

كلية العلوم
ⵜⴰⵎⴻⵔⴰⵏ ⵜⴰⵎⴰⵏⴰⵢⵜ
FACULTÉ DES SCIENCES



UNIVERSITE IBN ZOHR FACULTE DES SCIENCES

Département Informatique
Filière Sciences Mathématiques et Informatique

PFE

**Présenté par : ABOUSALIM Aissame
JABY Assil**

Pour l'obtention de la
Licence en Sciences Mathématiques et Informatique

Détection des panneaux routiers

Soutenu le

Encadré par : Pr. LAHMYED Redouan

Année universitaire 2020-2021

Dédicaces

Au meilleur des pères
A ma chère Mère Fatiha
Dont le mérite, les sacrifices et les qualités humaines
M'ont permis de vivre ce jour.

A ma sœur Khaoula
A qui je souhaite un avenir radieux plein de réussite

A mes Amis
A tous ceux qui me sont chers

AISSAME ABOUSALIM

Je dédie ce modeste travail

À mes très chers parents
À toute ma famille et mes amis
À tous ceux qui m'ont encouragé et Soutenu ...

ASSIL JABY

Remerciements

Nous remercions en premier lieu le tout puissant de nous avoir donné la chance, le courage et les moyens pour l'accomplissement de ce projet.

Nous remercions cordialement, Notre encadreur Pr. LAHMYED Redouan, de nous avoir proposé ce thème, de nous avoir guidé et suivi durant l'accomplissement de ce travail.

Finalement, nous remercions toute personne ayant contribué à l'accomplissement de ce travail de prêt ou de loin et qui nous ont été bénéfique durant notre parcours pour la réalisation de ce travail.

Nous remercions l'ensemble des Enseignants du département d'informatique pour les efforts qu'ils fournissent pour notre réussite dans nos études.

Résumé

La vision par ordinateur (Computer Vision) est une branche principale de l'intelligence artificielle, elle consiste à faire comprendre à n'importe quelle system informatique ce qu'il capte à travers une caméra ou support visuel, cette technologie est bien utile dans plusieurs domaines tel que la sécurité, l'imagerie médicale, la réalité augmentée et - notre point d'intérêt – la sécurité routière.

Dans le cadre de la réalisation du projet de fin d'études pour l'obtention de la licence en sciences mathématiques et informatiques et du aux accroissements des taux d'accident routiers au cours des dernières années, nous somme livre a réalisé un system permettant la détection des panneaux routiers et de les encadrer pourquoi pas emmètre un son de tel façon à ce que le conducteur ne manque pas le panneau en se reposant sur les nouvelles technologies tel que la reconnaissance des formes et le Machine Learning.

Notre mémoire est structurée comme suit :

Dans le chapitre 2, nous présentons un état de l'art sur les méthodes de détection de panneaux routiers dans les images et d'apprentissage M.L. Nous évoquons aussi les caractéristiques visuelles les plus utilisées pour décrire ce type d'objet ainsi que les bases d'images disponible à ce sujet. Dans le chapitre 3 nous y présentons également les résultats de détection obtenus au travers des images test de scènes routières Enfin, nous terminons notre mémoire par une conclusion générale.

Mots-clés : Vision par ordinateur, apprentissage m.l., panneaux routiers, traitement d'image, intelligence artificielle, taux d'accidents.

Abstract

Computer vision is a main branch of artificial intelligence, it consists of making any computer system understand what it captures through a camera or visual support, this technology is very useful in many areas such as safety, medical imaging, augmented reality and - our focus - road safety.

As part of the completion of the end of studies project for obtaining the degree in mathematical and computer sciences and due to the increase in road accidents' rates over the past few years, we are demanded to realize a system allowing the detection of road signs and to frame them, and why not, emits a sound so that the driver does not miss the sign by relying on new technologies such as shape recognition and Machine Learning.

Our thesis is structured as follows:

In chapter 2, we present a state of the art on the methods of detecting road signs in images and machine learning.

We also evoke the visual characteristics most used to describe this type of object as well as the bases of 'images available on this. In chapter 3 we present the detection results obtained through test images of road scenes. Finally, we end our thesis with a general conclusion.

Keywords: Computer vision, machine learning, artificial intelligence, road panels, image processing.

Table des matières

Dédicaces	ii
Remerciements	iii
Résumé	iv
Abstract	v
Table des matières	vi
Liste des figures	vii
Liste des tableaux	viii
Liste des abréviations	ix
Introduction générale	1
Chapitre 1 : Introduction.....	3
1.1 Généralités sur le traitement d’images	3
1.2 Généralités sur Machine Learning	12
1.3 Objectif du projet	16
Chapitre 2 : Les méthodes proposées	17
2.1 Méthodes basée sur le traitement (Forme)	17
2.2 Méthodes basée sur l’apprentissage	24
Chapitre 3 : Résultats.....	29
3.1 Hardware and Software	29
3.2 Base de données	33
3.3 Comparaisons et résultats	34
Chapitre 4 : conclusion	37
Conclusion Générale	38
Références	39

Liste des figures

Figure 1. Accident routier	1
Figure 2. Image numérique	3
Figure 3. Dimension d'une image	4
Figure 4. Résolution versus 1/8 résolution	4
Figure 5. Image avec et sans bruit	5
Figure 6. Comparaison de l'histogramme de deux images	5
Figure 7. Luminance d'une image	6
Figure 8. Rapport du contraste	6
Figure 9. Contour de formes	7
Figure 10. Range du GrayScale	7
Figure 11. Les 3 couches d'une image RGB	8
Figure 12. Représentation de la teinte, saturation et valeur	8
Figure 13. Représentation de l'espace $L^*a^*b^*$	9
Figure 14. Principe du filtre médian	10
Figure 15. Principe du filtre maximum	10
Figure 16. Principe du filtre minimum	11
Figure 17. La différence entre un petit et un grand flou gaussien	11
Figure 18. Les types de problèmes de machine learning	14
Figure 19. Apprentissage par renforcement	16
Figure 20. Organigramme de la méthode de Contour	17
Figure 21. Organigramme de la méthode de Hough	17
Figure 22. Image originale	18
Figure 23. Conversion en hsv	18
Figure 24. Image input	19
Figure 25. Image grayscale	19
Figure 26. Image filtrée	20
Figure 27. La transformée de hough circulaire	21
Figure 28. Détection	22
Figure 29. Masque rouge	23
Figure 30. Masque bleu	23
Figure 31. Masque rouge et bleu	23
Figure 32. Contours	24
Figure 33. Organigramme de la méthode basée sur l'apprentissage	25
Figure 34. Image input	25
Figure 35. Image redimensionnée (340, 220)	26
Figure 36. Image grayscale	26
Figure 37. Caractéristiques d'Haar	27
Figure 38. Détection	28
Figure 39. Specifications du matériel	32
Figure 40. Exemples des images des ensembles de données de GTSD	33
Figure 41. Détection des panneaux circulaires	34
Figure 42. Détection des panneaux circulaires	35
Figure 43. Fausses détections	35
Figure 44. Fausses non-détections	36

Liste des tableaux

Tableau 1. La précision et le temps de calcul de chaque méthode.....	33
---	----

Liste des abréviations

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

TSR : Traffic Sign Recognition, Reconnaissance de Panneaux Routiers.

SVM : Support Vector Machine, Séparateur à Vaste Marge.

ANN : Artificial neural network, Réseau neuronal artificiel.

RGB : Red Green Blue, Rouge Vert Bleu.

HSV : Hue Saturation Value, Teinte Saturation Valeur.

TH : Transformation de hough.

PSR : Panneaux de signalisation routière.

AdaBoost : Adaptive Boosting.

OpenCV : Open computer vision.

GTSD : German Traffic Sign Detection Benchmark.

Introduction générale



Figure 1. Accident routier

De nos jours, le taux d'accident de circulation produisant des victimes humaines ne cesse de croître. De nombreux gouvernements visent à réduire ce taux et différentes mesures ont été prises comme la réglementation de la circulation, l'interdiction de l'utilisation de dispositifs de distraction comme les téléphones portables.

Ceci dit, la technologie a été utilisée pour sauver des vies humaines. De nombreux appareils ont été construits dans ce but, et les systèmes d'aide à la conduite automobile (en anglais Advanced driver-assistance systems ou ADAS) sont un bon exemple. Ces systèmes ont été conçus comme une aide au conducteur d'un véhicule, dans le but d'éviter les situations dangereuses.

Ainsi, une aide à la conduite automobile est un système de sécurité active d'information ou d'assistance du conducteur conçu dans le but de :

- Éviter l'apparition d'une situation dangereuse risquant d'aboutir à un accident.
- Libérer le conducteur d'un certain nombre de tâches qui pourraient atténuer sa vigilance.
- Assister le conducteur dans sa perception de l'environnement (détecteurs de dépassement, de risque de gel, de piéton, etc.)
- Permettre au véhicule de percevoir le risque et de réagir de manière anticipée par rapport aux réflexes du conducteur.

Un module de reconnaissance des signes de trafic routier (Traffic Sign Recognition TSR en anglais) est un composant très souhaitable dans tout système d'aide à la conduite automobile commercial.

Le but de ce module est de reconnaître les panneaux de signalisation dans un flux vidéo provenant d'une caméra installée dans un véhicule, puis de fournir cette information à d'autres modules ADAS. Un tel type d'information est nécessaire pour de nombreuses tâches,

telles que la connaissance de la vitesse actuelle, guider le conducteur à travers la circulation, et la connaissance générale des zones potentiellement dangereuses.

Les implémentations architecturales communes d'un module de reconnaissance des signes de trafic routier ont divisé ce processus en deux étapes principales : une étape de détection et une autre étape de reconnaissance.

En général, l'étape de détection est effectuée pour identifier la région d'intérêt (ROI) en utilisant la segmentation de couleur, suivie d'une certaine forme d'identification de la forme. Les panneaux de signalisation candidats détectés sont ensuite soit reconnus ou rejetés pendant la phase de reconnaissance.

L'étape de reconnaissance est réalisée sur les panneaux de signalisation candidats, où certains candidats peuvent ne pas contenir de panneaux de signalisation. Cette étape est réalisée avec certaines techniques d'apprentissage automatique telles que le réseau de neurones artificiels (ANN), la machine à vecteurs de support (SVM).

La détection et la reconnaissance des panneaux de signalisation peuvent être confrontée à une ou plusieurs des difficultés suivantes :

- Des signes décolorés,
- La présence d'obstacles dans la scène,
- Des objets similaires dans la scène ou la couleur de fond similaire,
- Des signes routiers endommagés.

Chapitre 1 : Introduction

1.1 Généralités sur le traitement d'images

1.1.1 Introduction

Le traitement d'images est un domaine très vaste qui a connu, et qui connaît encore, un développement important depuis quelques dizaines d'années.

On désigne par traitement d'images numériques l'ensemble des techniques permettant de modifier une image numérique afin d'améliorer ou d'en extraire des informations.

De ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinentes.

1.1.2 Définition d'une image

Une image est plutôt difficile à décrire d'une façon générale. Une image est une représentation du monde. En traitement d'image, la majorité du temps, on considère qu'il s'agit d'une fonction mathématique de $R \times R$ dans R où le couplet d'entrée est considéré comme une position spatiale, le singleton de sortie comme l'intensité (couleur ou niveaux de gris) du phénomène physique.

Il arrive cependant que l'image soit dite "3D" donc la fonction est de $R \times R \times R$ dans R . Les images couleurs peuvent être représentées soit par trois images représentant les trois couleurs fondamentales (on va détailler cela après), soit par une image de $R \times R$ dans $R \times R \times R$.

L'image numérique est l'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter.

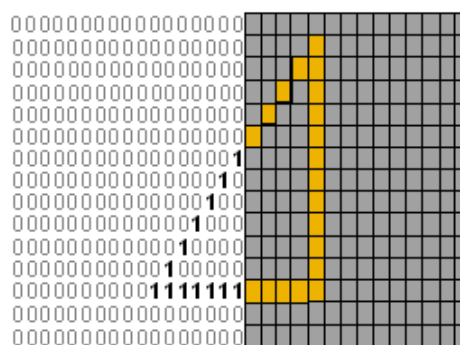


Figure 2. Image numérique

1.1.3 Caractéristiques d'une image numérique

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants :

1.1.3.1 Dimension

C'est la taille de l'image. 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.

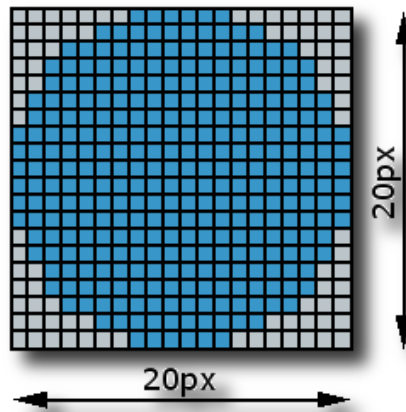


Figure 3. Dimension d'une image

1.1.3.2 Résolution

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur ; plus grand est ce nombre, meilleure est la résolution.

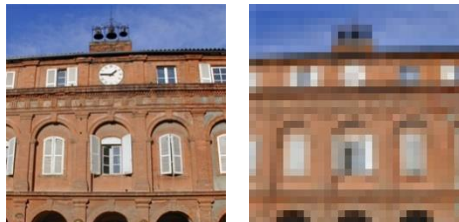


Figure 4. Résolution versus 1/8 résolution

1.1.3.3 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.

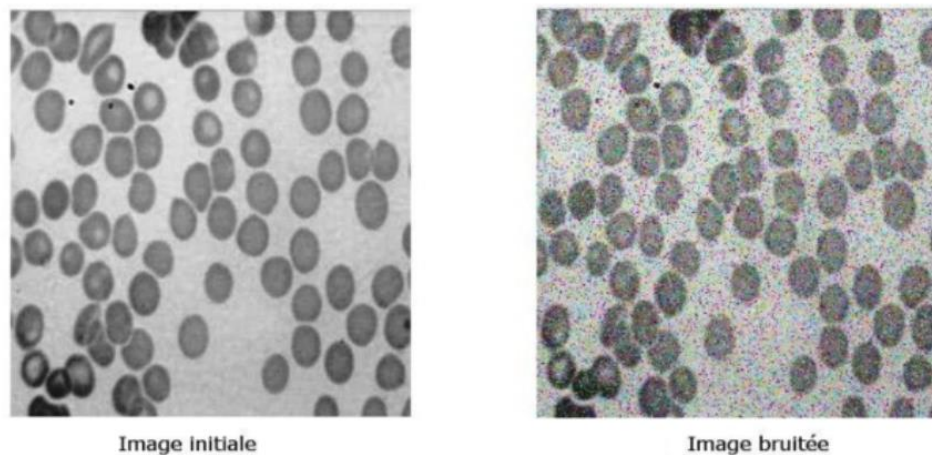


Figure 5. Image avec et sans bruit

1.1.3.4 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 dans l'image. Il permet de donner un grand nombre d'informations sur la distribution des niveaux de gris et de voir entre quelles bornes est répartie la majorité des niveaux de gris dans le cas d'une image trop claire ou d'une image trop foncée. Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci. Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant.

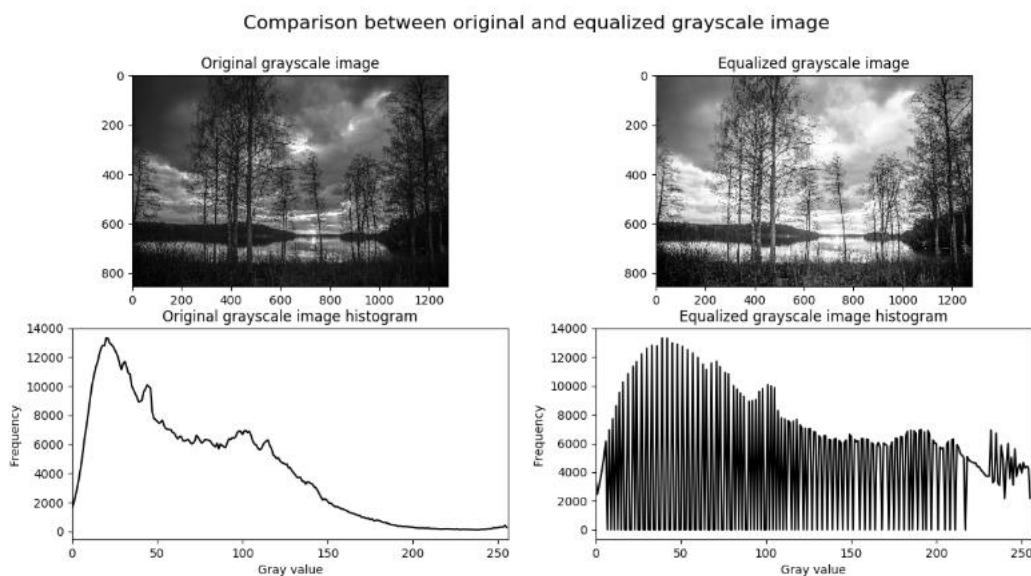


Figure 6. Comparaison de l'histogramme de deux images

1.1.3.5 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.

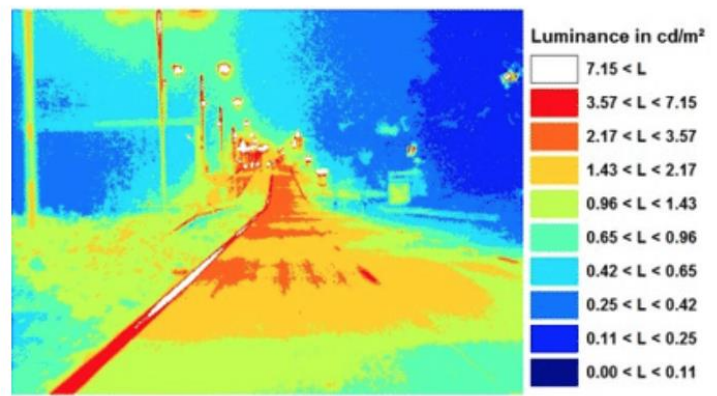


Figure 7. Luminance d'une image

1.1.3.6 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 L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste C est défini par le rapport :

$$C = \frac{L_1 - L_2}{L_1 + L_2}$$

Figure 8. Rapport du contraste

1.1.3.7 Contour

Le contournage d'image est un processus d'identification des contours structuraux des objets dans une image qui à son tour peut nous aider à identifier la forme de l'objet. Prenons un exemple ci-dessous. Sur la gauche, nous avons un rectangle creux et un cercle plein. Lorsque nous appliquons un contour avec la couleur rouge, ce que nous obtenons est l'image à droite. Comme vous pouvez le voir, on a identifié 3 objets, 2 rectangles et un cercle et leurs bordures ont été dessinées en rouge.

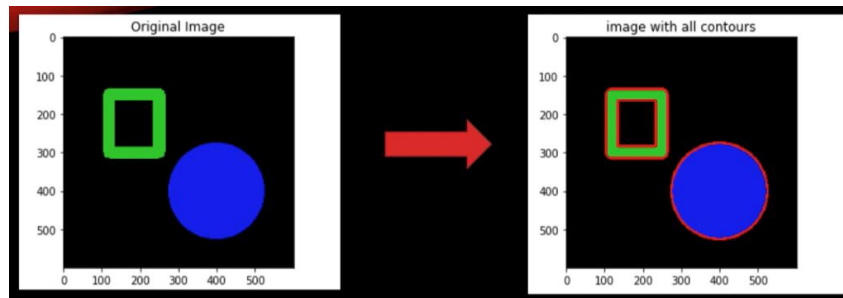


Figure 9. Contour de formes

1.1.4 Les espaces de couleurs

1.1.4.1 Définition

Un espace de couleur ou espace colorimétrique est une représentation numérique des couleurs dans un système colorimétrique ou dans un procédé de synthèse des couleurs. Compte tenu de la trivariance de la vision humaine, ces nombres se présentent généralement sous la forme de triplets. Chaque couleur peut donc être caractérisée par un point dans un espace à trois dimensions. On discutera les plus utilisés dans le domaine de traitement d'images.

1.1.4.2 Images à niveaux de gris (GrayScale)

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise entre 0 et 255. Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Pour cela, il faut que le matériel utilisé pour afficher l'image soit capable de produire les différents niveaux de gris correspondant. Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la 'couleur' de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux.

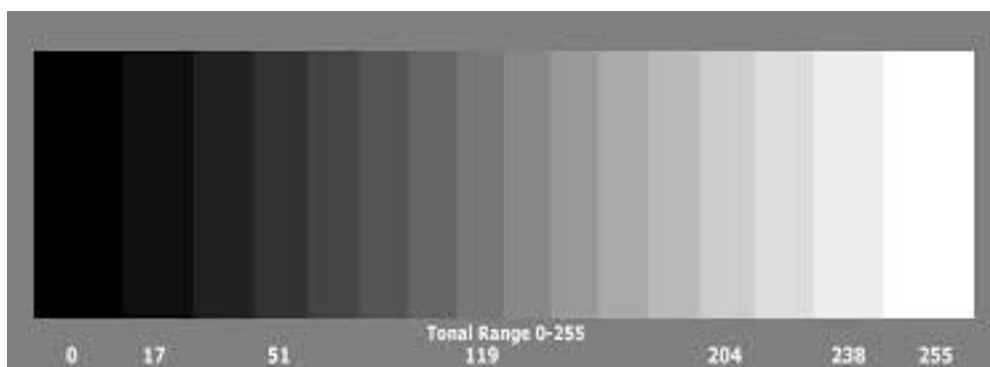


Figure 10. Range du GrayScale

1.1.4.3 Images en couleurs (RGB)

Même s'il est parfois utile de pouvoir représenter des images en noir et blanc, les applications multimédias utilisent le plus souvent des images en couleurs. La représentation

des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation. On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs, ...) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B.)

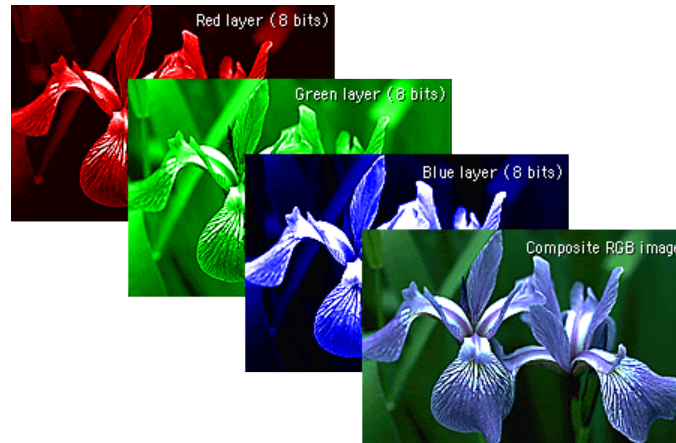


Figure 11. Les 3 couches d'une image RGB

1.1.4.4 TSV Teinte Saturation Valeur (HSV)

TSV est particulièrement utilisé dans les applications graphiques. Dans beaucoup de contextes applicatifs, un utilisateur est amené à choisir une couleur qui sera appliquée à un élément. Utilisée de cette manière, la roue TSV est souvent employée. Ainsi, la `Teinte` (Hue) est représentée par une région circulaire ; un triangle séparé peut être utilisé pour représenter la `Saturation` et la Valeur. Typiquement, l'axe vertical du triangle représente la Saturation, alors que l'axe horizontal représente la `Valeur` (value). De cette manière, une couleur peut être choisie d'abord en sélectionnant la Teinte sur la partie circulaire puis la Saturation et la Valeur sur le triangle.

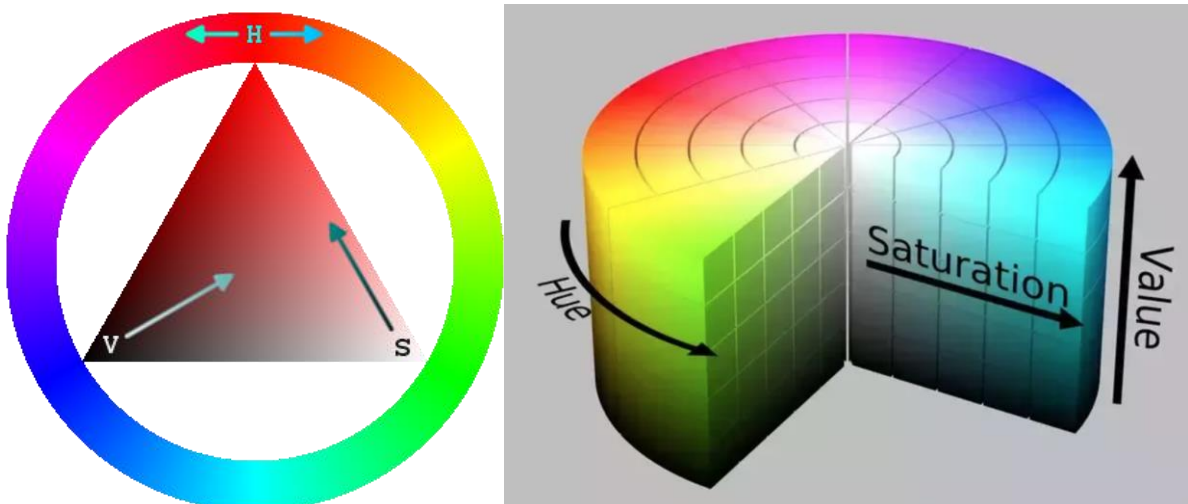
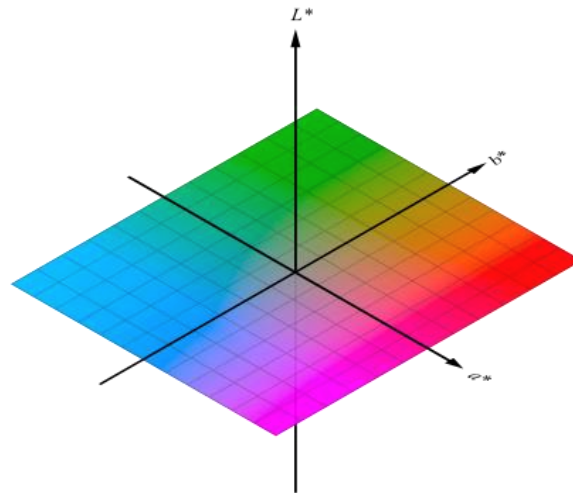


Figure 12. Représentation de la teinte, saturation et valeur

1.1.4.5 LAB (CIELAB)

L'espace chromatique $L^*a^*b^*$ CIE 1976, généralement nommé CIELAB, est un espace de couleur particulièrement utilisé pour la caractérisation des couleurs de surface. Trois grandeurs caractérisent les couleurs : la clarté L^* dérive de la luminance de la surface ; les deux paramètres a^* et b^* expriment l'écart de la couleur par rapport à celle d'une surface grise de même clarté. L'existence d'une surface grise, non colorée, achromatique, implique d'indiquer explicitement la composition de la lumière qui éclaire la surface colorée.



*Figure 13. Représentation de l'espace $L^*a^*b^*$*

1.1.5 Les filtres

1.1.5.1 Filtres linéaires

Un filtre linéaire transforme un ensemble de données d'entrée en un ensemble de données de sortie selon une opération mathématique appelée convolution. Lorsqu'il s'agit de données numérisées comme dans le cas du traitement d'image, la relation entre les valeurs des pixels de sortie et celle des pixels d'entrée est décrite par un tableau de nombres, généralement carré, appelé matrice de convolution. Le temps de calcul est souvent réduit lorsqu'on veut séparer un filtre en deux filtres dont la convolution mutuelle permet de le reconstituer. Cette remarque est utilisée en particulier pour créer un filtre à deux dimensions à partir de deux filtres à une seule dimension (vecteurs) dans le sens horizontal et le sens vertical.

1.1.5.2 Filtres non linéaires

Ils sont conçus pour régler les problèmes des filtres linéaires, sur tout pour ce qui concerne la mauvaise conservation des contours. Leur principe est le même que celui des filtres linéaires, il s'agit toujours de remplacer la valeur de chaque pixel par la valeur d'une fonction calculée dans son voisinage, la seule différence c'est que cette fonction n'est plus linéaire mais une fonction quelconque (elle peut inclure des opérateurs de comparaisons)

Filtre médian :

Sur un voisinage à huit, le nouveau niveau de gris du pixel centre est choisi comme étant la valeur médiane de tous les pixels de la fenêtre d'analyse centrée sur ce dernier. Son avantage est qu'il garde la netteté des éléments qui constituent l'image sans étaler les transitions.

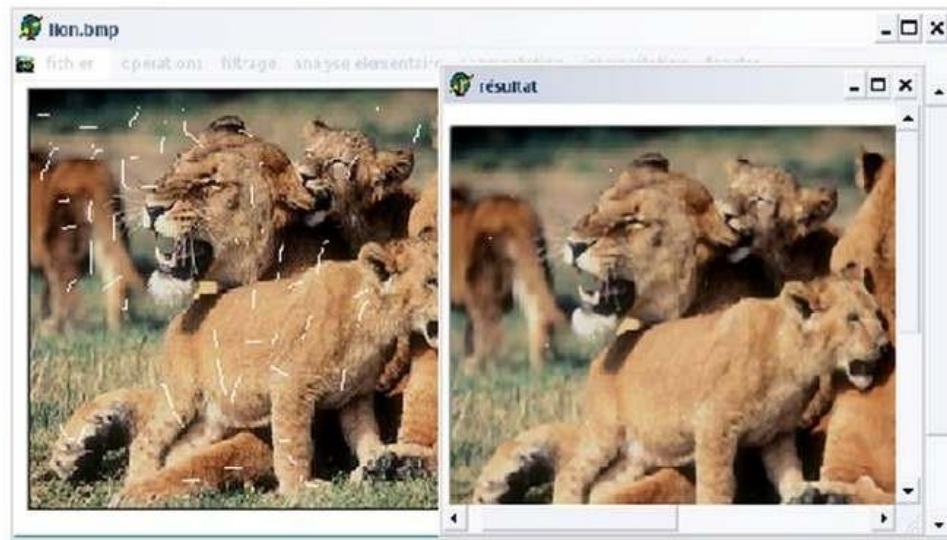


Figure 14. Principe du filtre médian

Filtre maximum :

Même principe mais la valeur choisie est la valeur maximale.

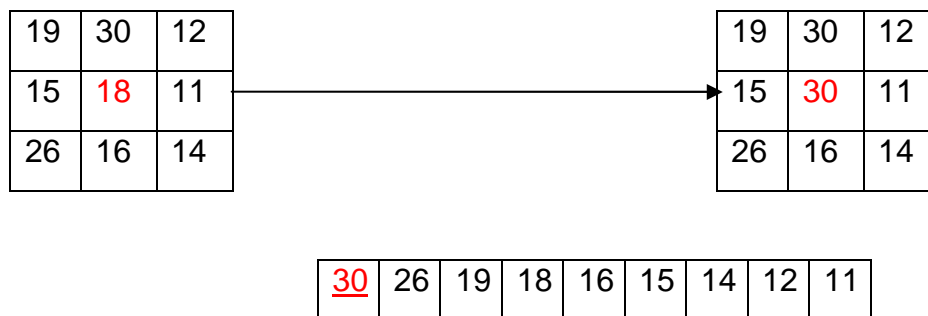


Figure 15. Principe du filtre maximum

Filtre minimum :

Même chose, sauf que la valeur choisie est la valeur minimale.

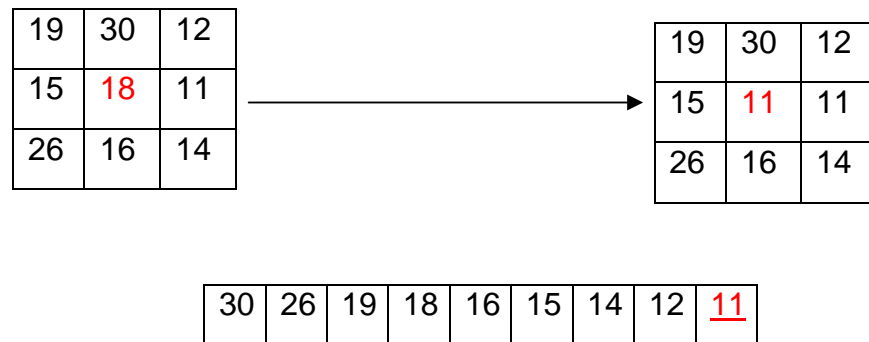


Figure 16. Principe du filtre minimum

Filtre Guassien :

Dans le traitement d'image, un flou gaussien (également connu sous le nom de lissage gaussien) est le résultat du flou d'une image par une fonction gaussienne (du nom du mathématicien et scientifique Carl Friedrich Gauss).

C'est un effet largement utilisé dans les logiciels graphiques, généralement pour réduire le bruit de l'image et réduire les détails. L'effet visuel de cette technique de flou est un flou lisse ressemblant à celui de la visualisation de l'image à travers un écran translucide, distinctement différent de l'effet bokeh produit par une lentille floue ou l'ombre d'un objet sous un éclairage habituel.

Le lissage gaussien est également utilisé comme étape de prétraitement dans les algorithmes de vision par ordinateur afin d'améliorer les structures d'image à différentes échelles.



Figure 17. La différence entre un petit et un grand flou gaussien

1.1.5.1 Lissage

Ceux-ci sont des filtres passe-bas qui coupent plus ou moins les plus hautes fréquences. Ils sont utilisés pour atténuer les bruits d'origines les plus diverses qui polluent l'information, en particulier dans la détection de contours considérée ci-après. Techniquement, il s'agit de traductions discrètes de filtres continus qui, comme ceux-ci, ne modifient pas le niveau global du signal. Les termes de la matrice de convolution sont donc généralement des entiers à diviser par leur somme.

1.2 Généralités sur Machine Learning

Le machine learning est un domaine passionnant. Issu de nombreuses disciplines telles que la statistique, l'optimisation, les algorithmes, le traitement du signal, c'est un domaine de recherche en constante évolution qui prend désormais pied dans notre société. Il a été utilisé dans la reconnaissance automatique de caractères ou les filtres anti-spam pendant des décennies, et est maintenant utilisé pour empêcher la fraude bancaire, recommander des livres, des films ou d'autres produits qui correspondent à nos goûts, reconnaître des visages dans le viseur de notre appareil photo ou traduire automatiquement une langue de texte d'une personne à une autre. Dans les prochaines années, l'apprentissage automatique pourrait nous permettre d'améliorer la sécurité routière (notamment grâce aux voitures autonomes), les interventions d'urgence en cas de catastrophes naturelles, le développement de nouveaux médicaments ou l'efficacité énergétique de nos bâtiments et industries.

1.2.1 Qu'est-ce que le machine learning ?

Qu'est-ce qu'apprendre ? comment apprend-on ? et qu'est-ce que cela signifie pour les machines ? Les problèmes d'apprentissage attirent des experts en informatique et en mathématiques ainsi que des neurologues, des éducateurs, des philosophes et des artistes. Fabien Benureau (2015) a proposé une définition des robots, des animaux de compagnie ou des humains qui conviennent aux programmes informatiques : « L'apprentissage est le changement de comportement basé sur l'expérience. Dans le cas du programme informatique qui nous intéresse dans ce livre, nous disons machine learning ou machine learning, lorsque le programme a la capacité d'apprendre sans programmer explicitement cette modification. Cette définition a été donnée par Arthur Samuel (1959). Par conséquent, on peut opposer un programme traditionnel qui utilise un processus et les données qu'il reçoit en entrée pour produire une réponse en sortie, tandis qu'un programme d'apprentissage automatique utilise des données et une réponse pour générer l'autorisation du premier.

Exemple :

Supposons qu'une entreprise souhaite connaître le montant total que les clients dépensent à partir de leurs factures. Il suffit d'appliquer l'algorithme classique, c'est-à-dire une simple addition : aucun algorithme d'apprentissage n'est requis. Supposons maintenant que nous souhaitions utiliser ces factures pour déterminer les produits qu'un client est le plus susceptible d'acheter dans un mois. Bien que cela puisse être pertinent, nous n'avons évidemment pas toutes les informations nécessaires pour le faire. Cependant, si nous avons un grand nombre d'historiques d'achats personnels, nous pouvons utiliser des algorithmes d'apprentissage automatique pour dériver des modèles prédictifs afin de fournir des réponses à nos questions.

1.2.2 Pourquoi utiliser le machine learning ?

L'apprentissage automatique peut être utilisé pour résoudre des problèmes -Nous ne savons pas comment le résoudre (comme dans l'exemple d'achat prévisionnel ci-dessus). Nous savons comment le résoudre, mais nous ne savons pas formaliser comment nous le résolvons en termes algorithmiques (par exemple, la reconnaissance d'images ou la compréhension du langage naturel est le cas).

On sait résoudre ce problème, mais le programme est trop gourmand en ressources de calcul (par exemple, pour prédire l'interaction entre macromolécules, la simulation est très lourde).

Par conséquent, lorsque les données sont (relativement) abondantes, mais que les connaissances ne sont pas facilement accessibles ou sous-développées, utilisez l'apprentissage automatique. Par conséquent, l'apprentissage automatique peut également aider les humains à apprendre : les modèles créés par des algorithmes d'apprentissage peuvent révéler l'importance relative de certaines informations, ou comment elles interagissent pour résoudre des problèmes spécifiques. Dans l'exemple de prédiction d'achat, la compréhension du modèle nous permet d'analyser quelles caractéristiques des achats passés peuvent prédire les achats futurs. Cet aspect de l'apprentissage automatique est largement utilisé dans la recherche scientifique

Ingrédients du machine learning :

L'apprentissage automatique repose sur deux piliers de base :

- D'une part, les données, qui sont un exemple à partir duquel l'algorithme va apprendre ;

- D'autre part, l'algorithme d'apprentissage, qui est le processus que nous exécutons sur ces données pour générer un modèle. L'exécution d'un algorithme d'apprentissage sur un ensemble de données est appelée formation

Ces deux piliers sont d'égale importance. D'une part, aucun algorithme d'apprentissage ne peut créer un bon modèle à partir de données non pertinentes. Le concept d'entrée et de sortie d'ordures stipule l'algorithme d'apprentissage auquel nous fournissons des données de mauvaise qualité Il ne pourra rien faire avec, sauf des prédictions grossières. En revanche, la qualité du modèle appris à l'aide d'algorithmes inappropriés sur des données connexes ne sera pas très bonne. Ce livre est dédié au deuxième de ces algorithmes d'apprentissage piliers. Cependant, une partie importante du travail d'un apprenant machine ou d'un scientifique des données ne doit pas être négligée en tant que travail d'ingénierie, notamment la préparation des données pour éliminer les données anormales, la gestion des données manquantes et la sélection des données pertinentes. Représentants et ainsi de suite.

Attention :

Bien qu'il soit d'usage d'appeler les deux du même nom, l'algorithme d'apprentissage automatique doit être distingué du modèle d'apprentissage : le premier utilise des données pour générer le second, qui peut ensuite être appliqué comme un programme ordinaire.

Par conséquent, les algorithmes d'apprentissage peuvent modéliser des phénomènes à partir d'exemples. Nous pensons ici que pour ce faire, nous devons définir et optimiser des objectifs. Par exemple, cela peut être pour minimiser le nombre d'erreurs que le modèle fait sur les exemples d'apprentissage.

Les algorithmes sont les moteurs du machine learning.

Tout comme un enfant apprend à identifier des fruits en les mémorisant dans un livre d'images, dans l'apprentissage supervisé l'algorithme est formé par un ensemble de données déjà étiquetées et ayant une sortie prédéfinie



Par conséquent, l'apprentissage automatique est basé sur les mathématiques, en particulier les statistiques, pour construire des modèles et déduire des données d'une part, et l'informatique d'autre part pour la représentation des données et la mise en œuvre efficace des algorithmes d'optimisation. La quantité croissante de données disponibles nécessite l'utilisation d'architectures informatiques et de bases de données distribuées. C'est un point important, mais nous ne le traiterons pas dans ce livre.

1.2.3 Types de problèmes de machine learning :

Le machine learning est un champ assez vaste, et nous dressons dans cette section une liste des plus grandes classes de problèmes auxquels il s'intéresse (**figure 18**)

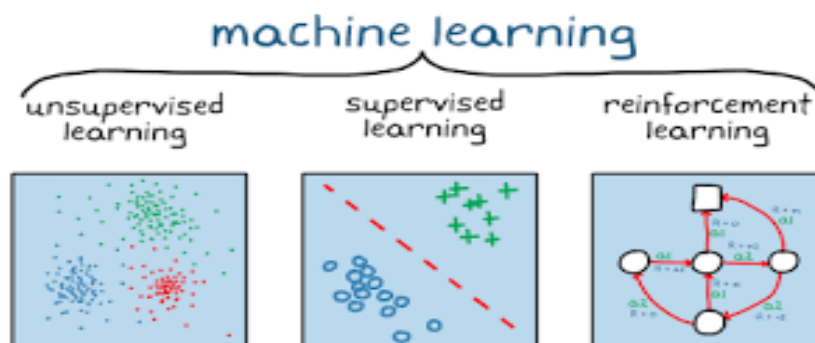


Figure 18. Les types de problèmes de machine learning

- Apprentissage supervisé :

L'apprentissage supervisé est probablement le type de problème d'apprentissage automatique le plus facile à appréhender : son but est d'apprendre à faire des prédictions à

partir d'une liste d'exemples étiquetés, c'est-à-dire avec la valeur à prédire. Le label agit comme un « enseignant » et supervise l'apprentissage de l'algorithme.

L'apprentissage supervisé est une forme d'apprentissage automatique qui crée des modèles d'intelligence artificielle basés sur des données d'apprentissage, ce qui signifie que les humains sont qualifiés (par exemple, en répertoriant les objets qui existent sur des milliers d'images dans l'ensemble d'apprentissage pour demander à l'algorithme d'établir des images de corrélation La relation entre la caractéristique et l'objet identifié qu'elle contient).

Techniquement, l'apprentissage supervisé induit une fonction de données d'apprentissage étiquetées composées d'un ensemble d'exemples d'apprentissage. Dans l'apprentissage supervisé, chaque exemple est une paire composée d'un objet d'entrée (généralement un vecteur) et d'une valeur de sortie attendue (également appelée signal supervisé). L'algorithme d'apprentissage supervisé analyse les données d'apprentissage et génère une fonction d'inférence qui peut être utilisée pour mapper de nouveaux exemples. La meilleure solution permettra à l'algorithme de déterminer correctement la classe l des instances invisibles. Cela nécessite des algorithmes d'apprentissage pour généraliser des données d'entraînement à des situations invisibles de manière "raisonnable".

- *Classification binaire :*

Dans le cas où les étiquettes sont binaires, elles indiquent l'appartenance à une classe. On parle alors de classification binaire.

Classification binaire est un problème d'apprentissage supervisé dans lequel l'espace des étiquettes est binaire, autrement dit $Y = \{0, 1\}$ est appelé un problème de classification binaire. Par exemple Identifier si une image contient ou non un panneau d'une forme cercle.

- **Apprentissage non supervisé :**

Dans l'apprentissage non supervisé, les données ne sont pas étiquetées. Modélisez ensuite les observations pour mieux comprendre leurs problèmes. L'apprentissage non supervisé fait référence à la situation dans laquelle les données ne sont pas étiquetées dans l'apprentissage automatique. Il s'agit donc de découvrir la structure derrière ces données non étiquetées. Étant donné que les données ne sont pas étiquetées, il est impossible pour l'algorithme de calculer la note de passage avec certitude. L'absence d'étiquettes ou d'annotations est une caractéristique des tâches d'apprentissage non supervisées, elles se distinguent donc des tâches d'apprentissage supervisées.

L'introduction de méthodes d'apprentissage non supervisées dans le système est un moyen de faire l'expérience de l'intelligence artificielle. De manière générale, les systèmes d'apprentissage non supervisé permettent d'effectuer des tâches plus complexes que les systèmes d'apprentissage supervisé, mais ils peuvent également être plus difficiles à prévoir. Par exemple, même si un système d'IA d'apprentissage non supervisé peut trier les chats et les chiens par lui-même, il peut également ajouter des catégories inattendues et indésirables et classer les races inhabituelles, introduisant ainsi plus de bruit d'ordre.

Il existe deux principales méthodes d'apprentissage non-supervisées :

- i. Les méthodes par partitionnement telles que les algorithmes des k-moyennes ou k-médoïdes.
- ii. Les méthodes de regroupement hiérarchique.

- **Apprentissage par renforcement :**

Dans l'apprentissage par renforcement, le système d'apprentissage peut interagir avec son environnement et effectuer des opérations. En échange de ces actions, il sera récompensé. Si l'action est un bon choix, la récompense peut être positive, sinon, elle peut être négative. Les récompenses apparaissent parfois après une série d'actions de longue durée ; c'est le cas par exemple pour les systèmes qui apprennent à jouer au Go ou aux Echecs. Par conséquent, dans ce cas, l'apprentissage comprend la définition de stratégies, c'est-à-dire des stratégies pour obtenir systématiquement les meilleures récompenses. Les principales applications de l'apprentissage par renforcement sont les jeux (échecs, go, etc.) et la robotique.

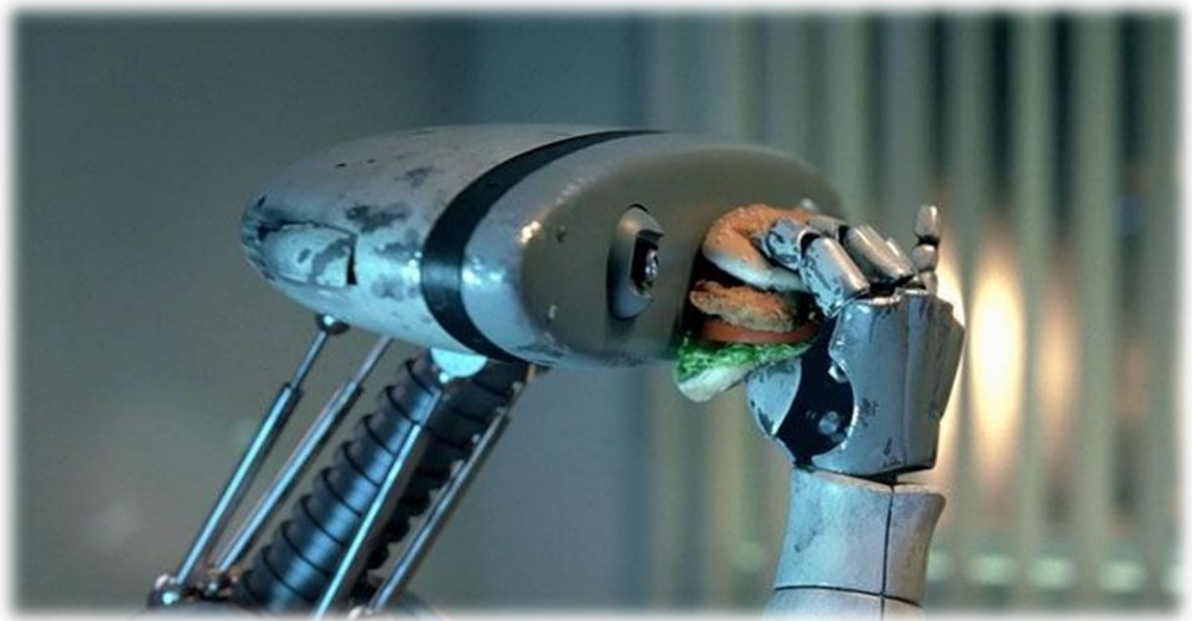


Figure 19. Apprentissage par renforcement

1.3 Objectif du projet

Vu l'intérêt d'une localisation précise des panneaux, notre projet de fin d'étude a pour objectif de développer un code capable de détecter des panneaux de signalisation sur des images. Nous nous sommes intéressés plus particulièrement à la détection de panneaux routiers de forme circulaires, en utilisant des méthodes de traitement d'image et de machine learning dans le but de minimiser les taux d'accident routier.

Chapitre 2 : Les méthodes proposées

2.1 Méthodes basée sur le traitement (Forme)

Lors de la réalisation de la reconnaissance nous faisons face à plusieurs difficultés parmi ces difficultés on cite :

- Les conditions d'éclairage
- Le bruit de capteur d'image
- La ressemblance de couleur

Il faut surmonter ces obstacles afin de réaliser une bonne reconnaissance d'objet.

Notre travail est basé sur des différentes étapes représentées par la Figure 20 et 21 :

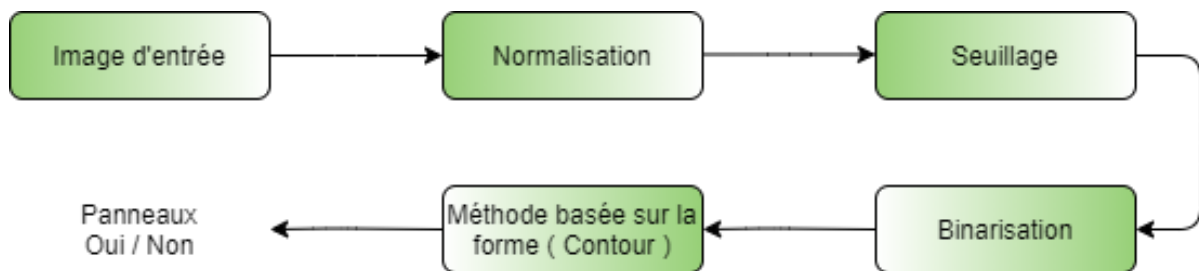


Figure 20. Organigramme de la méthode de Contour

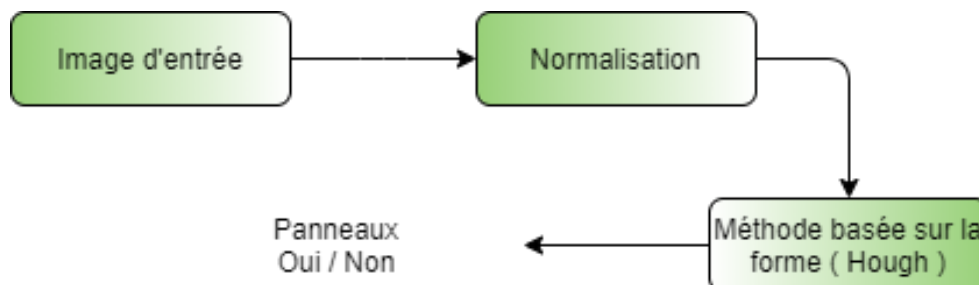


Figure 21. Organigramme de la méthode de Hough

2.1.1 Lecture de l'image

La lecture d'image se fait par appel à la fonction `imread()` qui prend en paramètre le chemin de l'image et le `'flag'` ou sa représentation en entier ; 1 pour `IMREAD_COLOR`, 0 pour `IMREAD_GRAYSCALE` et -1 pour `IMREAD_UNCHANGED`.

2.1.2 Normalisation

Selon la méthode, l'image d'input a besoin de différent changement pour la normaliser, pour nos méthodes les changements suivants seront nécessaires :

Conversion de BGR2HSV

La conversion en HSV se fait en appelant la fonction `cvtColor` qui prend 2 paramètres, la variable image et le type de conversion, pour notre cas c'est la conversion BGR2HSV.



Figure 22. Image originale



Figure 23. Conversion en hsv

Conversion de BGR2GRAY

Même chose que pour la conversion précédente, sauf qu'on change l'attribut BGR2HSV par BGR2GRAY



Figure 24. Image input



Figure 25. Image grayscale

Filtre gaussien

En appliquant le filtre gaussien sur l'image grayscale pour filtrer le bruit non-souhaitable.



Figure 26. Image filtrée

2.1.3 Forme

2.1.3.1 la transformée de hough circulaire

La transformée de Hough (TH) a une histoire de plus d'un demi-siècle et est l'un des premiers algorithmes dans le domaine de la vision par ordinateur. C'est un cadre général pour détecter des formes dans les images, et il convient également pour analyser des formes (lignes, cercles, etc.) et n'importe quoi (objets). Le TH a fait l'objet de nombreux travaux, portant sur la nature des formes, leur paramétrage, leur quantification, ou encore l'accélération d'algorithmes. Cependant, presque tous les algorithmes existants sont peu appliqués aux images de contours ou de points d'intérêt. Cependant, estimer la dérivée spatiale directement à partir du niveau de gris est suffisant pour appliquer TH de manière dense. De plus, pour les formes d'analyse telles que les lignes ou les cercles, cette estimation permet une correspondance unique de l'espace image à l'espace des paramètres (vote un à un), beaucoup plus rapide que la projection traditionnelle (votes un à plusieurs ou plusieurs à un).

La transformée de Hough (TH) peut également être utilisée pour trouver des cercles dans une image. La première description de l'utilisation de TH pour détecter des cercles dans des images réelles est Kimme, Ballard et Sklansky (Kimme, Ballard et Sklansky 1975).

Comme la droite, le cercle de rayon r et centre (a, b) , peut être décrit par une équation cartésienne :

$$(x - a)^2 + (y - b)^2 = r^2$$

L'ensemble des points (x, y) se situe sur le périmètre du cercle de centre (a, b) et de rayon r . Il s'agit donc cette fois de déterminer les paramètres d'un cercle à l'aide d'une transposition dans un espace à trois dimensions. Pour un point (x, y) quelconque dans l'image, alors l'Équation permet de calculer le triplet de paramètres (a, b, r) pour décrire ce cercle. Par rapport au cas de la droite, il y a donc un paramètre supplémentaire r à déterminer.

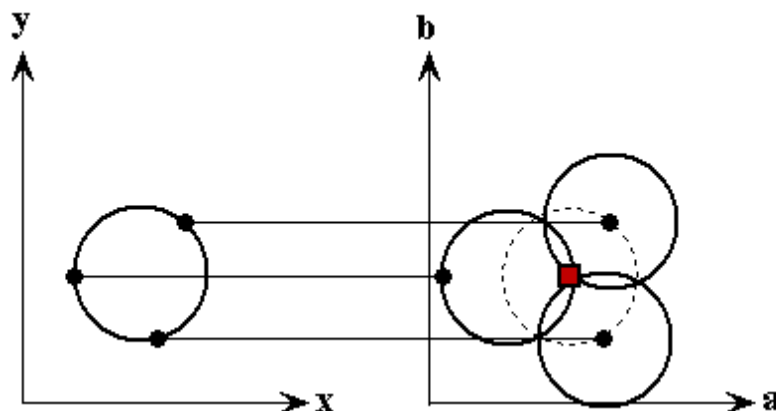


Figure 27. La transformée de hough circulaire

Pour appliquer la transformée de hough en a besoin de la phase de normalisation et plus précisément la conversion en grayscale et le filtre `gaussian blur`.

N.B Attention cette méthode ne dépend pas de la phase de seuillage

En appliquant la méthode sur l'image sur laquelle nous avons appliqué la normalisation. (**Figure 26**)



Figure 28. Détection

Conclusion :

- La transformée de Hough est utile pour la détection de contours descriptibles par peu de paramètres.
- Elle est robuste (bruit, occlusions).
- On peut envisager des algorithmes plus efficaces.

2.1.3.2 Contour

Cette méthode consiste à appliquer le suivi de contour aux masques binarisés produit par la phase de thresholding.

Seuillage et extraction de couleur

Maintenant c'est au tour du seuillage des valeurs hsv, cela se fait en créant des intervalles de valeur hsv représentant les couleurs souhaitant extraire.

Binarisation et création des masques

Par l'appel à la fonction `inRange` (prend en paramètre la variable image et les deux extrémités de nos intervalles) un masque est créé. Dans notre cas, on dispose de deux masques, un masque pour la couleur rouge et un autre pour la couleur bleue, il suffit d'additionner ces deux sous-masques pour créer un masque final. Le résultat est une image où chaque bit se trouvant dans l'intervalle défini est mis à 1 et vice versa.

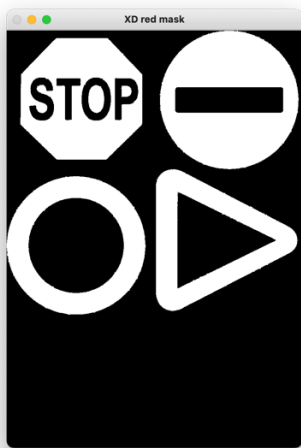


Figure 29. Masque rouge

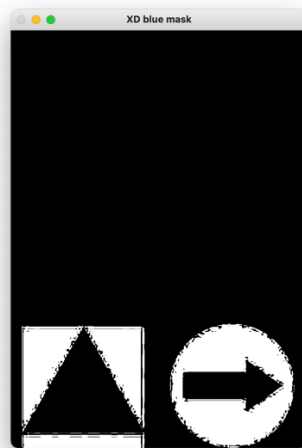


Figure 30. Masque bleu

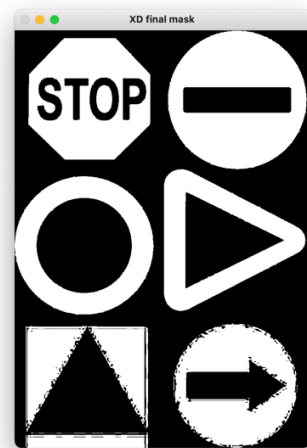


Figure 31. Masque rouge et bleu

Suivi de contour

Nous trouvons des contours dans notre image binaire, gérons la saisie de la valeur de tuple correcte de `cv2.findContours` en fonction de notre version OpenCV. Nous commençons à boucler sur chacun des contours individuels. Pour chacun d'eux, nous calculons le centre du contour, puis effectuons une détection de forme et un étiquetage. Enfin, nous dessinons les contours sur notre image puis nous affichons nos résultats.



Figure 32. Contours

Conclusion

Cette méthode est peu performante et requière beaucoup de paramétrage, ce qui nous a pousser à approfondir nos recherches.

2.2 Méthodes basée sur l'apprentissage

La grande variabilité des panneaux de signalisation (PSR) dans des environnements non contrôlés rend la détection et la classification un défi !

La reconnaissance PSR aide à empêcher les conducteurs de prendre des décisions inappropriées dans des situations potentiellement dangereuses.

La méthode d'identification PSR est utilisée en combinaison avec d'autres méthodes pour collecter des informations et mesurer la position et la direction de divers repères mobiles sur la plate-forme terrestre par rapport au cap.

Dans la littérature, on peut trouver deux principales approches :

1. Reconnaissance basée sur la couleur
2. Reconnaissance basée sur les niveaux de gris

La première s'appuie sur la couleur pour réduire les faux positifs dans le processus de reconnaissance, tandis que la méthode en niveaux de gris se concentre sur la forme géométrique de l'objet. Notre méthode est basée sur la seconde méthode.

La méthode est une approche complexe basée sur une cascade de détecteurs, où chacun est un ensemble de classificateurs dopés « Boosted » sur la base des caractéristiques de Haar.

Notre travail est basé sur des différentes étapes représentées par la Figure 33 :

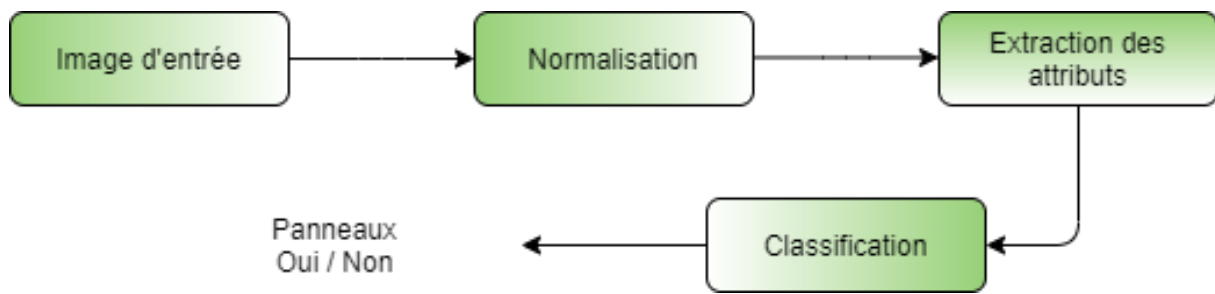


Figure 33. Organigramme de la méthode basée sur l'apprentissage

2.2.1 Normalisation

Une fois un signal localisé par le détecteur basé sur des filtres de Haar, il est normalisé :

- L'image est redimensionnée afin d'être compatible avec le classeur.
- L'image résultante est convertie en niveaux de gris, la luminosité est normalisée par une égalisation de l'histogramme.

2.2.1.1 Resizing

La redimensionne d'une image se fait en appelant la fonction `resize` qui prend 2 paramètres, la variable image et la nouvelle dimension (x, y).



Figure 34. Image input



Figure 35. Image redimensionnée (340, 220)

2.2.1.2 Grayscale

L'échelle de gris réduit simplement la complexité : d'une valeur de pixel 3D (R,V,B) à une valeur 1D. De nombreuses tâches ne s'en sortent pas mieux avec les pixels 3D



Figure 36. Image grayscale

2.2.2 Extraction des attributs

La deuxième étape consiste à extraire les caractéristiques définies par Viola et Jones dans leur article.

Les cascades de Haar crée les fonctionnalités qu'il utilise en appliquant un filtre carré 24x24 à l'image. Ce sont les filtres montrés dans l'image ci-dessous.

La caractéristique est obtenue en soustrayant la somme des pixels de l'image couverte par la zone blanche du filtre de la somme des pixels couverts par la zone bleue. Lorsque le filtre est appliqué à l'ensemble de l'image, ce processus génère de nombreuses fonctionnalités. Viola et Jones ont cité plus de 180 000 fonctionnalités dans leur article pour obtenir une image de 384 x 288.

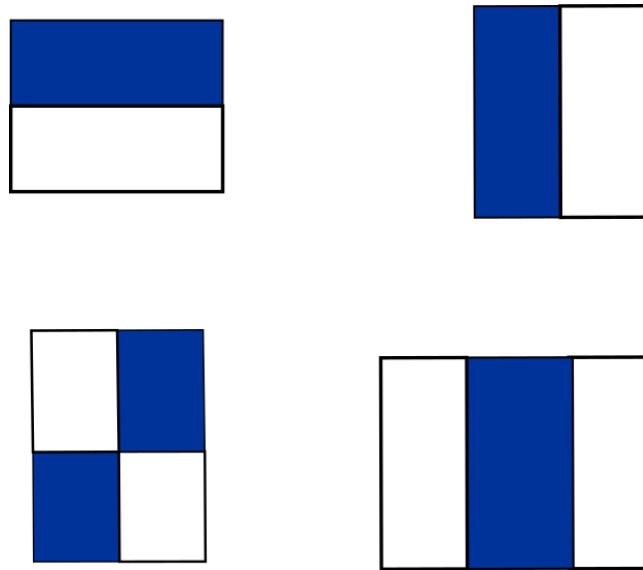


Figure 37. Caractéristiques d'Haar

Il faut choisir la plus pertinente parmi celles-ci. Grâce à l'algorithme AdaBoost, nous verrons qu'il possède des capacités de généralisation relativement importantes et offre un temps d'exécution inégalé pour des applications telles que la détection d'objets.

2.2.3 Classification

Boosting

Est une technique d'apprentissage puissant qui permet de combiner la performance de nombreuses fonctions simples de classification (ou faibles) pour produire un classificateur fort

AdaBoost

AdaBoost est un méta-algorithme. En termes d'apprentissage, il a construit une combinaison d'algorithmes d'apprentissage de base pour améliorer leur efficacité. L'algorithme adaptatif AdaBoost est amplifié car chaque classificateur est basé sur les objets suivants, qui ont été mal classés à l'aide des classificateurs précédents

Cette méthode est basée sur un algorithme adaptatif boosting (gain adaptatif) ou AdaBoost abrégée. Le sens de l'algorithme est que, si nous disposons d'un ensemble d'objets de référence, c'est-à-dire d'avoir les valeurs et la classe à laquelle ils appartiennent. Par exemple : (-1 \rightarrow pas de cercle +1 \rightarrow un cercle)

L'algorithme AdaBoost appelle un classificateur faible pour chaque cycle. Après chaque appel, les poids de distribution sont mis à jour pour répondre à l'importance de chaque objet en tant qu'ensemble d'apprentissage de classification. Ensuite, à chaque itération, le poids des objets mal classés augmente, de sorte que le nouveau classificateur "se concentre" sur ses objets.

Trois idées fondamentales sont à la base des méthodes de boosting :

- L'utilisation d'un comité d'experts spécialistes que l'on fait voter pour atteindre une décision.
- La pondération adaptative des votes par une technique de mise à jour.
- La modification de la distribution des exemples disponibles pour entraîner chaque expert, en surpondérant au fur et à mesure les exemples mal classés aux étapes précédentes.

L'idée est de refuser au plus tôt une image dès les premiers niveaux plutôt que de la faire descendre dans le classifieur. Le classifieur en cascade est généralement stocké dans un fichier xml.

En appliquant la méthode sur l'image sur laquelle nous avons appliqué la normalisation. (**Figure 36**)

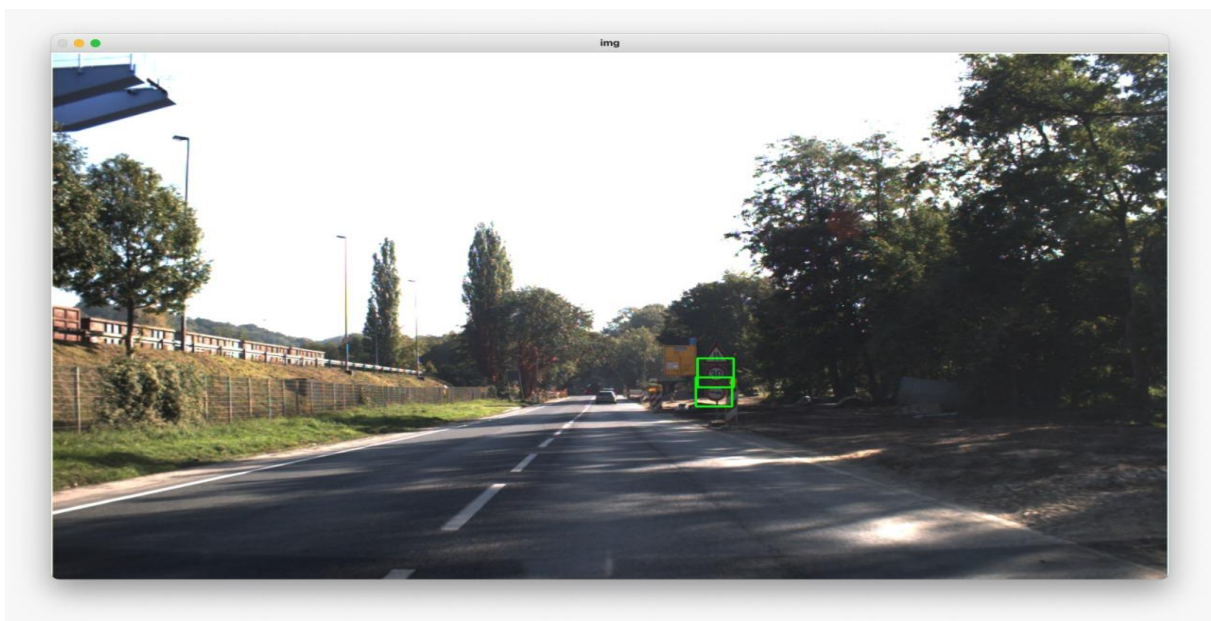


Figure 38. Détection

Chapitre 3 : Résultats

3.1 Hardware and Software

3.1.1 Environnement immatériel ou ‘Software’

3.1.1.1 Python

Langage de programmation utilisé en machine learning et en data science, le langage Python s'impose également dans d'autres secteurs d'activité grâce à sa simplicité et sa compatibilité. Mais de quoi parle-t-on exactement ?

Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ;

Python est un langage de programmation qui peut s'utiliser dans de nombreux contextes et s'adapter à tout type d'utilisation grâce à des bibliothèques spécialisées. Il est cependant particulièrement utilisé comme langage de script pour automatiser des tâches simples mais fastidieuses, comme un script qui récupérerait la météo sur Internet ou qui s'intégrerait dans un logiciel de conception assistée par ordinateur afin d'automatiser certains enchaînements d'actions répétitives.

On l'utilise également comme langage de développement de prototype lorsqu'on a besoin d'une application fonctionnelle avant de l'optimiser avec un langage de plus bas niveau. Il est particulièrement répandu dans le monde scientifique, et possède de nombreuses bibliothèques optimisées destinées au calcul numérique.



3.1.1.2 La librairie OpenCV

OpenCV est une puissante librairie open source offrant aux développeurs une série d'algorithmes graphiques grâce auxquels ils peuvent intégrer des fonctionnalités avancées à leurs projets sans avoir à tout programmer manuellement.

Elle supporte un grand nombre de langages de programmation dont C++, C, Python et Java. L'outil intègre également plusieurs modules complémentaires et répondant à des besoins spécifiques. Il est ainsi possible de profiter de structures et de fonctions de données, ou encore de bénéficier d'outils de traitement d'images approfondis.

En résumé elle est :

- Une librairie open source de traitement et analyse d'images et vidéos avec des Interfaces pour les principaux langages de programmation.
- Optimisée pour les applications temps réelles.
- Fournit une API bas et haut niveau.
- Utilisé aussi bien dans les laboratoires de recherche que dans l'industrie.

Fonctions :

- Manipulation d'images (chargement, sauvegarde, copie, conversion...).
- Manipulation et acquisition de vidéos.
- Manipulations de matrices et algèbre linéaire.
- Structure de données utilitaires variées (listes, files, ensembles, graphes...).
- Traitement d'images (filtrage, détections de discontinuités, morphologie mathématique...)
- Analyse d'images (composantes connexes, ajustement de primitives, transformée de distance...).
- Vision (calibration de caméra, stéréovision, recherche d'association...)
- Reconnaissance de forme.
- Interface graphique (affichage d'images, de vidéos, gestion des évènements...) .

Algorithmes d'apprentissages :

Certains algorithmes classiques dans le domaine de l'apprentissage artificiel sont aussi disponibles :

- K-means.
- AdaBoost et divers algorithmes de boosting.
- Réseau de neurones artificiels.
- Séparateur à vaste marge.
- Estimateur (statistique).

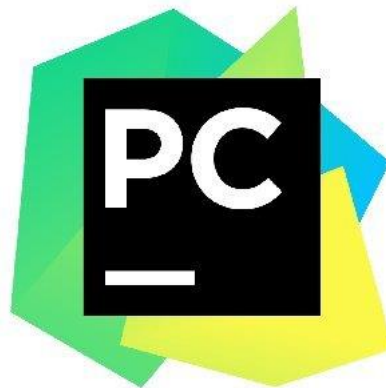
- Les arbres de décision et les forêts aléatoires.



3.1.1.3 PyCharm

PyCharm est l'IDE Python de JetBrains, conçu pour les développeurs Python professionnels. La complétion de code de pointe, la navigation de code, la refactorisation sûre et le débogage intelligent ne sont que quelques fonctionnalités importantes qui contribuent à faire du développement de logiciels professionnels une expérience plus productive et plus agréable.

PyCharm est livré avec une large prise en charge des frameworks Web Python, du développement JavaScript moderne, ainsi que des outils de base de données avancés et des intégrations d'outils scientifiques.



3.1.2 Environnement matériel ou 'Hardware'

Pour la réalisation de notre projet nous avons utilisé le matériel avec les caractéristiques suivantes :

- Model : MacBook Pro 15 Pouces (Mi 2015)
- Mémoire RAM : 16 GB 1600 MHz DDR3
- Processeur : 2,2 GHz Quad-cœur Intel Core i7
- System d'exploitation : MacOS Big Sur Version 11.0.1

macOS Big Sur

Version 11.0.1

MacBook Pro (Retina, 15-inch, Mid 2015)

Processor 2,2 GHz Quad-Core Intel Core i7

Memory 16 GB 1600 MHz DDR3

Startup Disk Mac HD

Graphics Intel Iris Pro 1536 MB

Figure 39. Specifications du matériel

3.2 Base de données

Pour évaluer la performance de notre système de détection, nous avons utilisé la base de données GTSD (German Traffic Sign Detection Benchmark).

Cette base contient un ensemble de données réaliste complète d'images de plus de 50.000 signes de la circulation. Il reflète les fortes variations de l'aspect visuel des signes en raison de la distance, l'éclairage, les conditions météorologiques, les occlusions partielles, et les rotations. Les images sont complétées par plusieurs ensembles de fonctionnalités pré calculées pour permettre l'application d'algorithmes d'apprentissage machine sans connaissances de base dans le traitement de l'image.

L'ensemble de données comprend 43 classes avec des fréquences de classe asymétriques. Un exemple d'images de base de données GTSD est illustré dans la figure 40



Figure 40. Exemples des images des ensembles de données de GTSD

3.3 Comparaisons et résultats

Nous avons testé les méthodes citées précédemment sur notre base de données d'images et les résultats sont les suivants :

	Méthode de Hough	Méthode de Contour	Méthode M.L.
Accuracy	27.20%	12.67%	75.59%
Temps de calcul	0.03388s	0.04600s	0.10007s

Tableau 1. La précision et le temps de calcul de chaque méthode

On peut clairement déduire que la méthode sur l'apprentissage gagne haut la main, cela dit celle-ci n'est pas parfaite, nous allons détailler cela ci-dessous.

La basse performance donnée par les méthodes basées sur le traitement d'image peut être expliquée par les contraintes suivantes :

- Bruits dans la production des masques
- Imperfections des intervalles choisis pour l'extraction des couleurs
- Méthodes nécessitant beaucoup de paramétrage et de preprocessing

Les résultats insatisfaisants des deux premières méthodes nous ont forcé à approfondir nos recherches et éventuellement proposer la méthode d'apprentissage, même si cette dernière n'est pas parfaite elle présente des résultats plus ou moins satisfaisants par rapport aux contraintes de temps que nous en avons fait face. Ci-dessous nous des figures montrant des vrais positifs, faux positifs et faux négatifs.

Vrai positif :

Nous pouvons observer les performances de notre méthode dans les figures ci-dessous en détectant que les panneaux circulaires.



Figure 41. Détection des panneaux circulaires

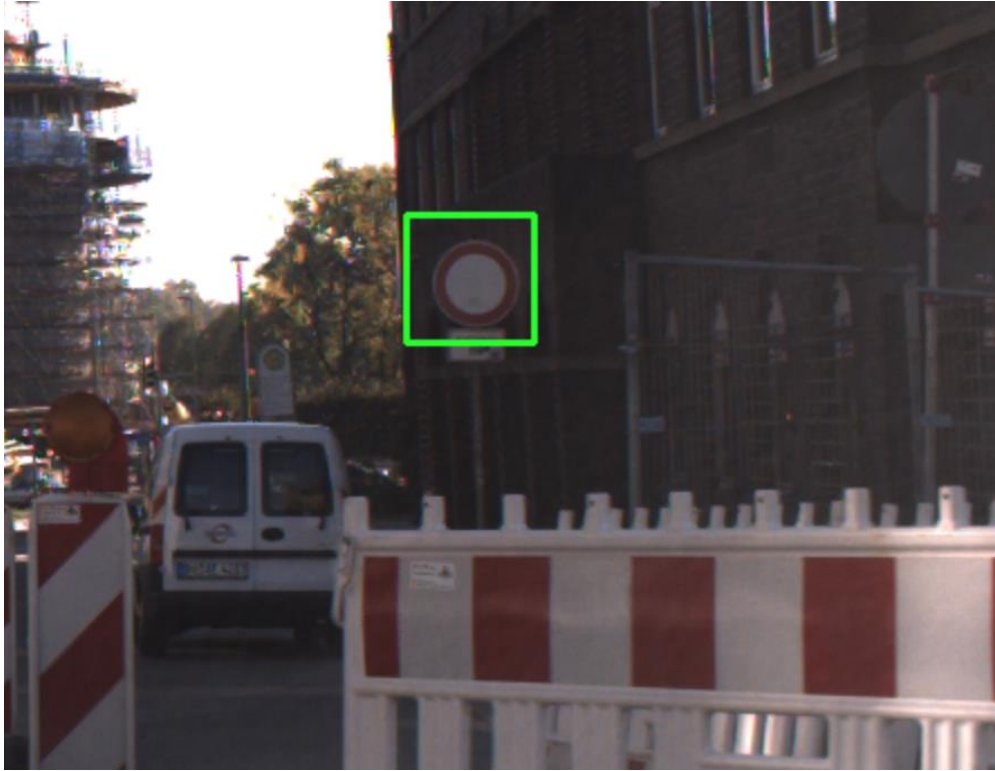


Figure 42. Détection des panneaux circulaires

Faux positif :

Dans les figures ci-dessous on peut voir des fausses détections, durant le calcul des pourcentages nous avons remarqué la dominance des fausses détections des roues de voitures ou des motos.



Figure 43. Fausses détections

Faux négatif :

Dans les figures ci-dessous on peut voir des fausses non-détections, durant le calcul des pourcentages nous avons remarqué la dominance des fausses détections des panneaux circulaires de couleur bleu.



Figure 44. Fausses non-détections

Chapitre 4 : conclusion

Suite à de longues séries de testing sur les méthodes basées sur le traitement d'image nous n'avons malheureusement pas réussi à obtenir des résultats satisfaisants. La recherche d'une meilleure méthode était la phase la plus difficile car on manquait de temps et de ressources, mais finalement nos efforts ont payé, d'où la bonne performance de notre méthodes basée sur l'apprentissage M.L. malgré les différents obstacles rencontrés (luminosité, ...).

Conclusion Générale

Nous avons présenté dans un premier temps une introduction au traitement d'image, sa définition, les différents domaines de son utilisation, puis on a présenté les différentes notions, méthodes et outils de traitement d'image qui sont nécessaires pour la réussite de nos méthodes. On a aussi introduit et détaillé la notion de l'apprentissage M.L., sa définition, ses types et aussi ses problèmes.

Dans un autre temps, on a représenté les différentes méthodes que nous avons proposées, sur quoi elles se basent, la façon dont elles marchent étape par étape, ensuite on a fait une comparaison des résultats et performances produites par chaque méthode et des figures pour accompagner ces derniers, et finalement une conclusion dont laquelle on présente nos derniers avis.

La vision par ordinateur et l'apprentissage M.L. peuvent jouer un grand rôle dans la réduction et la prévention des accidents routiers, et éventuellement sauver de milliers de vies et de ressources, et leur rôle ne s'arrête pas que dans ce domaine, bien au contraire la vision par ordinateur joue des rôles dans différents domaines tels que la médecine et la sécurité.

Références

BakIr, G., Hofmann, T., Schölkopf, B., Smola, A. J., Taskar, B., et Vishwanathan, S. V. N. (2007).

Predicting Structured Data. MIT Press, Cambridge, MA. <https://mitpress.mit.edu/books/predicting-structured-data>.

Barto, R. S. et Sutton A. G. (1998). Reinforcement Learning : An Introduction. MIT Press, Cambridge, MA. <http://incompleteideas.net/book/the-book-2nd.html>.

Benureau, F. (2015). Self-Exploration of Sensorimotor Spaces in Robots. Thèse de doctorat, Université de Bordeaux.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. IBM Journal of Research and Development, 44(1.2) :206–226.

Scott, D. W. (1992). Multivariate density estimation. Wiley, New York

Guillaume Cleuziou, Une méthode de classification non-supervisée pour l'apprentissage de règles et la recherche d'information, 10 juillet 2006

Pierre-Louis GONZALEZ, MÉTHODES DE CLASSIFICATION, Cnam, 2008

URL: https://fr.wikipedia.org/wiki/Caract%C3%A9ristiques_pseudo-Haar

URL: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

URL: <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf>

URL: <http://www.maia.ub.es/~sergio/files/Transport09.pdf> [2] « Combining Adaboost Learning and Evolutionary Search to select Features for Real-Time Object Detection », Andre Treptow and Andreas Zell, Evolutionary Computation, 2004. CEC2004.

Artificial Intelligence laboratory, August 2003 URL:
<http://dspace.mit.edu/bitstream/handle/1721.1/7276/AIM-2003-018.pdf?sequence=2>
« Explaining AdaBoost », Robert E.