il entre dans une zone qui a une nouvelle limitation indiquée pour un feu de circulation. Dans ce cas, ce sera le conducteur qui prendra la décision de respecter le signal ou non[19].
2. **Application active** : consiste à intervenir automatiquement sur la voiture quand il détecte le signal. Par exemple, si le conducteur conduit à une vitesse excessif et le système RSR détecte un signal d'arrêt et la voiture n'interprète pas que le conducteur a l'intention de s'arrêter, l'ordre de freiner sera directement dirigé vers la voiture pour éviter un éventuel accident[19].

3.2.2 Applications mobiles avec RSR

Les applications mobiles offrent une application passive de la technologie RSR et qui n'ont aucun type de communication avec l'ordinateur du véhicule dans le cas où la voiture en a un. Plusieurs applications permettent de détecter et de reconnaître les panneaux de circulation tels que myDriveAssist ou Skylten EU[20].

L'application la plus précise et professionnelle est myDriveAssist, qui permet de détecter les limitations de vitesse et les panneaux d'interdiction utilisant simplement l'appareil photo Smartphone il permet également d'avertir le conducteur dans le cas où il circule trop rapide grâce à l'utilisation permanente du signal GPS[20].



FIGURE 3.2 – Application mobile myDriveAssist

Cette application peut être téléchargée pour le système d'exploitation Android et pour IOS. Elle peut également être utilisée dans plusieurs pays et permet la mise à jour des limites de vitesse par les utilisateurs que le mobile ne détecte pas en raison du mauvais état de quelques signes[20].

Parmi les inconvénients de l'utilisation de cette application que la qualité dépend de la caméra du Smartphone et que la batterie de l'appareil se décharge rapidement[20].

3.2.3 Travaux de recherche

Dans cette section, nous discuterons brièvement des lignes générales qui suivent les principaux travaux de recherche sur la reconnaissance des panneaux de signalisation routière. Un grand nombre d'articles sur ce sujet [21] ont été publiés.

Ces systèmes comportent deux parties fondamentales : la détection des signaux dans l'image et la reconnaissance dans une base de données[22].

La plupart des articles scientifiques examinés utilisent la couleur comme l'un des principales fonctionnalités pour détecter les panneaux de circulation à côté des fonctions basée sur la forme[22].

Pour la détection et la segmentation de la couleur, la grande majorité des articles coïncidaient avec l'utilisation de seuils dans un espace colorimétrique HSV (Tonalité, Saturation et Valeur), car il est plus intuitif que les autres et nous permet également de dépendre moins de luminosité ou d'ombres de l'image[22].

Le seuillage sur cet espace peut être plus pratique dans certaines situations et généralement moins sensible aux changements dans les conditions d'éclairage[22].

En ce qui concerne la reconnaissance des signaux, il a été observé que la méthode SVM (Support Vector Machine) est une méthode d'apprentissage automatique très efficace dans ce cas [22].

Cette méthode présente de nombreux avantages tels qu'il existe une multitude d'implémentations déjà disponibles, l'inconvénient de cette méthode est que peut être très complexe et coûteuse avec un nombre élevé de classes[22].

L'autre méthode de classification, qui a fait l'objet de recherches approfondies dans les années passé de reconnaissance, est les réseaux de neurones. Cette méthode a démontré ses atouts pour des tâches telles que la classification, la localisation et la détection des images[22].

On peut dire que c'est la méthode la plus puissante innovant pour réaliser des algorithmes de reconnaissance en vision artificielle. Le principal inconvénient réside dans la nécessité d'un matériel très puissant, actuellement il existe des techniques novatrices de reconnaissances d'objet de formes libres ont été développées. Ces méthodes se basent sur l'extraction des points caractéristiques et le calcul de descripteurs invariants pour permettre une reconnaissance robuste et fiable de toute forme géométrique. Tel que : SURF, SIFT ... etc. Chacun des articles publiés sur ce sujet[23][24][25]possèdent leurs propre approche qui diffère de l'autres soit pour les étapes de la partie de détection ou la partie de reconnaissance. Un seul article [26] ressemble d'une manière générale, à notre approche mais il ya des différences dans les détails des étapes. Il est important de mentionner que le code source des applications existantes n'est pas disponible, aussi une haute sécurité couvre les mé-

moires contenant ce genre de logiciel, ça peut aller jusqu'à l'auto-effaçage en cas de tentative d'accès à cette mémoire. En plus, l'exclusivité de notre application est qu'elle est dédiée au code de la route de l'Algérie.

3.3 Les panneaux de signalisation routière

La signalisation routière constitue un élément fondamental de tout système de circulation. Les panneaux de signalisation routière sont généralement implantés de part et d'autres sur nos routes, indiquent les règles de la circulation établies pour permettre aux véhicules et aux piétons de se déplacer en toute sécurité sur les routes [27].

Les usagers de la route doit d'être bien sensés de la signalisation routière qui demeure aujourd'hui un des éléments phares de la prévention routière. Grâce aux signaux routière, les conducteurs sont informés des règles d'avertissements sur les dangers pouvant apparaître sur la route ou tout type d'informations intéressant le conducteur, parmi les quelles on peut citer ces exemples : [27]



FIGURE 3.3 – Exemples des dangers pouvant être signalés grâce aux signaux routière



FIGURE 3.4 – Exemples des indications fournies par les panneaux sur les directions à suivre

2. Panneaux d'avertissement de danger

Les signes de danger ont pour mission d'indiquer la nature d'un danger, leur objectif se conformer aux règles de comportement et d'éviter les chocs éventuels lors de la conduite[28]. Sa forme est triangulaire avec un bord rouge, comme indique la figure suivante.



FIGURE 3.7 – Signaux d'avertissement de danger

3. Panneaux de priorité

Ce sont destinées à informer les usagers de la route aux règles de priorité spéciales aux intersections ou aux passages étroits[?]. À l'intérieur de cette classe, nous pouvant trouver deux des signes les plus importants qui existent « stop » et « Cédez le passage » [28]. Comme vous pouvez le voir (Figure 3.8), ils n'ont pas de formulaire ou une couleur spécifique.

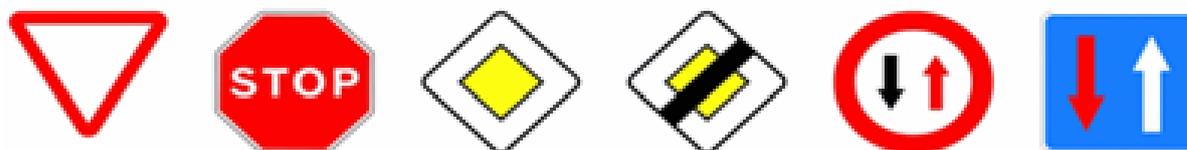


FIGURE 3.8 – Signaux de priorité

4. Panneaux de fin de l'interdiction ou de la restriction

Ces panneaux de fin d'interdiction ou de la restriction signalent que l'interdiction ou de la limitation est terminée[28]. Ils sont ronds et blancs avec une ligne diagonale noire (Figure 3.9).



FIGURE 3.9 – Signaux de fin d'interdiction ou de restriction

5. Panneaux des obligations

Ils sont destinés à obliger les usagers à respecter certaines prescriptions et certaines règles à partir du niveau du panneau ou devant celui-ci. Ces panneaux peuvent donc indiquer non seulement une direction que l'utilisateur devra suivre, mais aussi une vitesse minimum à respecter, ou encore une voirie obligatoire pour certains types d'utilisateurs[28]. Ils sont ronds avec un couleur bleu.



FIGURE 3.10 – Signaux d'obligations

3.4 Mise en œuvre (Conception et développement du projet)

L'objectif principal de ce projet est la réalisation d'une application pour la détection et la reconnaissance des panneaux de signalisation en temps réel, et pour cela nous avons suivi l'approche suivante :

3.4.1 Description de l'approche

Dans le but de réaliser ce projet de fin d'étude, nous proposons une approche basée sur les principales étapes ci-dessous.

1. Étape de prétraitement de l'image et détection des panneaux de signalisation, qui comprend les sous étapes suivantes :
 - Appliquer le filtre médian pour supprimer le bruit de l'image entrée.
 - Les images sont converties de modèle Rouge, Vert et Bleu (RVB) vers le modèle teinte, saturation et valeur (HSV) pour la segmentation.
 - Appliquer le filtre de lissage gaussien pour minimiser des faux résultats dans la détection.
 - Détection basée sur la couleur (isoler juste les couleurs qui nous intéressent le bleu et le rouge).
 - Détection basée sur la forme (utilisation de la transformé de hough et la détection des contours).
2. Étape de reconnaissance des panneaux de signalisation (extraction de caractéristiques des points d'intérêts on applique l'algorithme SURF).

3.4.1.1- Schéma de description de la proche

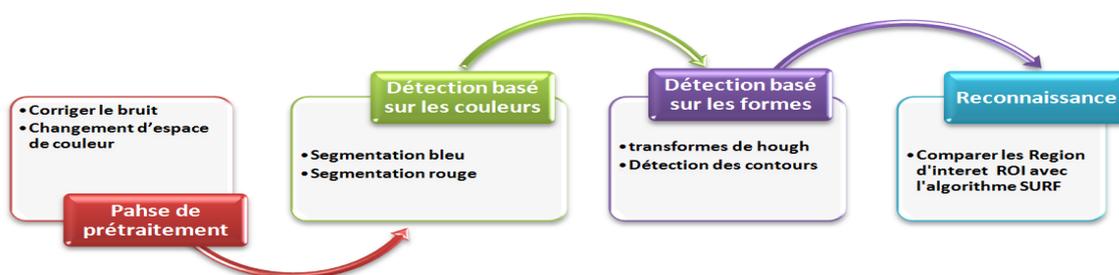


FIGURE 3.11 – Schéma de description de la proche

3.4.2 Le travaille avec OpenCV

OpenCV en Java est plus compliqué comparé à d'autres langages, tels que Python ou C ++, nous avons trouvé une grande difficulté de travailler à cause du manque de ressources en java-Opencv.

Ce qui suit est une description détaillée de tous les aspects analysés pour la mise en œuvre des algorithmes utilisés. Bien que les algorithmes soient déjà implémentés dans OpenCV, la difficulté des applications consiste à savoir comment utiliser ces outils pour optimiser le résultat final en tenant compte des différents paramètres et configurations possibles pour chacun des algorithmes et les critères à ajouter, ainsi que les temps de traitement.

Au début de notre démarche, nous avons utilisé Opencv 3 jusqu'à la partie de reconnaissance. Nous n'avons pas pu avancer à cause du manque des algorithmes dans le package Opencv 3 (Surf et Sift). Après des recherches, nous avons découvert que ces algorithmes sont des modules non libres et les auteurs de Surf et Sift exigent des droits de licence pour l'utilisation de leurs algorithmes originaux[29]. C'est pour ça nous avons décidé de revenir à une version ultérieure Opencv 2.4.13, où ces algorithmes sont libres pour des contextes académiques et de recherche.

3.4.3 Etape de prétraitement de l'image et de détection

Ce processus comprend les étapes suivantes :

3.4.3.1. Prétraitement de l'image

Cette phase a lieu juste après l'acquisition des images et a pour objectif d'améliorer la qualité de l'image en vue de sa segmentation. L'image brute peut inclure un bruit en raison d'un périphérique de capture moins efficace ou à cause d'un éclairage incorrect. Certaines parties de cette image peuvent avoir des contours nets ou des changements de couleur abrupts pouvant affecter le processus de détection[30][5]. Les techniques de prétraitements qu'on va utiliser sont :

3.4.3.1.1. Filtrage : Pour réduire le bruit ou adoucir l'image, nous avons utilisé deux types, médian et gaussien[30][5].

a - Filtre médian : très efficace pour éliminer le bruit du sel et du poivre. On utilise la fonction qui appartient à la classe **Imgproc** dans Opencv.

medianBlur(Mat src, Mat dst, int ksize) : filtrer L'image avec un filtre médian en faisant une convolution de l'image entrée, qui dans ce cas est Mat src, enregistrez-la à la sortie dans Mat dst. Le paramètre ksize est la taille du noyau doit être un entier impair positif[30][5].

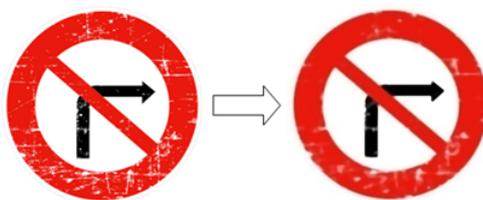


FIGURE 3.12 – Exemple de filtre médian

b - Filtre gaussien : pour le lissage des bords aiguisés, éliminer du bruit existant et diminuer les faux résultats. Dans notre cas, nous avons appliqué ce filtre au début des étapes de détections et pendant ces derniers, avec la fonction qui appartient à la classe **Imgproc** dans Opencv[30][5].

GaussianBlur (Mat src, Mat dst, Taille ksize, double sigmaX, double sigmaY) : filtrer L'image avec un filtre gaussien en faisant une convolution de l'image d'entrée, qui dans ce cas est Mat src, enregistrez-la à la sortie dans Mat dst. Le paramètre ksize est la taille du noyau gaussien, tandis que sigmaX et sigmaY sont les écarts types dans les directions x et y respectivement[30][5].



FIGURE 3.13 – Exemple de filtre gaussien

3.4.3.1.2. Transformation de l'espace colorimétrique de RVB vers HSV

Nous avons décidé de travailler avec cette espace de couleur car il est plus résistant aux changements d'éclairage et les étendues des couleurs sont plus claires dans différentes conditions d'éclairage[5].

Espace colorimétrique HSV

Bien souvent les couleurs sont codés en Rouge Vert Bleue on peut représenter les couleurs par le format HSV. L'espace HSV (Hue, Saturation, Value) ou TSV (Teinte Saturation Valeur) est un espace colorimétrique, défini en fonction de ses trois composantes[5] :

- **Teinte (H)** : le type de couleur. La valeur varie entre 0 et 360.
- **Saturation (S)** : l'« intensité » de la couleur. La valeur varie entre 0 et 100%. Plus la saturation d'une couleur est faible, plus l'image sera « grisée » et plus elle apparaîtra fade, il est courant de définir la « dé saturation » comme l'inverse de la saturation.
- **Valeur (V)** : la « brillance » de la couleur, elle varie entre 0 et 100%. Donc l'espace de couleur HSV sépare les informations de couleurs en teinte, saturation et valeur et ca nous aident à mieux séparer les couleurs. Dans OpenCV, la valeur H est normalisée en 0-180, les valeurs S et V sont normalisées en 0-255[5].

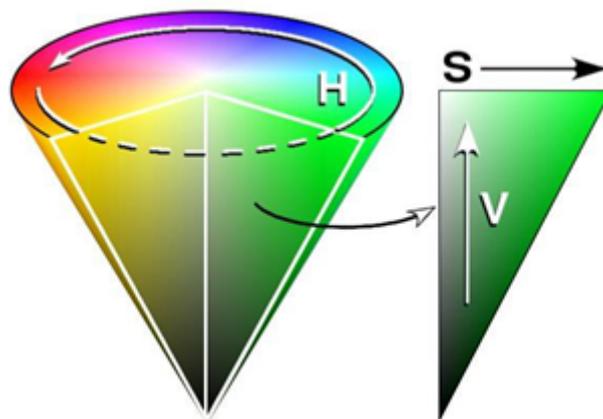


FIGURE 3.14 – Espace colorimétrique HSV[5]

* Pour effectuer la conversion de RVB vers HSV, OpenCV propose une classe appelée `Imgproc` dans laquelle il existe un certain nombre de méthodes et d'identificateurs. Dans ce cas, nous allons utiliser la méthode suivante[5].

`cvtColor (src mat, mat dst, code int)` : où `src mat` est la matrice d'image entrée, `mat dst` est la matrice d'image destinée où l'image est stockée une fois convertie, et enfin le code de la conversion à effectuer, dans ce cas, il s'agit d'un `src mat` RVB et lorsqu'on souhaite un passage dans l'espace HSV, le code sera `RGB2HSV`[5].



FIGURE 3.15 – Image entrée en RVB

Dans cette image, on observe comment l'image en entrée est effectuée dans un espace colorimétrique RVB, qui est ensuite converti comme déjà expliqué à HSV et on obtient l'espace de couleur suivant.



FIGURE 3.16 – Image sortie en HSV

3.4.3.2. Détection basée sur les couleurs

Pour arriver à l'analyse des informations qui nous intéressent, la couleur est un caractère fondamentale pour la distinction des objets, et dans le cas des panneaux de signalisation encore plus, car il permet de ne pas seulement l'identification d'un signal par rapport à d'autres objets, mais aussi la différenciation entre certains types de signaux.

* Segmentation des couleurs rouge et bleu

Une fois que nous avons l'image en HSV, il est nécessaire de déterminer la couleur à rechercher.

Les bordures des panneaux d'interdiction et les panneaux de danger sont des bordures rouge dans le contour et pour les panneaux d'obligation et les panneaux d'indication les bordures sont bleues qui aide à détecter ceux de ce type.

Pour trouver les couleurs, la méthode suivante a été utilisée[30] :

inRange (Mat src, Scalar lowerb, Scalar upperb, Mat dst) : Un seuil d'une couleur est défini pour déterminer les pixels souhaités la sortie de cette méthode est une matrice binaire qui donne 1 pour le pixel de couleur qu'il est inclut entre le seuil spécifié et 0 tous ceux qui ne le sont pas[30].

De cette façon, nous avons obtenus une matrice pour travailler contient uniquement les pixels de la couleur recherchée.

Concernant les paramètres de cette méthode, nous travaillons avec une matrice d'entrée précédemment convertie en HSV, deux vecteurs scalaires précédemment défini, l'un pour la valeur haute et l'autre pour la valeur basse (valeurs exprimées pour l'espace couleur HSV), et enfin le Mat dst qui est la matrice destination ou le résultats sera stockés après cette méthode[30].

1. Seuil utilisé pour la couleur rouge :

```
Core.inRange(hsv, new Scalar(0, 50, 100), new Scalar(10, 255, 255), lowredimg);/  
Core.inRange(hsv, new Scalar(160, 50, 50), new Scalar(179, 255, 255), upredimg);
```

2. Seuil utilisé pour la couleur bleu :

```
Core.inRange(hsv, new Scalar(75,150, 50), new Scalar(130, 255, 255), intervalbleu1);
Core.inRange(hsv, new Scalar(75,100, 100), new Scalar(129, 200, 200), intervalbleu2);
```

Dans la figure suivante (Figure 3.17), nous allons voir les résultats obtenus lorsqu' on applique ce traitement sur une image (a) en RVB un filtre de couleur pour la couleur bleu (c) et pour la couleur rouge (d) bien sur tous d'abord nous passant à l'espace de couleur HSV(b) :

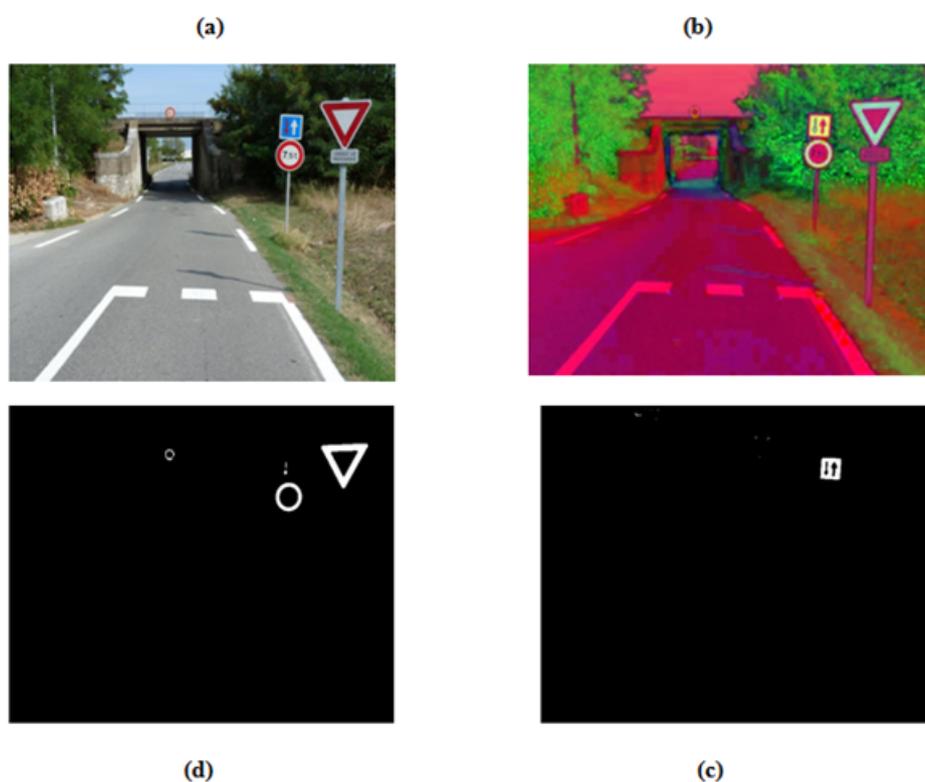


FIGURE 3.17 – Exemple de segmentation bleu et rouge

* Remarque

On peut voir sur la (Figure 3.18), la détection basée sur les couleurs (la segmentation par couleurs) donne toutes les informations sur les objets en rouge ou en bleu présentes dans une image qui n'a aucun rapport avec les panneaux recherchés.



FIGURE 3.18 – Détection des objets rouges qui ne présente pas forcément des panneaux

Pour cette raison, nous devons classifier une image en fonction de sa forme pour une détection plus précise.

3.4.3.3. Détection basée sur la forme

La détection basée sur la forme permet de détecter une information de forme particulière. C'est-à-dire, des objets circulaires, triangulaire, rectangulaires, ou octogonaux.

3.4.3.3.1. Identification des panneaux circulaires

Une fois il n'y a que des pixels blancs qui correspondent à bord de la couleur recherchée, il est temps de chercher des cercles pour pouvoir trouver des panneaux de signalisation rond (les interdictions qui correspondent à ceux qui ont le cadre rouge, ou les obligations qui sont les bleus).

* Une étape avant la détection des cercles consiste à appliquer un filtre gaussien .

* OpenCV possède une méthode qui appartient à la classe `Imgproc` appelée `HoughCircles`.

HoughCircles (`image Mat`, `cercles Mat`, `méthode int`, `double dp`, `double minDist`, `double param1`, `double param2`, `int minRadius`, `int maxRadius`) :[30]

Cette fonction recherche les cercles dans une image d'entrée avec échelle de gris. Dans notre cas, cette image d'entrée est celle de la sortie du filtre gaussien et correspond au paramètre d'image `Mat`, les autres paramètres importants à décrire sont

les paramètres `Mat cercle` qui est un vecteur dans lequel les cercles trouvés sont stockés, `minDist` qui détermine la distance minimale entre les centres de deux cercles trouvés, `MinRadius` et `MaxRadius` qui déterminent le rayon minimum et maximum qu'ils doivent avoir les cercles recherchés.

Une fois que le programme a été capable de trouver les cercles, on peut dessiner ces cercles. Pour cela, la classe `Core` a été utilisée et à l'intérieur de celui-ci à la fonction de cercle.

Cercle (`Mat img`, `centre du point`, `rayon int`, `couleur scalaire`, `épaisseur int`) : [30]

Cette fonction trace un cercle dans l'image désirée, qui dans ce cas est le `Mat img` qu'il correspond à celui que nous souhaitons voir apparaître sur l'écran, Pour que la fonction dessine le cercle il est nécessaire de déterminer le centre, le rayon, la couleur et l'épaisseur du cercle dessiné. Le centre et le rayon sont obtenus à partir des paramètres des cercles obtenus et enregistré dans les cercles `Mat` de l'étape précédente, tandis que la couleur et l'épaisseur sont peut être défini dans l'énoncé de la fonction.



FIGURE 3.19 – Exemple des cercles bleu et rouge détecté

3.4.3.3.2. Détection d'autres panneaux :

Pour chercher d'autres types de panneaux de signalisation, par exemple triangulaire ou rectangulaire, nous avons utilisé la méthode de contour.

a- Détection de contour :

Cette étape consiste à récupérer les contours d'image, c'est la méthode de segmentation la plus simple pour séparer les zones d'une image correspondant aux objets que nous souhaitons analyser. Cette séparation est basée sur les points où la différence d'intensité d'un pixel et d'un autre dépasse un certain seuil[31].

Pour différencier les pixels qui nous intéressent des autres (qui seront finalement rejetés), nous effectuons une comparaison de chaque valeur d'intensité de pixel par rapport à un seuil (déterminé après plusieurs tentatives)[31].

Donc pour la détection des contours de l'image, la première étape est prendre l'image segmentée par couleur précédemment ensuite réaliser un seuillage, pour améliorer la détection de contours, avec la fonction `threshold` appartient à la classe `Imgproc` de `OpenCV` [30] :

```
threshold(Mat bleurouge, imagethresho, 100, 255,  
Imgproc.THRESH_BINARY) :[30]
```

Un seuil est défini pour déterminer les pixels souhaités, la sortie de cette méthode est une matrice binaire qui donne 1 pour les pixels inclut entre le seuil spécifié et 0 pour tous ceux qui ne le sont pas.

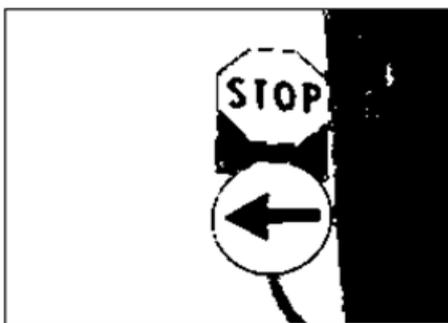


FIGURE 3.20 – Image binarisée

* Une étape avant la détection des contours : appliquer un filtre gaussien (figure 3.21)

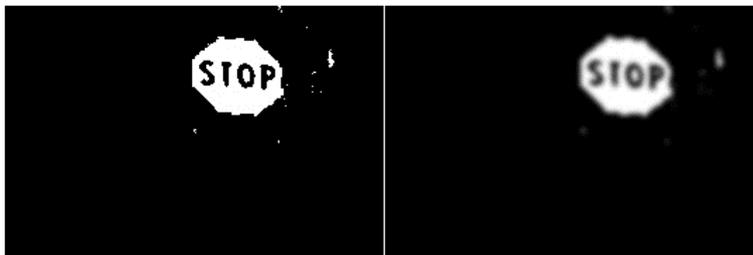


FIGURE 3.21 – Image binarisée sans et avec filtrage gaussien

* En suite, on va trouver les contours avec la fonction `findContours` appartient à la classe `Imgproc` de `OpenCV`.

```
findContours(imagebinarise, contours, hierarchy, Imgproc.RETR_TREE,  
Imgproc.CHAIN_APPROX_SIMPLE, new Point(0, 0)) :[30]
```

En entrée, `findContours` prend l'image binarisée, et renvoie les contours dans une variable `contours` pour stocker les points formant les contours trouvées. **hierarchy** est le vecteurs de contours qui permet d'obtenir la hiérarchie des contours. Les autre paramètres sont le mode de récupération des contours (par exemple les contours extérieur seulement ou comme dans notre cas récupérer tous les contours organisée chaque contour avec leur hiérarchie), et la méthode d'approximation de contour (soit stockés tous les points ou seulement les extrémités nous avons choisi la dernière pour économiser la mémoire). Le dernier paramètre est le point de départ qui constitue le premier point du contour. Par défaut, c'est le point de coordonnée (0,0)[31][30].

Maintenant, nous devons voir si l'un des contours ressemble à un triangle ou rectangle ou octogone, pour cela nous suivent les étapes suivantes :

1. Calculez le périmètre du contour avec la fonction **Imgproc.arcLength()** en lui donnant comme paramètres le contour avec lequel le périmètre doit être calculé et en second paramètre « True » si le contour est fermée ou non[30].
2. Rapprocher les contours on utilise la fonction **Imgproc.approxPolyDP()** avec trois paramètres. Le premier est le contour auquel nous voulons appliquer l'approche, le second est la déviation maximale par rapport au périmètre effectuer, par exemple, si vous souhaitez convertir une courbe en une ligne,

la distance n'est pas la même. Pour ce faire, le périmètre a été calculé ainsi qu'un pourcentage (presque 1% dans notre cas) du périmètre du contour. Plus ce pourcentage est grand, moins l'approximation sera précise. L'approximation sert à diminuer le nombre de points sur un contour. En d'autres termes, nous allons essayer de décrire les contours par un ensemble de segments[30].

3. Une fois que nous avons l'approximation correspond au nombre de segments et donc au nombre de côtés qui constitue la forme.

Ainsi :

- Si le nombre de côté égale à 3, il s'agit d'une forme triangulaire.
- Si le nombre de côté égale à 4, la forme de type rectangulaire.
- Si le nombre de côté égale à 8, il s'agit d'une forme octogonale.

4. Créez un rectangle autour du coté (point) du forme et découpez-le avec les fonctions :

rect = `Imgproc.boundingRect(points)`.

extract = `src.submat(rect)` : (Extrait une sous-matrice rectangulaire de la matrice source.)



FIGURE 3.22 – Exemple de détection des contours de formes déferent

b- Résultats et remarques :

- Lors de la recherche du forme à 8 côtés (pour trouver le panneau de stop) nous avons remarqué que la détection fait aussi pour les cercles (figure 3.22) donc nous avons décidé de travailler seulement avec la méthode de détection de contour et créer deux classes : une classe pour traiter les contours rouges (avec 3 et 8 côtés) et autre pour les contours bleu (avec 4 et 8 côté).
- Nous avons remarqué aussi qu'il ya des faux contours détectés (ciel, feux des voitures, trottoirs). Nous avons essayé de corriger cette erreur par trouver les meilleurs seuils pour les couleurs bleus et rouges et limiter la taille des formes extraites pour éliminer les petite ou les grandes formes détectées. Mais nous avons réussi seulement à minimiser le taux d'erreur, sans oublier que la qualité des images a un rôle très important dans la détection, et notre images de tests sont des captures prise à partir des séquences vidéo qui ont été enregistrées de l'intérieur du voiture avec un mobile simple.



FIGURE 3.23 – Exemple de détection des faux contours bleu et rouge

* Maintenant, il faut savoir à quel panneau correspondent les formes extraites, pour cela on passe à l'étape de reconnaissance.

3.4.4 Etape de reconnaissance

Une fois que les formes (cercle, triangle, rectangle, octogone) est recherché dans les images entrée on passe à l'extraction des régions d'intérêt de ces formes pour effectuer l'identification ultérieure par des caractéristiques communes des deux images, et cela en comparant leurs « caractéristiques » avec celles d'un ensemble de panneaux de référence stockés dans une base de données.

Pour ce faire, il y a tellement de techniques de reconnaissance, la bibliothèque OpenCV offrent une grande variété des algorithmes comme SIFT, SURF, ORB,...etc. Dans notre travail pour l'extraction des caractéristiques nous avons choisi de travailler avec la méthode SURF grâce à l'invariabilité à l'échelle et de rotation.

3.4.4.1- La méthode SURF

Pour l'identification d'objet, nous avons utilisé une technique basée sur les descripteurs locaux invariants : la méthode SURF [Bay et al., 2008]. SURF (Speeded Up Robust Features) est un algorithme robuste d'extraction de points d'intérêt qui calcule les descripteurs locaux invariants associés à ces points caractéristiques. L'avantage majeur de cet algorithme est sa rapidité par rapport à d'autres techniques similaires utilisant les descripteurs locaux invariants telles que la méthode SIFT (Scale-Invariant Feature Transform) [Juan Gwun, 2009][32][33].

La méthode SURF est inspirée de l'algorithme SIFT qui est le précurseur dans le domaine d'extraction des points invariants. Toutefois, l'algorithme SIFT présente l'inconvénient d'être très lent et donc, il n'est pas adapté aux applications temps-réel. SURF et SIFT sont tous les deux des algorithmes robustes bases sur les descripteurs locaux invariants[32][33].

Le principe général de ces deux méthodes est le suivant :

1. Détecter les points d'intérêts.
2. Définir une région d'intérêt autour de chaque point caractéristique.
3. Calculer les descripteurs locaux à partir des régions normalisées.
4. Mise en correspondance des descripteurs[32][33].

3.4.4.2- Détection, extraction et comparaison des caractéristiques

La première chose à comprendre est le concept de "**caractéristiques**" en anglais "**features**" d'une image. Une caractéristique est un point d'intérêt de l'image, qui peut être des structures, telles que des points, des angles, bords, ou résultats de traitements appliqués à une région partielle ou sur le total de l'image, tels que coins, max, min, sommet, direction du gradient, Les points d'intérêts sont des points qui contiennent beaucoup d'informations relativement à l'image[34].

Pour qu'une caractéristique soit considérée comme bonne ("**good feature**"), c'est-à-dire utile, il est nécessaire de répondre à certaines exigences : la propriété principale qu'ils doivent avoir est l'invariance. C'est-à-dire la capacité à conserver les mêmes caractéristiques, la même « **signature** » pour un type de transformation. La nature de ces transformations peut être géométrique. Il s'agit des transformations classiques telles que la rotation, le changement d'échelle ou la translation.

Il peut également s'agir de l'invariance à la photométrie, i.e. L'intensité lumineuse[34].

La localisation des "caractéristiques" d'une image ou d'une région est obtenue par un détecteur de caractéristiques "**feature detector**"[34].



FIGURE 3.24 – Exemple de points d'intérêts détecté

Les informations décrivant ces caractéristiques ou points d'intérêt sont obtenues par un extracteur de descripteur "**descriptor extractor**" et sont stockées dans des vecteurs multidimensionnels appelés descripteurs "**descriptors**"[34].

Il existe différents algorithmes pour obtenir les descripteurs, avec différentes manières de décrire et de coder l'information (valeurs réelles, binaires ...). La dernière étape consiste à comparer les deux images ou régions. Ça se fait avec des comparateurs de descripteurs ou "**descriptor matchers**", en conséquence des paires de descripteurs (une pour chaque image) sont obtenues avec plus grande similitude entre eux [34].

Chaque paire est composée de deux vecteurs, l'un de la première image et l'autre est le vecteur de la seconde image avec l'indice de similarité le plus élevé de tous les descripteurs de la deuxième image[34].

Il existe différentes façons de calculer la similarité entre deux vecteurs, tels que "**La distance euclidienne**" pour les valeurs réelles ou "**La distance de Hamming**" pour les valeurs binaires, de cette manière, l'attribut distance est introduit : l'indice de similarité qui ont deux paires de vecteurs[34].

De la même manière que la bibliothèque OpenCV fournit une classe de type Mat pour stocker des images, il propose aussi des sous classes de la classe Mat :

MatOfKeyPoints :qui permettent de stocker les caractéristiques détectées (les points clés).

MatOfDMatch :ou les paires de descripteurs associées "**matches**" sont stockés.

L'organigramme suivant montre le processus de détection, d'extraction et de comparaison des caractéristiques à partir de deux images initiales. Il est possible que l'image qui est utilisé comme modèle ne nécessite aucun type de traitement supplémentaire[34].

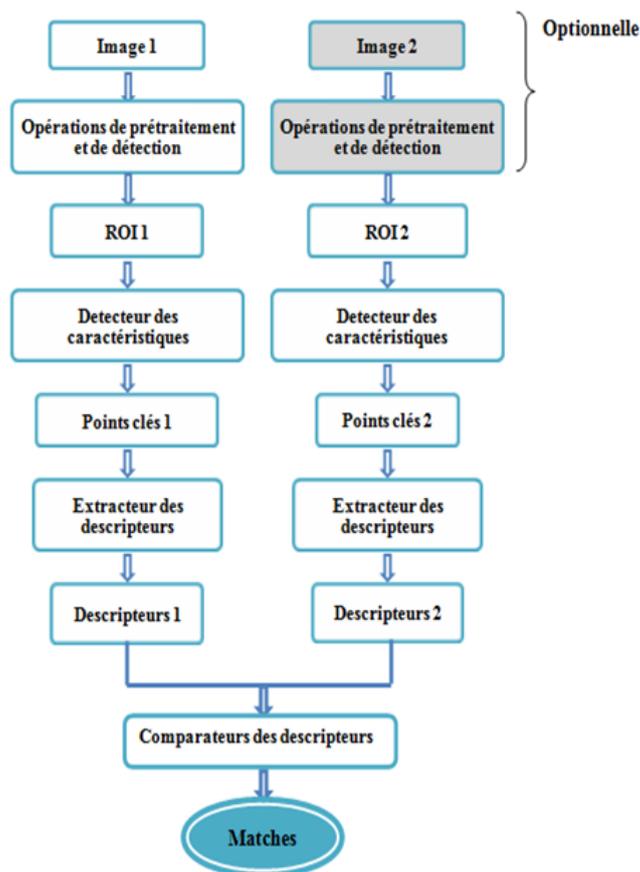


FIGURE 3.25 – Organigramme de détection, d'extraction et de comparaison des caractéristiques

Comme déjà mentionné ci-dessus, les comparateurs de descripteurs associer les paires de descripteurs de la première image avec son descripteur le plus similaire de la deuxième image, mais cela ne signifie pas qu'elles sont vraiment similaires.

Donc la dernière étape consiste à obtenir ce qu'on appelle les **"good matches"** ou des bonnes paires des descripteurs. La distinction entre les bons et les mauvais matchs est établie par le programmeur, et c'est la quantité de **«good matches»** qui nous a permis de décider s'il y a une "correspondance" ou non.

Dans notre cas, la distance minimale entre toutes les paires de descripteurs est calculée.

Une fois la liste des **"good matches"** obtenue, la correspondance ce fait a partir des **« good matches »**.

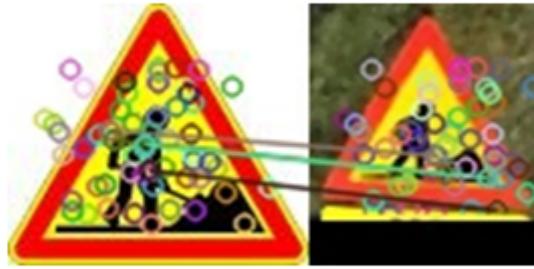


FIGURE 3.26 – Comparaison des caractéristiques

3.4.4.3- Diagramme UML

Pour la création du diagramme UML on a utilisé un plugin sur Netbeans s'appelle "easyUML". Avec ce plugin nous allons créer automatiquement un diagramme UML de notre projet, la classe « **mainclass** » est la classe principale de notre application.

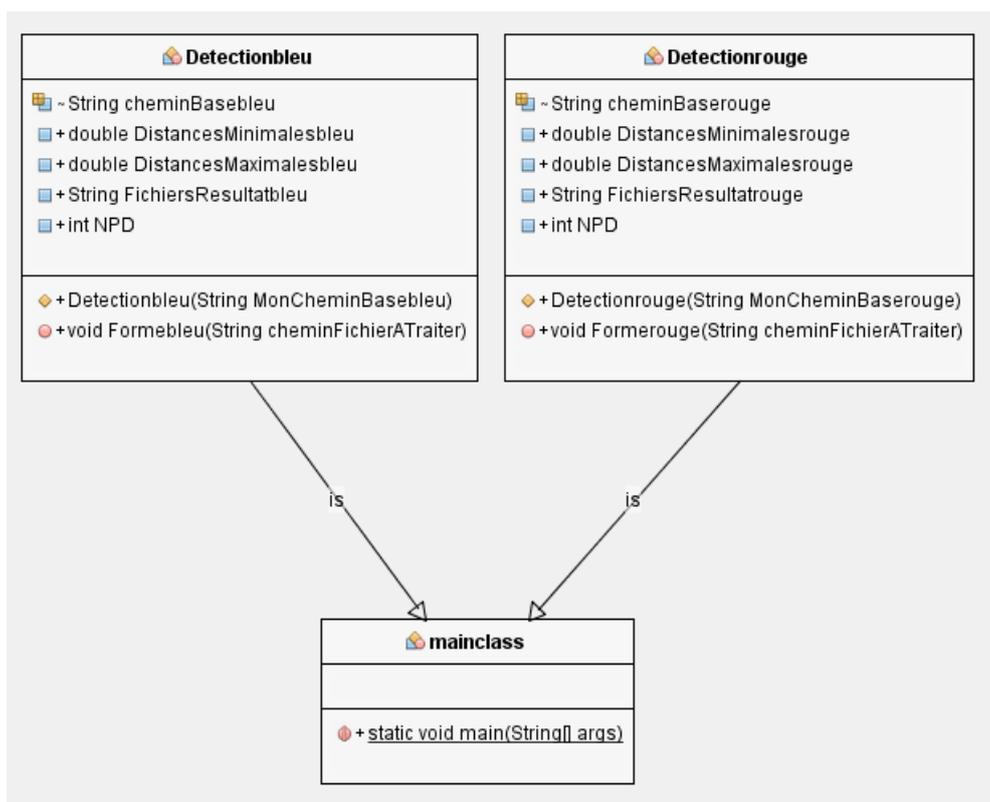


FIGURE 3.27 – Diagramme UML de notre Application

3.4.4.4- Phase de tests

Les tests de cette partie sont réalisés avec les matériels suivants :

Matériel utilisé			
Modèle	Processeur	RAM	Système d'exploitation
LENOVO Think Pad X201 Tablet	Intel (R) Core (TM) i7 CPU L 640 @ 2.13 GHz	4.00 Go	Windows 7 Professionnel 64 bits
HP Notebook	Intel (R) Core (TM) i3-6006U CPU @ 2.00 GHz	4.00 Go	Windows 8.1 Professionnel 64 bits

TABLE 3.1 – Description du matériel utilisé pour les tests

- Les tableaux suivants montrent les résultats obtenus après testé de 80 images avec les 2 algorithmes SURF et SIFT.
- La base de données utilisée dans ce projet contient 145 images la majorité avec une résolution 100x100 pixels.

Bonne détection : Le nombre de bonne extraction des panneaux de signalisation routière a partir des images de teste.

Bonne reconnaissance : La bonne identification des panneaux extraite lors de la comparaison avec les images de la base de données.

P.S.R : Panneaux de Signalisation Routière.

Taux globale=(Nombre des images X 100)/Nombre des images de teste (5)

	Nombre des images	Taux globale
Bonne détection du P.S.R	71	88,75%
Bonne reconnaissance du P.S.R	69	86,25%
Faux détection du P.S.R	9	11,25%
Faux reconnaissance du P.S.R	11	13,75%

TABLE 3.2 – Résultats des tests avec SURF

	Nombre des images	Taux globale
Bonne détection du P.S.R	62	77,50%
Bonne reconnaissance du P.S.R	57	71,25%
Faux détection du P.S.R	18	22,50%
Faux reconnaissance du P.S.R	23	28,75%

TABLE 3.3 – Résultats des tests avec SIFT

PC	Temps d'exécution	Avec SURF	Avec SIFT
	LENOVO Think Pad X201 Tablet		1s à 2s
HP Notebook		1s à 2s	4s à 5s

TABLE 3.4 – Temps d'exécution avec SURF et SIFT

Alors :

Le taux de reconnaissance avec l'algorithme SURF (86,25%) est meilleur que SIFT (71,25%) ainsi l'algorithme SURF est plus rapide que SIFT, de cette raison on a décidé de choisir l'algorithme Surf dans notre travail.

- Nous avons constaté une difficulté d'identification de quelques panneaux.
- Dans le cas de similarité de panneaux (Figure 3.28).

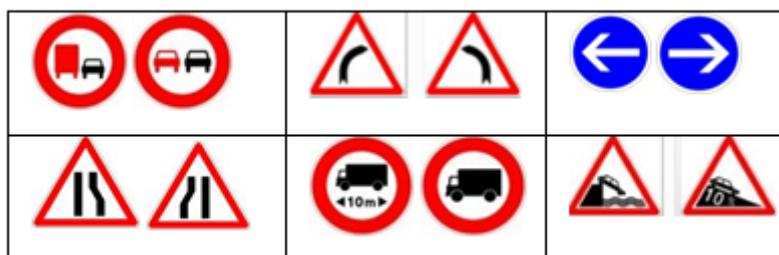


FIGURE 3.28 – Panneaux similaire

- Ou dans le cas où il ya des obstacles (Figure 3.29).



FIGURE 3.29 – Cas des obstacles

3.4.4.5- Les interfaces de l'application

- Dans notre présentation de l'application, on utilise trois types d'affichage de la reconnaissance soit par image, texte, ou avec son.

1. Page de garde



FIGURE 3.30 – Page de garde

2. Fenêtres d'authentification et d'accueil



FIGURE 3.31 – Page d'authentification



FIGURE 3.32 – Page d'accueil

3. les Fenêtres de Roconnaissance

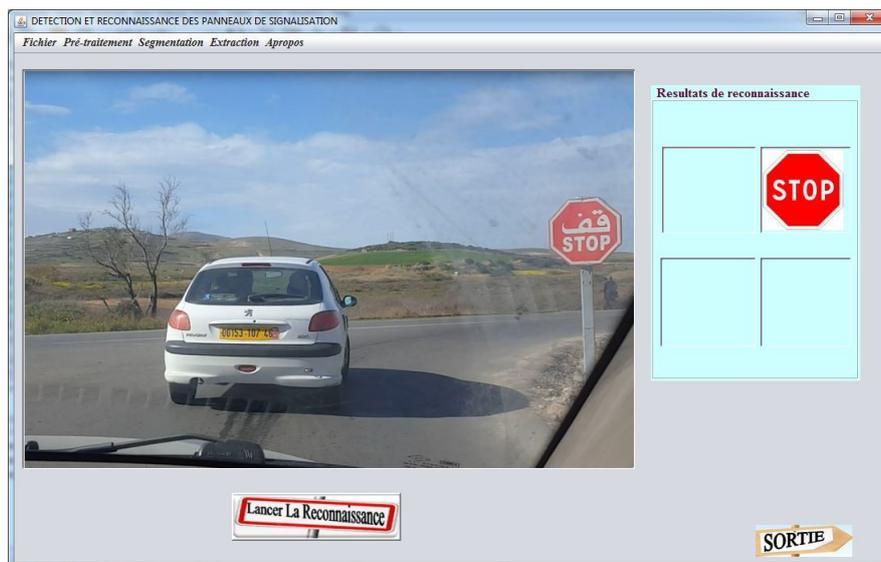


FIGURE 3.33 – Roconnaissance Rouge

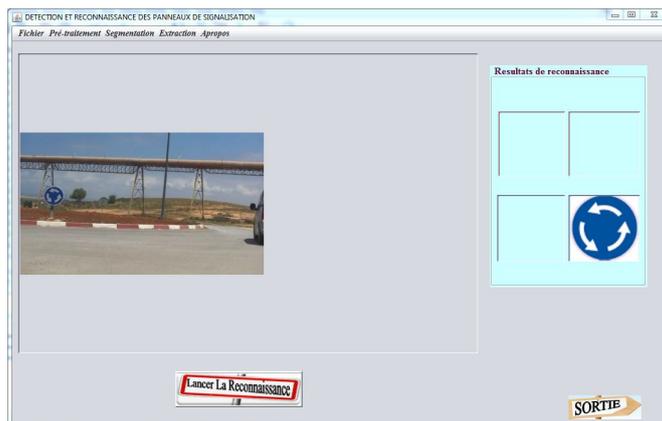


FIGURE 3.34 – Roconnaissance Bleu

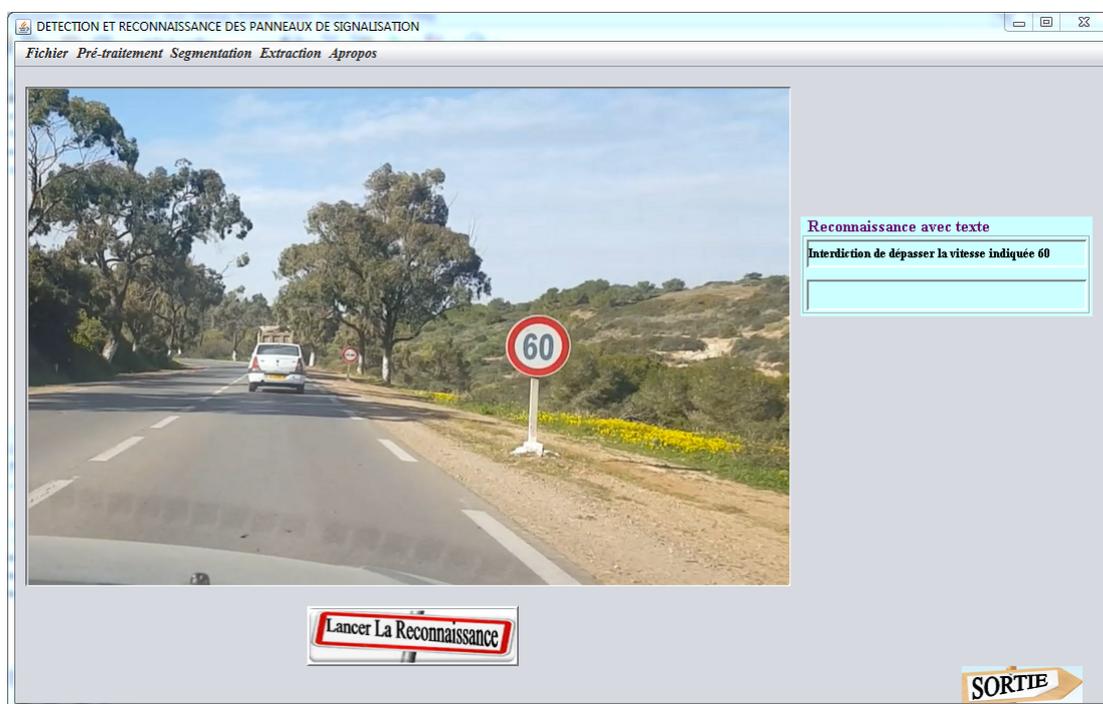


FIGURE 3.35 – Roconnaissance avec texte

3.5 Conclusion

Dans ce chapitre nous avons présenté les démarches suivies pour l'élaboration d'un outil de détection et de reconnaissance des panneaux de signalisation, notre système inclut presque tous les types de panneaux, avec l'utilisation d'un descripteur SURF qui nous a permis de gagner du temps, on a conclu aussi que l'amélioration de l'étape de détection donne de bons résultats et augmente le taux de réussite.

Conclusion et perspectives

Dans les systèmes modernes de sécurité des véhicules, de nombreuses technologies pourraient nous aider à conduire et informer le conducteur en cas de danger.

La capacité à surveiller en permanence les panneaux de signalisation de restrictions et les avertissements sur la route, conduit à ce que le conducteur est souvent distrait du contrôle du véhicule. Ainsi, cela augmente les risques d'accident.

La solution consiste à développer activement des systèmes de détection et de reconnaissance des panneaux de signalisation pour informer le conducteur.

Les systèmes existants sont des systèmes informatiques complexes, trop coûteux à mettre en œuvre, et leur code ni gratuit ni disponible. Pour réduire les coûts de mise en œuvre de ce type de systèmes, il est nécessaire de mettre au point des applications de détection et de reconnaissance des panneaux de signalisation.

À la suite de ce travail, nous avons développé une application de détection et de reconnaissance des panneaux de signalisation. Au début, il y avait un processus de mise en œuvre pour collecter les images de tests à partir d'une séquence vidéo capturée à l'intérieur de la voiture. Après, on applique un traitement primaire sur les images obtenues pour réduire le bruit. En suite, on applique des algorithmes pour la segmentation de couleur (transformation de l'espace de couleur de RVB vers HSV) et autres pour la segmentation des formes (détection des contours) dans le but de minimiser et localiser les régions à traiter (ROI) afin d'obtenir les couleurs et les formes souhaités. Une fois ces régions extraites, la dernière étape consiste à identifier de quel panneau il s'agit dans la partie de reconnaissance où on utilise le descripteur SURF parmi les nouveaux descripteurs qui montrent leurs efficacité dans les applications de temps réel.

Au cours du processus de développement de l'application, plusieurs techniques de traitement d'image existantes ont été testées, comme les filtres, la transformée de Haugh qui nous donne des bons résultats pour la détection des cercles, mais seulement certaines sélectionnées et incluses dans notre approche afin d'améliorer son efficacité. Le choix des techniques de notre approche était justifié par une analyse théorique suivi par des tests.

La mise en œuvre logicielle de l'application a été réalisée à l'aide de l'environnement de développement JAVA car cet environnement est multiplateforme avec l'utilisation de la librairie OpenCV 2.4.13 puisque il fournit un ensemble de packages conçus pour travailler avec la vidéo et les images, ainsi qu'une large gamme de filtres et des algorithmes de traitement des images. Nous avons réussi à obtenir les meilleurs résultats possibles pour le traitement des images.

Les résultats de cette approche montrent que l'application de détection et de reconnaissance des panneaux de signalisation est efficace et peut être appliquée plus tard à des systèmes de reconnaissance des panneaux de signalisation visant le code de la route d'Algérie, et peut être pour aider les gens dans les travaux publics et des transports pour détecter les panneaux en cas de changement. Bien que l'application contienne un taux (13,75%) de fausse reconnaissance découvertes lors des tests, celles-ci peuvent probablement être surmontées par des modifications. Ces modifications consistent à essayer d'autres algorithmes et calculs qui nous donne une meilleure précision des couleurs et des contours ça demande un large temps pour étudier et comprendre toutes les fonctions et essayer tous les cas possibles avec des différents algorithmes.

Durant ce projet, nous avons eu l'occasion de se rapprocher du monde de traitement d'images, ces outils, et découvrir l'importance de ce domaine actuellement.

Abréviation

RSR : Reconnaissance de la Signalisation Routière.

PPP ou, en anglais DPI : Point Par Pouce.

OpenCV : Open Source Computer Vision.

ADAS : Advanced Driver-Assistance Systems.

RVB : Rouge, Vert, Bleu.

HSV : Hue Saturation Value.

SURF : Speeded Up Robust Features.

SIFT : Scale-Invariant Feature Transform.

ROI : Region Of Interest.

Bibliographie

- [1] Mohamed SANDELI. Traitement d'images par des approches bio-inspirées application a la segmentation d'images. *Département Informatique Fondamentale Et Ses Applications, Faculté Des Nouvelles Technologies De L'information Et De La Communication*, 2014.
- [2] KARAM Fouad and IMOULOUDENE Salah Eddine. *Transfert sécurisé des données visuelles (images) dans un réseau intranet selon l'architecture client/serveur*. PhD thesis, Université Abou Bakr Belkaid– Tlemcen, 2015.
- [3] Djemaa BOUKHLOUF. *Résolution de problèmes par écosystèmes : Application au traitement d'images*. PhD thesis, Université Mohamed Khider-Biskra, 2005.
- [4] Sami TOUHAMI. Classification non supervisée de pixels d'images couleur par analyse d'histogrammes tridimensionnels. 2005.
- [5] BENSENOUCI Hanane and AHMED BRAHIM Asma. *DETECTION DES LESIONS LUMINEUSES DANS LES IMAGES RETINIENNES*. PhD thesis, Université Abou Bakr Belkaïd de Tlemcen, 2013.
- [6] Fabio LA ROCCA. Introduction à la sociologie visuelle. *Sociétés*, pages 33–40, 2007.
- [7] BENFRIHA Sarra and HAMEL Asma. *Segmentation d'image par Coopération rgion-contours*. PhD thesis, Université Kasdi Merbah-Ouargla, 2016.
- [8] MADJAOUI Amina and FARES Fadia. *Segmentation des Images par Contours Actifs : Application sur les Images Satellitaires à Haute Résolutions*. PhD thesis, Université Abou Bakr Belkaid– Tlemcen, 2012.
- [9] Rachida LAKHDARI. *La détection des micros calcifications dans l'image mammographie*. PhD thesis, Université Abou Bakr Belkaid– Tlemcen, 2011.
- [10] Moustafa BENALI. *Reconnaissance Automatique des Chiffres Manuscrits*. PhD thesis, Université Abou Bakr Belkaid– Tlemcen, 2017.

- [11] DAHMANI Sarra and HABIB Meryem. *La géolocalisation basée sur l'utilisation des systèmes d'informations géographiques*. PhD thesis, Université Abou Bakr Belkaid– Tlemcen, 2016.
- [12] Houassine CHARIF. *Segmentation d'images par une approche biomimétique*. PhD thesis, Université de Boumerdès-M'hamed Bougara, 2012.
- [13] Nicolas Vandembroucke and Ludovic Macaire. Porebski ,alice. 2019.
- [14] Aldjia BOUCETTA. *Etude de l'effet des Transformées de Décorrélation en Compression des Images Couleurs RGB*. PhD thesis, Université de Batna 2, 2010.
- [15] SID Zine El Abidine, AOUI Lyazid, and MOUSSA Semchedine. *Nouvelle Approche de filtrage Non-linéaire d'Images Numériques*. PhD thesis, Université Abou Bakr Belkaid– Tlemcen, 2011.
- [16] Bendaoud Mohamed HABIB. *Développement de méthodes d'extraction de contours sur des images à niveaux de gris*. PhD thesis, Université Mohamed Boudiaf des Sciences et de la Technologie-Mohamed Boudiaf . . . , 2017.
- [17] Fouzia HACHEMI. *La détection et suivi des objets en mouvement dans une scène vidéo en utilisant la bibliothèque OpenCV*. PhD thesis, Université Abou Bakr Belkaid– Tlemcen, 2017.
- [18] Yasmina BOUBEKEUR. *Identification automatique de mots clés dans les textes arabes*. PhD thesis, Université de Djilali BOUNAÂMA Khemis Miliana, 2016.
- [19] Mohamed Amir BENLOUCIF. *Coopération homme-machine multi-niveau entre le conducteur et un système d'automatisation de la conduite*. PhD thesis, Université de Valenciennes et du Hainaut-Cambresis, 2018.
- [20] Luke WROBLEWSKI. *Mobile first*. Editions Eyrolles, 2012.
- [21] Détection et classification automatiques des panneaux de signalisation. In *Huitième atelier international sur l'analyse d'images pour les services interactifs multimédias (WIAMIS'07)*.
- [22] LE TAM T, TRAN SON T, MITA Seichii, and NGUYEN Thuc D. Real time traffic sign detection using color and shape-based features. In *Asian Conference on Intelligent Information and Database Systems*, pages 268–278. Springer, 2010.
- [23] SOHEILIAN Bahman, ARLICOT Aurore, and PAPARODITIS Nicolas. Extraction de panneaux de signalisation routière dans des images couleurs. In *Reconnaissance des Formes et Intelligence Artificielle*, pages 1–8, 2010.

- [24] SHOPA P, SUMITHA N, and PATRA Psk. Détection et reconnaissance des panneaux de signalisation à l'aide d'opencv. In *Conférence internationale sur la communication de l'information et les systèmes intégrés (ICICES2014)*, pages 1–6, 2014.
- [25] LORSAKUL Auranuch and SUTHKORN Jackrit. Reconnaissance des panneaux de signalisation via réseau neuronal sur opencv : vers un système intelligent d'assistance véhicule / conducteur. In *4ème conférence internationale sur les robots omniprésents et l'intelligence ambiante*, pages 22–24, 2007.
- [26] KYU Htet Wai. Reconnaissance des panneaux de signalisation en temps réel à l'aide du descripteur surf. *Journal international de la recherche scientifique technologique*, 2016.
- [27] Philippe DA COSTA. *Sécurité routière et circulation : la responsabilité des différents acteurs*. Les éditions des journaux officiels, 2007.
- [28] ANCEAUX Françoise, RAJAONAH Bako, and LEI Paris V. Projet sari-predit 3. 2006.
- [29] *Apprentissage OpenCV : la vision par ordinateur avec la bibliothèque OpenCV*. 2008.
- [30] GARCIA Gloria Bueno, SUAREZ Oscar Deniz, ARANDA José Luis Espinosa, TERCERO Jesus Salido, GRACIA Ismael Serrano, and ENANO Noelia Vállez. *Learning Image Processing with OpenCV*. Packt Publishing Ltd, 2015.
- [31] Jérôme DA RUGNA. *De l'usage des méthodes bas niveau pour la recherche d'image par le contenu*. PhD thesis, Université Jean Monnet-Saint-Etienne, 2004.
- [32] Diarra MAMADOU. *Extraction et fusion de points d'intérêt et textures spectraux pour l'identification, le contrôle et la sécurité*. PhD thesis, 2018.
- [33] Jean CAMILLERAPP. Utilisation des points d'intérêt pour rechercher des mots imprimés ou manuscrits dans des documents anciens. In *CIFED 2012-colloque international sur l'écrit et le document*, 2012.
- [34] Surf : fonctionnalités robustes accélérées. In *conférence européenne sur la vision par ordinateur*, pages 404–417, 2006.