وزارة التعليم العـالي والبحـث العـلمـي
Ministère de l'enseignement supérieur et de la recherche scientifique
جـامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA
كلية التكنولوجيا
Faculté de Technologie
قسم الإلكترونيك
Département d'Électronique

Mémoire de Master

Détection et poursuite d'objets par les méthodes de viola-Jones et KLT

Filière Électronique
Spécialité Electronique des Systèmes Embarqués

Réalisé par :

KARIM Mounia Zhoor

&
ADADAINE Amina

Encadré par :

Mme. BOUGHERIRA Hamida

Année Universitaire 2019-2020

Je dédie ce modeste travail

À mes très chers parents

À ma chère sœur

À mon fiancé

À toute ma famille et mes amis

À tous ceux qui m'ont encouragé et Soutenu…

Mounia

Je dédie ce travail,

Fruit de nombreuses années d'étude à :

-Tous mes enseignants.

-Mes chers parents, pour leur sacrifice, leurs encouragements et conseils.

- Mon mari, pour son encouragement et son aide.

- Ma chère fille Sidra.

- Mon frère Hemza, Ma sœur Ouiame et toute ma famille.

Je tiens, particulièrement, à remercier mon encadreur Madame Bougherira Hamida
J'exprime toute ma gratitude et mon respect aux membres de jury qui me feront l'honneur.

Amina

Résumé :

La vision par ordinateur est une branche de l'intelligence artificielle dont le but est de permettre à une machine de comprendre ce qu'elle «voit » lorsqu'on la connecte à une ou plusieurs caméras.
Avec la généralisation de l’utilisation des images numériques, l’analyse du mouvement dans les vidéos s’est révélée être un outil indispensable pour des applications aussi diverses que la vidéo surveillance, la compression vidéo, l’imagerie médicale, la robotique, l’interaction homme machine
Ce projet de fin d’étude consiste en une étude de différentes méthodes de détection et suivi d’un ou plusieurs objets dans une vidéo. Nous sommes intéressés par la détection des visages et d’objets par les méthodes viola-Jones, klt , détection selon la couleur RGB nous avons implémenté ces méthodes et nous les avons appliqué avec succès.
Mot clés : détection de visage, détection d’objets, méthode viola-Jones, KLT algorithme, méthode des couleur, suivi de visage et objets, matlab.

ملخص:

الرؤية الحاسوبية هي فرع من فروع الذكاء الاصطناعي ، والهدف منها هو تمكين الجهاز من فهم ما "يراه" عندما يكون متصلاً بكاميرا واحدة أو أكثر.
مع الاستخدام الواسع لنطاق الصور الرقمية ، أثبت تحليل الحركة في مقاطع الفيديو أنه أداة لا غنى عنها لتطبيقات متنوعة مثل المراقبة بالفيديو ، وضغط الفيديو ، والتصوير الطبي ، والروبوتات ، وما إلى ذلك، التفاعل بين الإنسان والآلة
يتكون مشروع نهاية الدراسة هذا من دراسة طرق مختلفة لاكتشاف وتعقب كائن أو أكثر في مقطع فيديو. نحن مهتمون باكتشاف الوجوه والأشياء من خلال طرق فيولا جونز ، klt، والكشف حسب اللون RGB قمنا بتطبيق هذه الطرق وقمنا بتطبيقها بنجاح.

Abstract :

Computer vision is a branch of artificial intelligence, the goal of which is to enable a machine to understand what it "sees" when it is connected to one or more cameras.
With the widespread use of digital images, motion analysis in videos has proven to be an indispensable tool for applications as diverse as video

surveillance, video compression, medical imaging, robotics, etc. man-machine interaction

This end-of-study project consists of a study of different methods of detecting and tracking one or more objects in a video. We are interested in the detection of faces and objects by the viola-Jones methods, klt, detection according to the color RGB we have implemented these methods and we have applied them successfully.

Introduction générale :

La vision par ordinateur est devenue une branche de l'intelligence artificielle et a pour but de permettre à une machine de comprendre ce qu'elle «voit » lorsqu'on la connecte à une ou plusieurs caméras. Et aussi c'est un domaine qui consiste à réaliser des algorithmes capables de traiter et d'analyser des images ou des séquences d'images, et de résoudre des problèmes tels que reconnaître un visage, suivre un objet, le localiser dans l'espace...etc en temps réel, sur des systèmes embarqués, et éventuellement connectés. Ces algorithmes doivent donc être rapides et efficaces.

La détection des objets en mouvement, est la détection d'un ensemble de régions d'intérêt en mouvement dans la scène observée.

La poursuite d'objets est équivalente au suivi de primitives ou de régions, et a pour but de déterminer la position de chaque primitive ou région dans l'image à chaque instant.

La détection et Le suivi d'un objet dans une image numérique reste encore un problème sensible quand il s'agit d'obtenir des résultats précis parce qu'il existe différents mouvements et scènes.

Les applications de la détection d'objets sont nombreuses. On peut notamment citer le suivi d'objets, l'analyse du comportement, la compression vidéo, la reconstruction 3D...etc.

Les techniques de détection du mouvement peuvent être considérées comme un prétraitement permettant de réduire la quantité d'informations à analyser par rapport un objet en mouvement est détecté si sa position change relativement à celle d'un ensemble d'objets statiques ou s'il est localisé dans une image acquise à l'instant t de la séquence à une position différente de celle qu'il occupait dans l'image précédente.

La problématique détection des objets en mouvement, est en général une première étape pour des outils automatiques de vision par ordinateur. Ces outils peuvent avoir pour vocation, soit uniquement de détecter, soit de détecter et reconnaître, soit de détecter et suivre des objets pour, par exemple, analyser le comportement ou la trajectoire de ces objets.

La problématique de poursuite d'objet est aussi importante et nécessite la mise en place de méthodes simples et robustes. Tous ces sujets font l'objet

d'un grand nombre de travaux, mais il n'existe pas, à l'heure actuelle, d'algorithmes aboutis s'adaptant à n'importe quelle situation.

L'objectif de notre projet est d'implémenter un ensemble d'algorithmes pour la détection et la poursuite d'objets, et de visages dans une séquence d'images ou une vidéo.

Nous avons étudié trois méthodes de détection et poursuite des objets en mouvement dans une scène vidéo: la méthode de viola-jones ,la méthode de Kanade-Lucas-Tomasi (KLT) et la méthode de détection de couleur .

Dans notre travail, nous nous intéressons à la méthode de détection d'objets de Viola-Jones [17]. Développée par Paul Viola et Michael Jones en 2001, cette méthode peut détecter rapidement et précisément les objets dans les images et fonctionne particulièrement bien avec le visage humain (Viola & Jones, 2001). Malgré son âge, cette méthode est toujours d'actualité dans la détection de visage aux côtés de plusieurs méthodes de l'intelligence artificielle comme les CNN (Convolutional Neural Networks). Le cadre de détection d'objets, Viola-Jones combine les concepts de fonctionnalités (que nous implémentons dans notre projet) de type *Haar* , d' *images intégrales* , de *l'algorithme AdaBoost* et *de classificateurs en cascade* pour créer un système de détection d'objets rapide et précis .

Une fois les caractéristiques de l'objet extraites par la méthode de Viola-Jones, nous utilisons la méthode KLT(Kanade-Lucas-Tomashi) [18], pour poursuivre l'objet dans une vidéo, et la méthode de détection et poursuite selon la couleur (RGB)[19]

Notre mémoire est composé de trois chapitres :

Chapitre1 : Nous parlons de généralités sur la détection et poursuite d'objets et présentons l'état de l'art de quelques méthodes de détection et de poursuite, avec leurs principes, avantages, et inconvénients.

Chapitre2 : nous développons les méthodes, que nous avons retenues, de détection et poursuite d'objet en mouvement.

Chapitre3 : conclut ce mémoire en présentant l'implémentation des algorithmes, et l'ensemble des résultats expérimentaux obtenus pour la détection et le suivi d'objets dans une séquence vidéo

Ce travail est terminé par une conclusion générale et une bibliographie.

Liste des figures

Liste des figures

Liste d’abréviation

Fps:frame par seconde

R-CNN: Region-based Convolutional Neural Networks

SSD:solid-state drive

CIE:Commission Internationale de l'Eclairage

GRB: red green blue

1.1 Introduction :

Ce projet se concentre principalement sur deux taches la détection et la poursuite des visages et d'objets. Il existe dans la littérature une multitude de méthode de détection et poursuite de visage.

Ce chapitre introductif présent la détection d'objets en mouvement dans une image numérique et présente des différentes méthodes classiques, on étudie la technique de détection de visage et d'objets.

1.2 La détection et la poursuite :

En vision par ordinateur on désigne par détection d'objet (ou classification d'objet) une méthode permettant de détecter la présence d'une instance (reconnaissance d'objet) ou d'une *classe d'objets* dans une image numérique. Une attention particulière est portée à la détection de visage et la détection de personne

1.2.1 Détection de visage :

La détection de visage a de très nombreuses applications directes en vidéosurveillance, biométrie, robotique, commande d'interface homme-machine, photographie, indexation d'images et de vidéos, recherche d'images par le contenu, etc. Elle permet également de faciliter l'automatisation complète d'autres processus comme la reconnaissance de visage ou la reconnaissance d'expressions faciales. [1]

Figure1.1 : détection de visage par viola-Jones

1.2.2 Détection d'objet :

La détection d'objets est un domaine très actif de la recherche qui cherche à classer et localiser des régions/zones d'une image ou d'un flux vidéo. Ce domaine est à la croisée de deux autres : la classification d'image et la localisation d'objets. En effet, le principe de la détection d'objets est le suivant : pour une image donnée, on recherche les régions de celle-ci qui pourraient contenir un objet puis pour chacune de ces régions découvertes, on l'extrait et on la classe à l'aide d'un modèle de classification d'image – par exemple –. Les régions de l'image d'origine ayant de bons résultats de classification sont conservés et les autre jetés. Ainsi, pour avoir une bonne méthode de détection d'objets, il est nécessaire d'avoir un algorithme solide de détection de régions et un bon algorithme de classification d'images**. [2]**

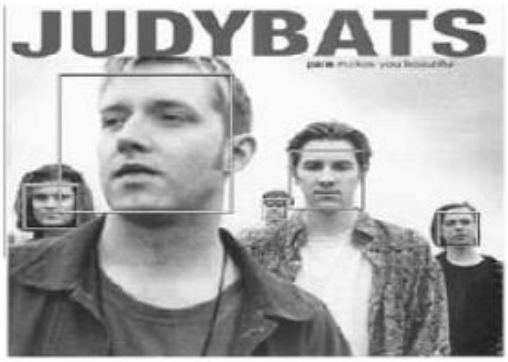


Figure1. 2 : Détection de visage par la méthode de Viola et Jones. [1]

1.2.3 Définition :

1.2.3.1 images numériques :

Une image numérique est une image (dessin, icône, photographie, ...) crée, traitée, stockée sous forme binaire (suite de 0 et de 1). Lorsqu'on agrandit une image numérique, on voit que celle-ci est composée d'un ensemble de "points", appelés pixels [3]

1.2.3.2 pixels :

Le pixel (abréviation venant de l'anglais : Picture élément) est l'élément de base d'une image ou d'un écran, c'est-à-dire un point. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image.[3]

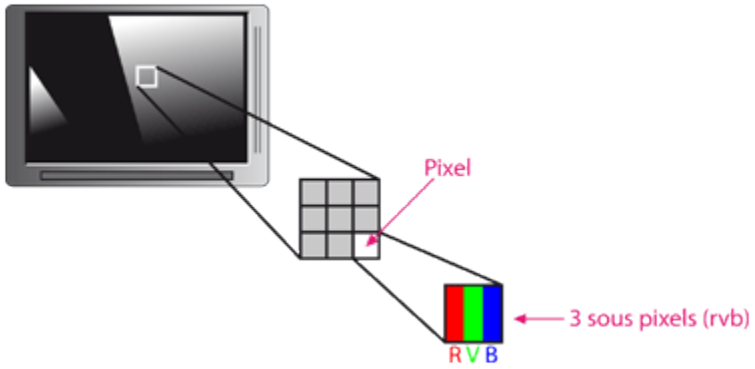


Figure1.3: le pixel sur un écran [20]

1.2.3.3 Vidéo :

Technique ou ensemble de techniques permettant la formation, l'enregistrement, le traitement, la transmission ou la reproduction d'image de télévision ou d'image analogues ou de signaux occupant une largeur de bande comparable sur un écran de visualisation. [4]

La vidéo est une succession d'images animées défilant à une certaine cadence afin de créer une illusion de mouvement pour l'œil humain.
Elle peut être analogique (signal continu d'intensité de luminance) ou numérique (suite de trames ou images). [5]

1.2.3.4 Frame par seconde(Fps) :

Frame par seconde soit le nombre d'image par seconde cette valeur définit la fluidité d'un jeu si cette valeur descend trop bas l'ordinateur commence à ramer. Terme très employé dans les jeux vidéo [6] .pour la vision humaine on utilise 24 im/s en raison de phénomène de persistance rétinienne.

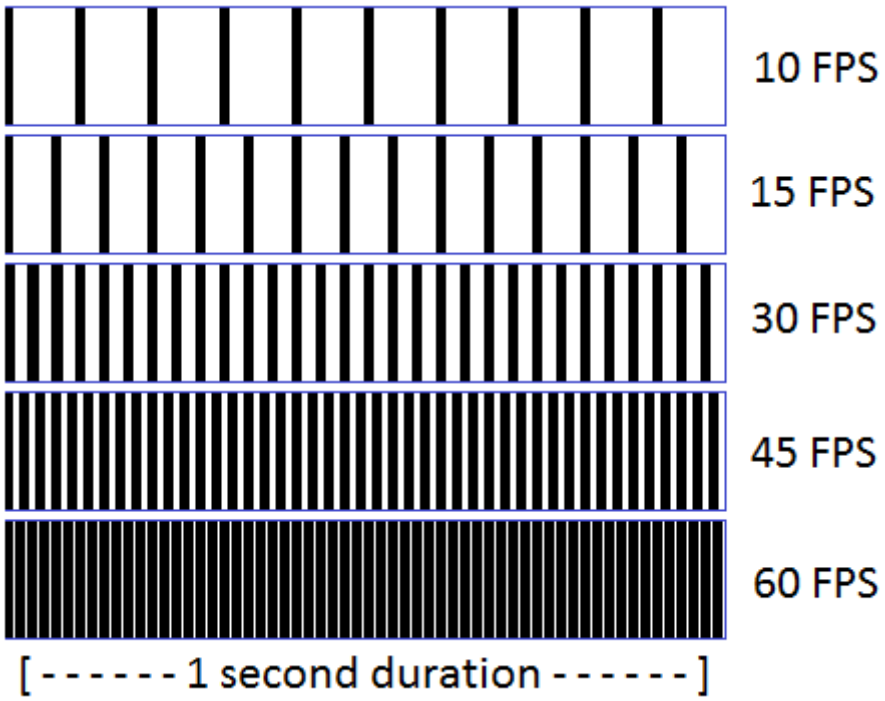


Figure1.4 : Frames Par Second [21]

1.2.4 Poursuite de cible :

Lorsqu'un objet bouge devant une caméra ou lorsque la caméra bouge, il se produit une variation de l'illumination du capteur d'image (rétine) modifiant la distribution des niveaux de gris [7]

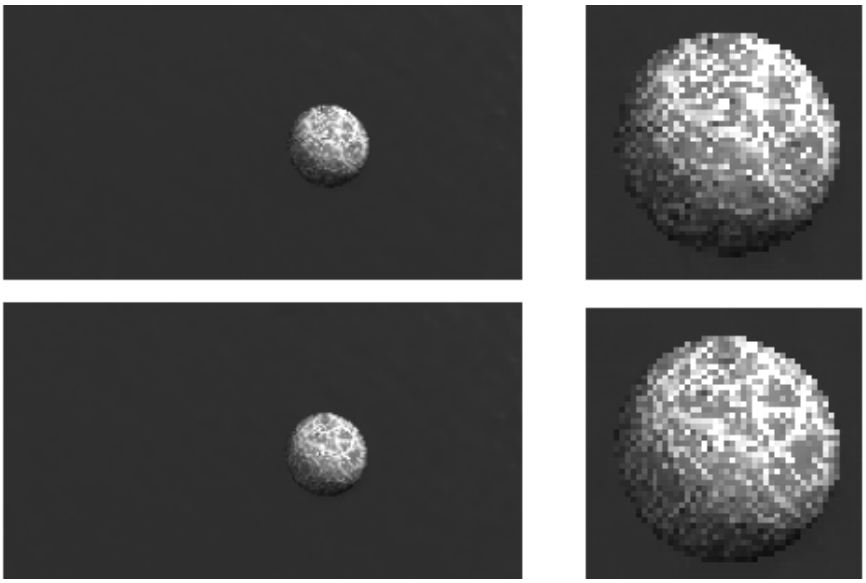


Figure 1.5 : deux images consécutives de la séquence "Boule de marbre" et détails.[7]

La poursuite de cible est une technique permettant de trouver, dans une séquence d'images, la position d'une zone prédéfinie correspondant à un motif particulier dont on souhaite connaître la position. [7]

1.2.5 Reconnaissance d'objets (l'aspect général) :

La reconnaissance d'objets est un domaine très actif en vision par ordinateur. L'approche générale consiste à considérer que l'on dispose d'une banque de données où sont stockés des modèles d'objets et une vue du monde réel. On doit répondre à une question essentielle: L'objet extrait (par les moyens précédents) existe-t-il dans la banque de données. Le problème dépend de la modélisation d'objets adoptée. Dans la majorité des cas cette modélisation est purement géométrique: un ensemble de caractéristiques appelées primitives. L'opération de reconnaissance consiste en l'appariement des ces caractéristiques et celles décelées dans l'image. [8]

1.3 Détection et la poursuite d'objet :

Depuis quelques années la détection des formes, des objets ainsi que celle des visages sont prises comme domaines de recherche par plusieurs personnes et sociétés ce qui amène à l'existence de plusieurs approches pour la détection des visages et des objets dans une image. [1]

1.3.1 Méthode détection de point d'intérêt : [9]

La détection de points d'intérêt (ou coins) est une étape préliminaire à de nombreux processus de vision par ordinateur. Les points d'intérêts, dans une image, correspondent à des doubles discontinuités de la fonction d'intensités. Celles-ci peuvent être provoquées, comme pour les contours, par des discontinuités de la fonction de réluctance ou des discontinuités de profondeur. Ce sont par exemple : les coins, les jonctions en T ou les points de fortes variations de texture. [9]

Figure1.6 : Différents types de points d'intérêts : coins, jonction en T et point de fortes variations de texture. [9]

1.3.1.1 Avantages des points d'intérêt :

1. Sources d'informations plus fiable que les contours car plus de contraintes sur la fonction d'intensité.

2. Robuste aux occultations (soit occulté complètement, soit visible).

3. Pas d'opérations de chainage (-> contours !).

4. Présents dans une grande majorité d'images. [10]

1.3.1.2 Différentes approches pour la détection des points d'intérêt :

De nombreuses méthodes ont été proposées pour détecter des points d'intérêts. Elles peuvent être classées grossièrement suivant trois catégories :

1. Approches contours : l'idée est de détecter les contours dans une image dans un premier temps. Les points d'intérêts sont ensuite extraits le long des contours en considérants les points de courbures maximales ainsi que les intersections de contours. [9]

2. Approches intensité : l'idée est cette fois-ci de regarder directement la fonction d'intensité dans les images pour en extraire directement les points de discontinuités. [9]

3. Approches à base de modèles : les points d'intérêts sont identifiés dans l'image par mise en correspondance de la fonction d'intensité avec un modèle théorique de cette fonction des point d'intérêts considérés. [9]

➔ Les approches de la deuxième catégorie sont celles utilisées généralement. Les raisons sont : indépendance vis à vis de la détection de contours (stabilité), indépendance vis à vis du type de points d'intérêts (méthodes plus générales).[9]

1.3.1.3 le détecteur de Harris : [16]

Le détecteur de Moravec fonctionne dans un contexte limité. Il souffre en effet de nombreuses limitations. Harris et Stephen ont identifié certaines limitations et en les corrigeant en ont déduit un détecteur de coins très populaire : le détecteur de Harris. La fenêtre rectangulaire et binaire utilisée par l'opérateur de Moravec est remplacée par une fenêtre circulaire de type gaussienne afin de réduire le bruit au niveau de la réponse du détecteur.

Algorithme

1- Calcul des dérivées premières

2- Calcul des matrices d'auto-corrélation

3- Calcul de la réponse

4- Extraction des points d'intérêt Suppression du non-maxima Sélection des points d'intérêt. [16]

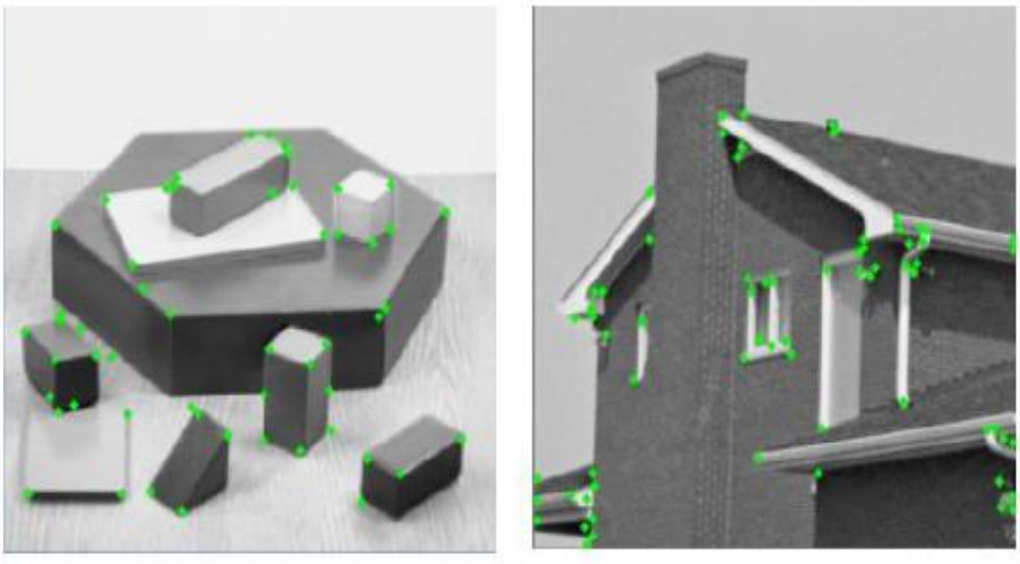


Figure 1.7 : Détecteur de Harris [16]

1.3.2 Méthode de détection de contours :[11]

La détection est basée sur la dérivation selon les deux coordonnées. Si on considère classiquement les signaux comme des sommes de sinusoïdes, la dérivation apparaît comme un filtre passe-haut qui introduit donc du bruit à l'origine de faux contours. Pour l'amateur il est recommandé, avant d'utiliser un filtre simple, d'atténuer ce bruit par passage dans un filtre flou. Des méthodes plus élaborées ont été systématisées pour les professionnels. [11]

- **Filtre dérivées premières :** Le filtre le plus simple consiste à calculer les différences entre pixels voisins sur les horizontales puis sur les verticales. Chaque extremum correspond à un point d'un contour.[11]
- **Filtre de Sobel *:*** la technique précédente est améliorée en remplaçant le filtre rectangulaire par un filtre triangulaire.[11]
- **Filtre dérivées secondes :** celles-ci se calculent simplement en différences finies et c'est maintenant un changement de signe qui correspond à un point d'un contour. On les utilise généralement à travers leur somme qui est le laplacien.[11]

1.3.3 Méthode base sur les réseaux de neurones artificiels et le Deep Learning :

Aujourd'hui, les réseaux de neurones ont de nombreuses applications dans des domaines très variés :

- traitement d'image : compression d'images, reconnaissance de caractères et de signatures, reconnaissance de formes et de motifs, chiffrement, classification, ...
- traitement du signal : traitement de la parole, identification de sources, filtrage, classification,

Les réseaux de neurones sont utilisés comme méthode d'apprentissage en profondeur (deep Learning), l'un des nombreux sous-domaines de l'intelligence artificielle. Ils ont été proposés pour la première fois il y a environ 70 ans pour tenter de simuler le fonctionnement du cerveau humain, mais sous une forme beaucoup plus simplifiée. Des «neurones» individuels sont connectés en couches, des pondérations étant attribuées pour déterminer la façon dont le neurone répond lorsque des signaux sont propagés à travers le réseau. Auparavant, les réseaux de neurones étaient limités par le nombre de neurones qu'ils pouvaient simuler et, par conséquent, par la complexité de l'apprentissage qu'ils pouvaient atteindre. Mais au cours des dernières années, grâce aux progrès réalisés dans le développement de matériel, nous avons pu construire des réseaux très profonds et les former sur d'énormes bases de données afin de réaliser des avancées décisives en matière d'intelligence artificielle. [12]

Ces avancées ont permis aux machines d'égaler et de dépasser les capacités de l'homme à effectuer certaines tâches (deep Learning). Une de ces tâches est la reconnaissance d'objet. Bien que les machines aient toujours été incapables de faire correspondre la vision humaine, les progrès récents en apprentissage en profondeur ont permis de créer des réseaux de neurones capables de reconnaître des objets, des visages, du texte et même des émotions.

- Objets représentes par un ensemble de segments de droite.

• Aucune étape pour la prédiction.

• Appariement se fait par un réseau de Hopfield.[12]

pour avoir une bonne méthode de détection d'objets, il est nécessaire d'avoir un algorithme solide de détection de régions et un bon algorithme de classification d'images.

La clef de la réussite est détenue dans l'algorithme de classification d'image. Depuis les résultats du challenge Imagent 2012, le deep Learning (et notamment les réseaux de convolution) est devenue la méthode number #1 pour résoudre ce genre de problème. Les recherches en détection d'objets ont donc tout naturellement intégré les modèles de classification d'image, ce qui a permis de créer les réseaux tels que SSD et R-CNN (le R signifiant ici Région).

L'image illustre le résultat d'une détection d'objets (de voitures en l'occurrence). On observe que plusieurs objets peuvent être découverts et localisés dans une même image.

SSD et R-CNN ne sont jamais utilisés seuls, ils sont couplés avec un réseau pré-calculé de classification d'image. Un réseau auquel on a calculé les poids est appelé un modèle. Il existe ainsi des SSD Mobile net, des R-CNN Inception, etc. [2]

• **Avantages de détection de nuerons**

- Appariement plus robuste.
- Résiste au changement de mouvements [1]

• **Inconvénients de détection de nuerons**

- Méthode couteuse en temps de calcule sur machine Conventionnelle.
- Problème dans le choix de paramètre [1]

Figure 1.8 : Object détection using deep Learning [22]

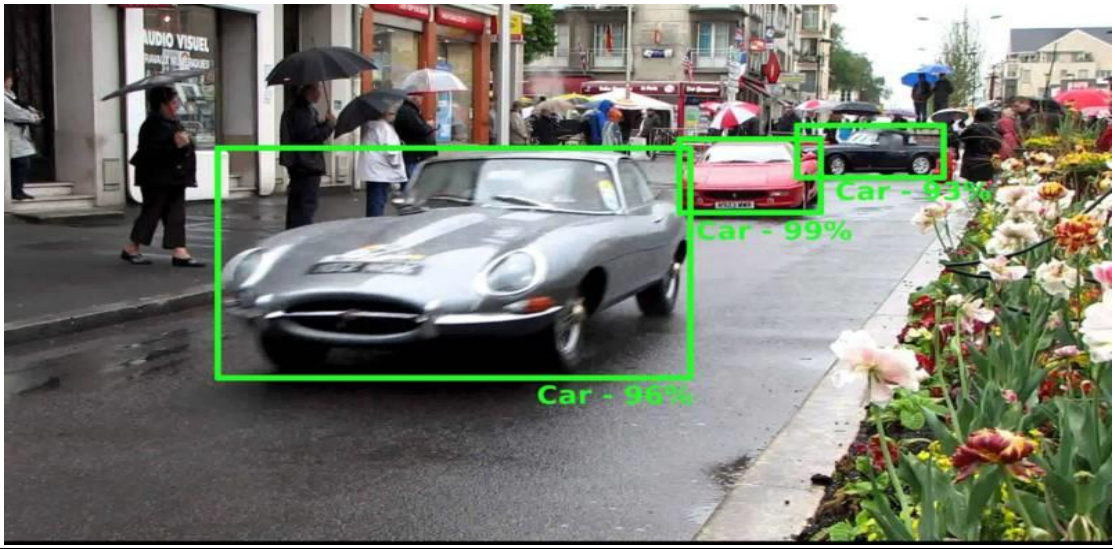


Figure 1.9: Exemple de détection d'objets par un modèle de deep learning[2]

1.3.4 Méthode de détection par la couleur de peau (Skin color) :

Pour les visages la couleur de la peau de l'être humain est une caractéristique spécifique c'est pour ça elle a été utilisée pour la détection des visages. En effet la couleur de la peau est différente

selon les personnes et leur origine (Africain, Européen, Asiatique...). Dans ce contexte plusieurs études ont démontré que la plus grande différence s'étend largement entre intensités plutôt que la chrominance lumineuse.[13]

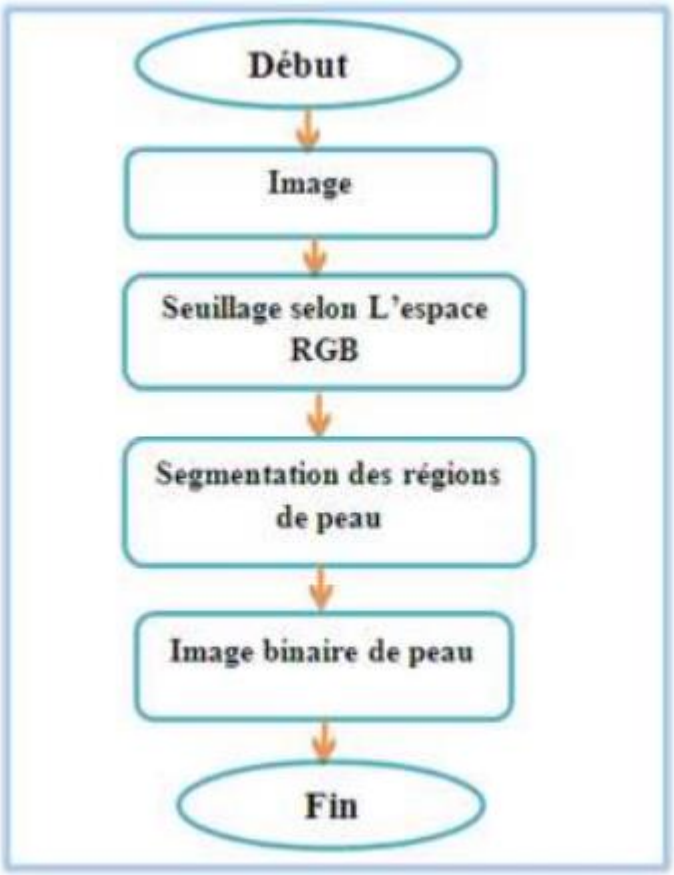


Figure 1.10 **:** Illustration du processus de détection de peau [13]

Figure 1.11 : La détection de la couleur de peau [13]

1.3.4.1 détection de couleur de peau par l'espace RGB :[13]

Le codage RGB, mis au point en 1931 par la Commission Internationale de l'Eclairage (CIE) consiste à représenter l'espace des couleurs à partir de trois rayonnements monochromatiques de couleurs : • Rouge (de longueur d'onde égale à 700,0 nm), • Vert (de longueur d'onde égale à 546,1 nm), • Bleu (de longueur d'onde égale à 435,8 nm). Etant donné que le codage RGB repose sur trois composantes proposant la même gamme de valeur, on le représente généralement graphiquement par un cube dont chacun des axes correspond à une couleur primaire (figure1.12)

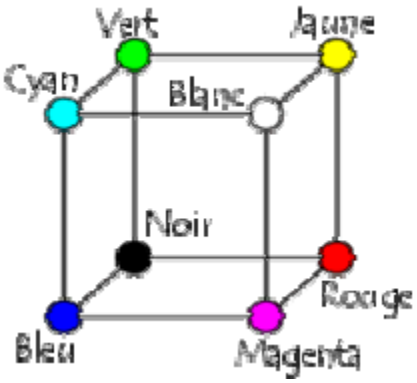


Figure 1.12: Présentation des couleurs dans l'espace RGB [13]

1.3.4.2 Seuillage selon l'espace RGB :[13]

La détection de la couleur de peau en utilisant l'espace RGB ne nécessite aucun modèle de peau et aucune transformation des couleurs, elle vérifie simplement une suite des contraintes pour décider si un triplet de couleur (R, G, B) représente une couleur de peau ou non, ces contraintes sont divisées en deux, selon l'état de l'éclairage, fort (jour) ou faible (nuit)

Eclairage fort :

(R > 95) ET (G > 40) ET (B > 20) ET

(MAX(R, G, B) - MIN(R, G, B) > 15) ET

(ABS(R-G) > 15) ET (R > G) ET (R > B)

Eclairage faible :

((R > B) ET (G > B)) OU ((R > 220) ET

(G > 210) ET (B > 170) ET (ABS(R-G) <= 15))

1.3.4.3 Segmentation des régions de peau : [13]

Puisque les régions de peau sont plus lumineuses que les autres parties des images, ces régions peuvent être segmentées du reste par un seuillage. Ce processus produit une image binaire dont les "1" représentent les Pixels supérieurs à un certain seuil, et les "0" représentent les pixels inférieurs à ce seuil. On peut formuler cette procédure par :

x < Seuil => x=0

x ≥ Seuil => x=1

Tel que x représente les valeurs des Pixels de l'image résultat du seuillage dans l'espace RGB. Le seuil optimal choisi est pris dans l'intervalle 0.35 à 0.45. Un des résultats est illustré dans la figure1.13

Image originale Image de peau

Figure 1.13 : Transformation de l'image originale en image de peau[13]

1.3.4.4 Filtrage des régions de peau :[13]

En utilisant le résultat de la détection de la peau, nous procédons à l'extraction des régions qui peuvent probablement contenir un visage humain. Ce processus se résume par les étapes suivantes :

• Application des opérateurs morphologiques.

• Etiquetage de l'image binaire.

• Application des contraintes de ratio et des contraintes de surface sur chaque segment.

1.3.5 la détection de visage par Modèles géométriques :[1]

Pour profiter des informations des caractéristiques visuelles du visage, les chercheurs utilisent des points caractéristiques du visage pour former un modèle géométrique du visage. Pour ce faire, ils ont extrait les 12 points suivants :

• Les quatre coins des yeux : A, B, C, et D.

• Le point S centre de gravité de A, B, C et D (milieu de la tête).

• Le point G à la base du nez.

• Les points extrêmes droit et extrême gauche du nez : E et F.

• Le point H milieu du contour supérieur de la lèvre supérieure.

• Les deux coins de la bouche I et J.

• Le M à l'intersection des droites (SG) et (IJ) représentant le milieu de la bouche.[1]

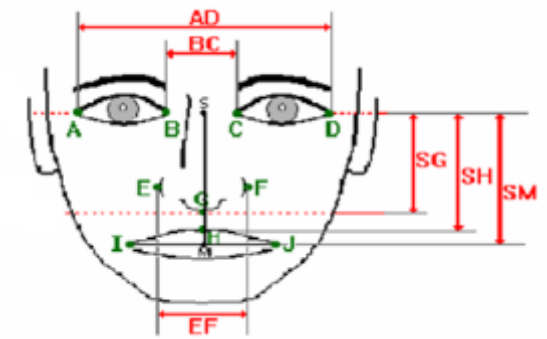


Figure 1.14 : Modèle géométrique du visage.[1]

Ils se basent sur les heuristiques au sujet de l'aspect de visages.il peut être facile de créer une afin de décrire le visage humain, mais ce qui n'est pas évident c'est la traduction de ces heuristiques en des règles de classification d'une qualité efficace. [1]

Pour une image d'entrée, les caractéristiques faciales modélisées par ces règles sont tout d'abord extraites, puis codées afin de déterminer les candidats de visage. [1]

Par exemple, un visage apparaît généralement dans une image avec une bouche, un nez, deux yeux et deux sourcils symétriques. Les rapports reliant ces caractéristiques peuvent être dessinés par leurs

distances et positions relatives. Mais, il n'est pas toujours évident de créer une règle adéquate qui contient uniquement une classe de visages.[1]

La difficulté de modélisation des connaissances humaines dans des règles adéquates demeure l'entrave de cette méthode. Par conséquent, les règles strictes ne garantissent pas la détection de visages qui ne répond pas à toutes ces règles. Les règles générales aboutissent, de leur part, à de nombreuses fausses détections.[1]

De plus, il est difficile de généraliser cette approche pour la détection des visages en différentes poses en raison des problèmes de génération des règles pour tous les cas possibles. En contrepartie, cette technique a fourni de bons résultats pour la détection de visages en vue de face dans des scènes simples, mais reste limitée au niveau d'extraction des caractéristiques faciales [1]

1.3.6 Méthode de Viola et Jones : [14]

L'algorithme de Viola-Jones tire son nom de deux chercheurs en vision par ordinateur qui ont proposé la méthode en 2001: Paul Viola et Michael Jones.

Ils ont développé un cadre général de détection d'objets capable de fournir des taux de détection d'objets compétitifs en temps réel. Elle fait partie des toutes premières méthodes capables de détecter efficacement et en temps réel des objets dans une image Il peut être utilisé pour résoudre divers problèmes de détection**,** mais la principale motivation vient de la détection des visages.
L'algorithme Viola-Jones comporte 4 étapes principales :

1. Sélection des fonctionnalités de type Haar
2. Créer une image intégrale
3. Exécution de la formation AdaBoost
4. Création de cascades de classificateur

Étant donné une image, l'algorithme examine de nombreuses sous-régions plus petites et essaie de trouver un visage en recherchant des caractéristiques spécifiques dans chaque sous-région. Il doit vérifier de

nombreuses positions et échelles différentes, car une image peut contenir de nombreux visages de différentes tailles. Viola et Jones ont utilisé des fonctionnalités de type Haar pour détecter les visages. [14]

Viola et Jones l'on généralisé en 2005 pour détecter d'autres types de cibles, comme des voitures ou des avions et même des personnes.
En tant que procédé d'apprentissage supervisé, la méthode de Viola et Jones nécessite de quelques centaines à plusieurs milliers d'exemples de l'objet que l'on souhaite détecter, pour entraîner un classifieur[2]. Une fois son apprentissage réalisé, ce classifieur est utilisé pour détecter la présence éventuelle de l'objet dans une image en parcourant celle-ci de manière exhaustive, à toutes les positions et dans toutes les tailles possibles.[11]

Considérée comme étant l'une des plus importantes méthodes de détection d'objet, la méthode de Viola et Jones est notamment connue pour avoir introduit plusieurs notions reprises ensuite par de nombreux chercheurs en vision par ordinateur, à l'exemple de la notion d'image intégrale ou de la méthode de classification construite comme une cascade de classifieurs boostés. [11]

1.3.7 Étude comparative qualitative de le tableau (comparaison des méthodes de détection d'objets)pour montre les avantages et les inconvénients des méthodes que nous avons présentées :

Les méthodes	Avantage	Inconvénient
Point d'intérêt	-Présents dans une grande majorité d'images. -Sources d'informations plus fiable que les contours car plus de	

	contraintes sur la fonction d'intensité.[10]	
Réseau de neurones	- Résiste au changement de mouvements - Appariement plus robuste[1]	- Méthode couteuse en temps de calcul sur machine conventionnelle. - Problème dans le choix de paramètre[1]
Viola-Jones	- utilise adaboost Rapide	Approche indirecte _ Chaque nouvel apprentissage est très coûteux en temps de calcul _ Repose sur le fait d'avoir un grande nombre d'attributs et d'échantillons classés **(expertisés**
contour	-son implémentation est très facile a mettre en œuvre et son cout en calcul très faible un seul paramètre est nécessaire pas de seuil significatif rapidité, précision [15]	-Plus grand sensibilité au bruit [15]
Les couleurs	Rapidité, détection de la peau [1]	Les yeux, l'arrière-plan [1]

Tableau de comparaison des méthodes de détection d'objet table1.1

1.4 La poursuite d'objets :

Après avoir détecté les objets dans une séquence d'image, nous nous intéressons dans cette partie à la phase de suivi. [5]

La problématique de poursuite d'objet est aussi importante et nécessite la mise en place de méthodes simples et robustes et reste encore un problème sensible quand il s'agit d'obtenir des résultats précis parce qu'il existe différents mouvements et scènes.

1.4.1 Algorithmes de suivi :

Le suivi d'objets dans une séquence d'images est un problème classique et largement étudié dans le domaine de l'analyse du mouvement, du fait de ses nombreuses applications parmi lesquelles on peut noter la surveillance vidéo la robotique, la compression de vidéos, l'étude de la déformation d'objets, . . . Ainsi, de nombreux algorithmes de suivi ont été proposés ces dernières années. [5]

1.4.2 Classification des algorithmes de suivi :

Les algorithmes de suivi peuvent globalement être classés en deux catégories :

Les algorithmes orientés mouvement, et ceux orientés modèle. [5]

1.4.3 La poursuite d'objets par KLT algorithme :

La méthode appelée KLT Tracking est une méthode de suivie d'objet en mouvement dans une séquence vidéo. Elle a été mise au point par trois hommes: Kanade, Lucas et Tomaci d'où le nom de la méthode.

La genèse de cette méthode provient de Kanade et Lucas au travers de leurs premiers algorithmes publiés à partir de 1981. Avec le concours de Tomasi, cette méthode a évoluée et est devenu la méthode de tracking KLT (Kanade - Lucas - Tomaci).[24]

Le principe de cette méthode est le suivant:

- Sur la première image d'une séquence vidéo, on repère les points d'intérêts de l'objet à suivre. Pour rappel, les points d'intérêts d'un objet dans une image représentent les points pour lesquels nous pouvons observer une double discontinuité de l'intensité. Cette double discontinuité correspond à des coins, nous pouvons l'observer sur l'image ci dessus où les points d'intérêts sont marqués d'un petit carré noir. Il existe

plusieurs méthodes permettant de détecter ces points d'intérêt. Une méthode rapide connue et mise en œuvre au cours des travaux dirigés d'imagerie numérique est la méthode de Harris.[24]

- Après avoir repéré ces points d'intérêts sur la première image de la séquence, le but de la méthode est de retrouver ces points sur l'image suivante. Puis, à son tour, cette dernière image servira de base à la recherche de ces mêmes points d'intérêts sur l'image suivante et ainsi de Suite sur toute la séquence vidéo. [24]
  Cette méthode s'appuie sur la minimisation d'une somme de différences carrées en utilisant un modèle de translation. En fonction des dimensions de l'objet à suivre ainsi que des zones texturée des images, une fenêtre de pixels autour de chaque point d'intérêt est déterminée. La taille de cette fenêtre s'étalonne généralement de [3x3] à [25x25]. La minimisation de l'erreur résiduelle entre le point d'intérêt de l'image au temps t et des pixels de la fenêtre au temps t+dt donne le déplacement d'un point d'intérêt. Sur chaque image, les déplacements de tous les points d'intérêts de l'objet sont calculés, donnant ainsi le déplacement général de celui-ci dans la séquence vidéo. [24]

1.5 Conclusion :

Dans ce chapitre nous avons vu des définitions associées au thème (image , pixel, détection d'objet , reconnaissance d'objet , vidéo …), ensuite nous avons présenté quelques méthodes de ce sujet (la méthode viola&jones , Méthode base sur les réseaux de neurones , méthode de la détection de visage par Modèles géométriques , la méthode de détection par la couleur de peau (Skin color) , la méthode la détection de contours …

La liste des méthodes présentées dans ce chapitre n'est pas exhaustive, mais constitue un aperçu de la diversité des méthodes qui existent pour faire la détection d'objet.

Pour notre projet la méthode que choisissons d'implémenter est la méthode de viola-jones pour la détection d'objet, que nous adaptons pour la poursuite d'objets.

Dans le chapitre suivant nous décrivons les différents étapes et algorithmes en détail.

.

2.1 Introduction :

Pour Résoudre le problème de détection et poursuite de visage et d'objets , des algorithmes efficaces et robustes sont exigés. En effet, vu l'importance de la détection de visage pour n'importe que système d'analyse de visage et dans le but d'identifier toutes les régions d'image qui contiennent un visage indépendamment de la position, de l'orientation ou de l'éclairage, plusieurs recherches ont été faites. En particulier, de nombreuses techniques ont été développées pour détecter des visages dans des images fixes et d'autres pour leur poursuite dans des scènes vidéo .Dans ce chapitre on étudie quelques algorithmes de détection et de poursuite de visage et d'objets que nous utilisions dans notre projet

2.2 Algorithme de détection d'objets par viola-Jones :

La méthode de Viola et Jones est une méthode de détection d'objet dans une image numérique, proposée par les chercheurs Paul Viola et Michael Jones. Elle fait partie des premières méthodes capables de détecter efficacement et en temps réel des objets dans une image. Inventée a l'origine pour détection des visages, elle peut également être utilisée pour détecter d'autres types d'objets comme des voitures ou des avions. La méthode de Viola et Jones est l'une des méthodes les plus connues et les plus utilisées, en particulier pour la détection de visages et la détection de personnes.[1]

2.2.1 Principe de la méthode :

La méthode de Viola & Jones consiste à balayer une image à l'aide d'une fenêtre de détection de taille initiale 24px par 24px (dans l'algorithme original) et de déterminer si un visage y est présent. Lorsque l'image a été parcourue entièrement, la taille de la fenêtre est augmentée et le balayage recommence,

jusqu'à ce que la fenêtre fasse la taille de l'image. L'augmentation de la taille de la fenêtre se fait par un facteur multiplicatif de 1,25. Le balayage, quant à lui, consiste simplement à décaler la fenêtre d'un pixel. Ce décalage peut être changé afin d'accélérer le processus, mais un décalage d'un pixel assure une précision maximale.

Cette méthode est une approche basée sur l'apparence, qui consiste à parcourir l'ensemble de l'image en calculant un certain nombre de caractéristiques dans des zones rectangulaires qui se chevauchent. Elle a la particularité d'utiliser des caractéristiques très simples mais très nombreuses.

Il existe d'autres méthodes mais celle de Viola & Jones est la plus performante à l'heure actuelle. Ce qui la différencie des autres est notamment :

- l'utilisation d'images intégrales qui permettent de calculer plus rapidement les caractéristiques
- la sélection par boosting des caractéristiques
- la combinaison en cascade de classifieurs boostés, apportant un net gain de temps d'exécution [17].

2.2.2 Organigramme principal

Les étapes de l'algorithme viola-jones son données (figure 2.1) et décrites dans ce qui suit

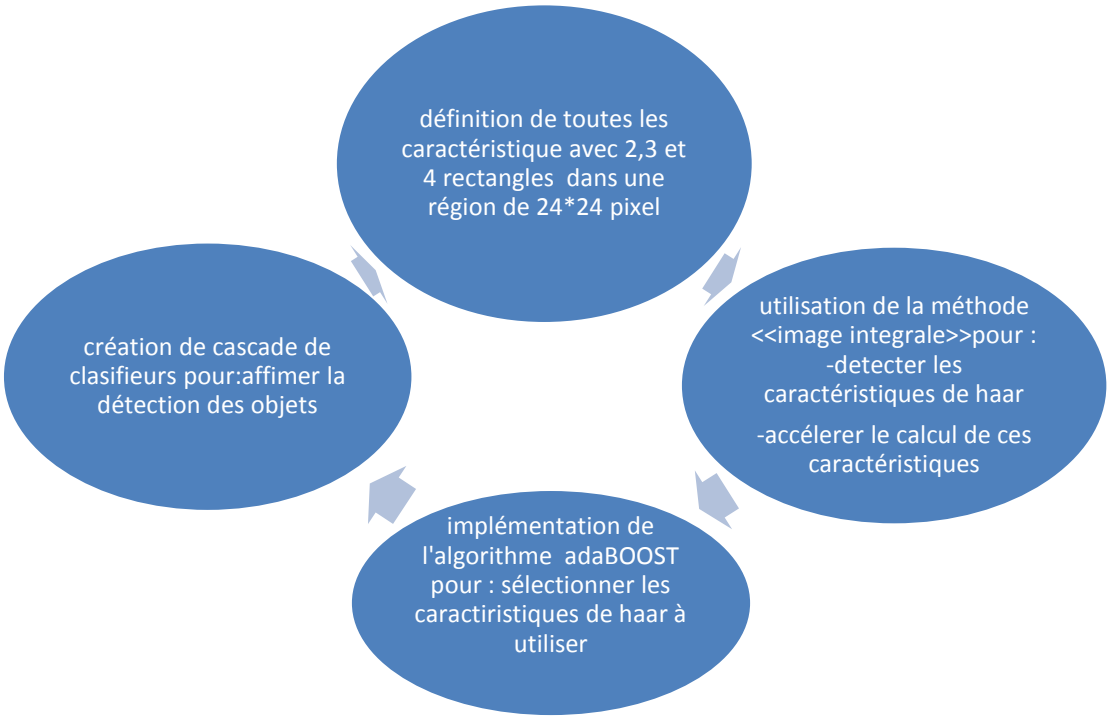


Figure 2.1 : organigramme détection de visage par méthode ≪Viola & Jone≫

2.2.3 Extraction des caractéristiques de type haar [17] :

Une caractéristique est une représentation synthétique et informative, calculée à partir des valeurs des pixels. Les caractéristiques utilisées ici sont les caractéristiques pseudo-haar. Elles sont calculées par la différence des sommes de pixels de deux ou plusieurs zones rectangulaires adjacentes.

Penons un exemple (figure 2.2). Voici deux zones rectangulaires adjacentes, la première en blanc, la deuxième en noire :

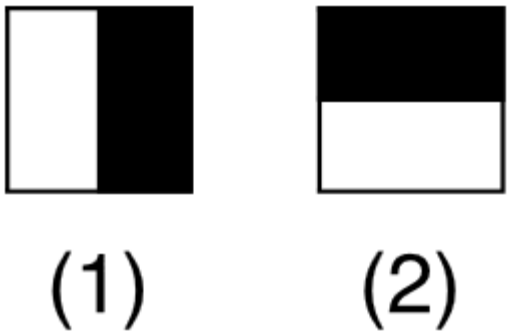


Figure 2.2 **:** deux zones rectangulaires adjacentes, la première en blanc, la deuxième en noire.

Les caractéristiques sont calculées en soustrayant la somme des pixels noirs à la somme des pixels blancs .Elles sont calculées à toutes les positions et à toutes les échelles dans une fenêtre de détection de petite taille, typiquement de 24x24 pixels ou de 20x15 pixels. Un très grand nombre de caractéristiques par fenêtre est ainsi généré, Viola et Jones donnant l'exemple d'une fenêtre de taille 24 x 24 qui génère environ 160 000 caractéristiques.

L'image précédente présente des caractéristiques pseudo-haar à seulement deux caractéristiques mais il en existe d'autres, allant de 4 à 14, et avec différentes orientations.(figure2.3)

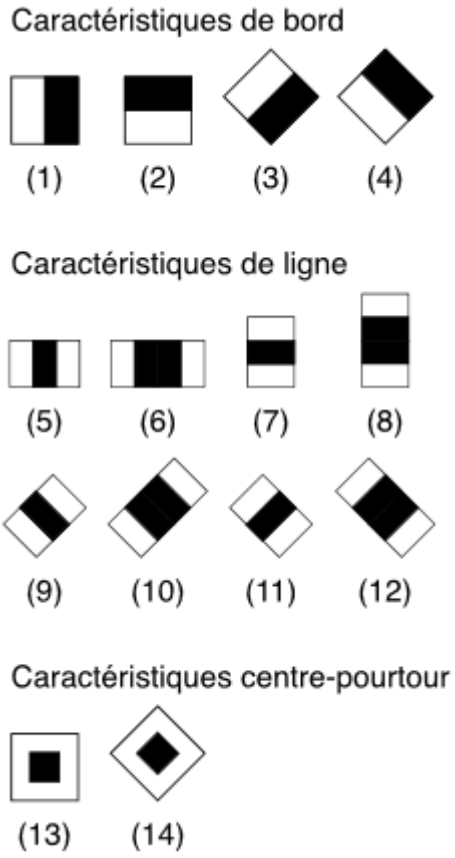


Figure 2.3 : caractéristiques pseudo-haar

à seulement deux caractéristiques, allant de 4 à 14, et avec différentes orientations.

Malheureusement, le calcul de ces caractéristiques de manière "classique" coûte cher en terme de ressources processeur, c'est là qu'interviennent les images intégrales.

2.2.4 Utilisations des images intégrales [1] :

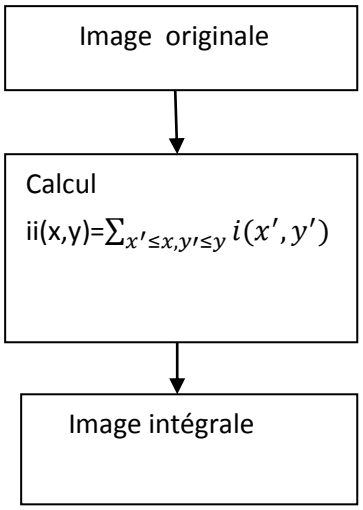


Figure 2.4 : Organigramme de l'image intégrale.

Les caractéristiques de rectangle peuvent être calculées de façon très efficace et rapide en utilisant une représentation intermédiaire de l'image appelée image intégrale. La valeur de l'image intégrale aux coordonnées (x, y) est la somme de tous les pixels au-dessus et a gauche de (x, y), y compris ce dernier point. Figure (2.5)

Soit ii l'image intégrale de l'image initiale i et i(x, y) est la valeur de l'image intégrale au point (x, y): On peut également définir l'image intégrale ii par:

ii(x,y)=$\sum_{x'\leq x,y'\leq y} i(x',y')$…..(2.1)

Ou ii(x, y) est l'image intégrale et i(x, y) est l'image originale**.**

Comme nous utilisons cette nouvelle représentation pour diminuer le temps de calcul, on peut expliquer ses avantages. Tout d'abord, elle peut être calculée par un moyen efficace en utilisant la paire des récurrences suivante:

S(x,y)=S(x,y-1)+i(x,y)…..(2.2)

ii(x,y)=ii(x-1,y)+s(x,y)…..(2.3)

ou s(x, y) s est la somme cumulative.

$\forall x, S(x,-1) = 0$ Et $\forall y, ii(-1,y) = 0$…..(2.4)

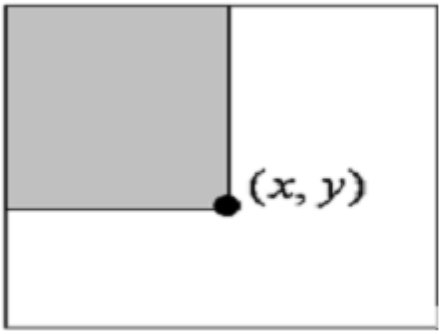


Figure 2.5 : valeur de l'image intégrale au point (x,y)**.**

En utilisant l'image intégrale, toute somme rectangulaire peut être calculée en quatre opérations (voir la figure 2.6). Clairement la différence entre les deux sommes rectangulaires peut être calculée en huit opérations, la caractéristique à deux rectangles définis ci-dessus, implique des sommes rectangulaires adjacentes qui peuvent être calculé en six opérations, huit dans le cas d'une caractéristique à trois rectangles, et neuf pour la caractéristique à quatre rectangles.

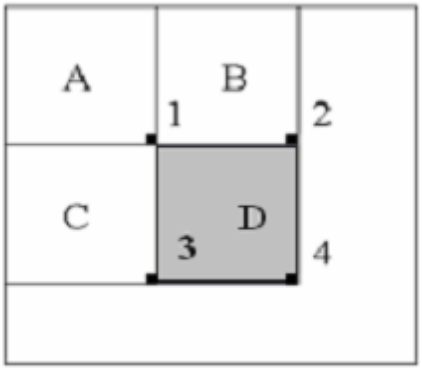


Figure 2.6 : Calcul de la somme du rectangle D avec l'image intégrale.

La somme des pixels dans le rectangle D peut être calculée avec quatre opérations. La valeur de l'image intégrale au point 1 est la somme des pixels dans le rectangle A. La valeur a la position 2 est A + B, a la position 3 est A + C, et a la position 4 est A + B + C + D. la somme dans D peut être calculée comme 4 +1-(2 +3).

Parce que la somme d'intensités dans une région rectangulaire ((x1 ; y1) ; (x2 ; y2)) Est :

$r= II(x_1,y_1)- II(x_1,y_2)- II(x_2,y_1)+ II(x_2,y_2)$……....(2.5)

Ou II est l'image intégrale.

Exemple :

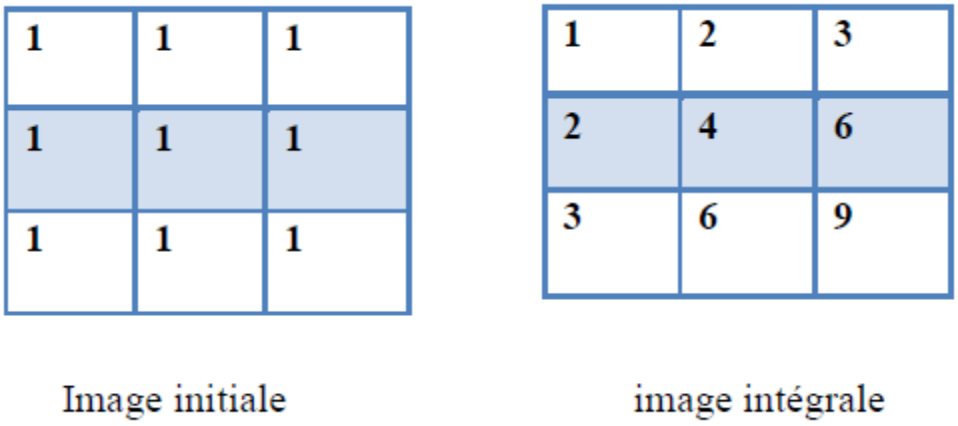


Figure 2.7: image intégrale.

2.2.5 Adaboost [23] :

L'algorithme "AdaBoost" (Adaptive Boosting), développe par Freund et Schapire (Freundet Schapire [1995]), est un algorithme qui crée impérativement une combinaison linéaire de classificateurs de la forme

$H(x) = signe \sum_{t=1}^{T} \alpha t ht(x)$…… (2.6)

Ou ht représente un classificateur faible et αtson poids.

2.2.5.1Algorithme adaboost [23] :

Entrées: Une base d'apprentissage A = {X1;X2,…. XM}

Les étiquettes Y = {y1; y2,…... yM} ou $yi \in \{-1,1\}$

Algorithme de classification C, Nombre d'itérations T

Sorties: Classificateur fort H

$D1 = \frac{1}{M}$ ….. (2.7)/* Initialiser le vecteur de poids des exemples avec

i = 1; 2; ..., M */

Pour t = 1 à T Faire

/* Trouver un classificateur faible h_t qui minimise l'erreur selon D_t*/

ht = Entrainer Classificateur Faible(C; A_t; Dt)

ϵ_t = Erreur (h_t; A) /* Taux d'erreur de h_t */

$\alpha_t = \frac{1}{2} \ln(\frac{1-\epsilon t}{\epsilon t})$ …… (2.8) /* Poids du ht */

$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_{tY_1} h_t(X1))}{Z_t}$ ….(2.9) /* Mettre _a jour le vecteur de poids */

Fin Pour

Retourner $H(x) = signe \sum_{t=1}^{T} \alpha t h t(x)$…….. (2.6)

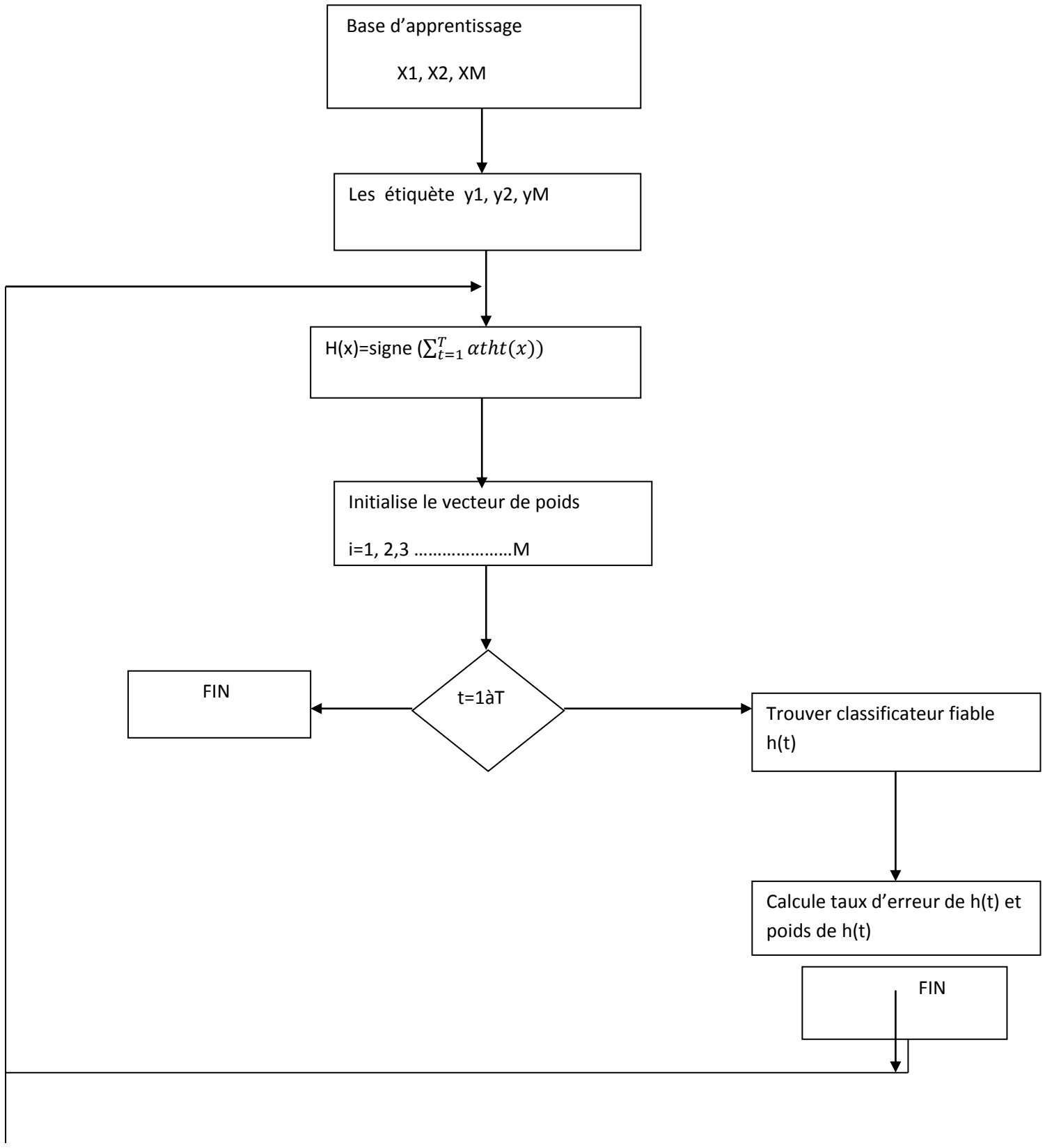


Figure 2.8 : Organigramme d'Adaboost

"AdaBoost" est un algorithme qui prend en entrée, en plus d'un ensemble d'apprentissage, Un second algorithme de classification. A chaque itération, "AdaBoost" appelle le second algorithme afin d'apprendre un nouveau classificateur qui corrige les erreurs de prédiction de la combinaison linéaire courante. Une fois ce nouveau classificateur appris, il est ajouté dans la combinaison linéaire avec un poids associé qui maximise la performance de La nouvelle combinaison, A chaque étape t de l'algorithme, l'apprenant cherche un bon classificateur h ∈ $\{-1,1\}$pour la distribution de probabilité, à priori, sur les exemples d'apprentissage. Cette distribution est représentée grâce à un poids D_t^ipour chaque exemple i qui est mis a jour en fonction de la qualité de la prédiction obtenue a l'étape précédente. L'une des idées principales est de donner aux exemples difficiles à classer un poids D_t^i élevé, (et inversement) de manière a amener l'algorithme à se concentrer sur ces derniers. Chaque classificateur est affecté d'un coefficient proportionnel à l'erreur pondérée. Après T étapes, le classificateur final obtenu est H(x).
La figure 2.9 illustre un exemple d'"AdaBoost" appliqué sur des données en deux dimensions

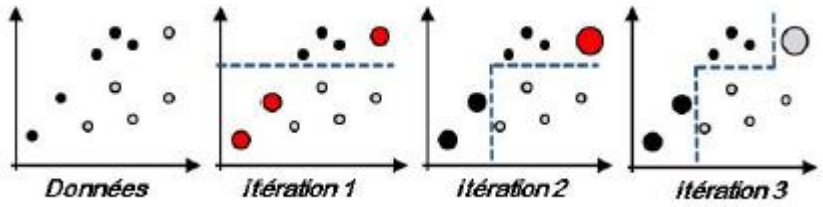


Figure 2.9: Exemple d'AdaBoost[23].

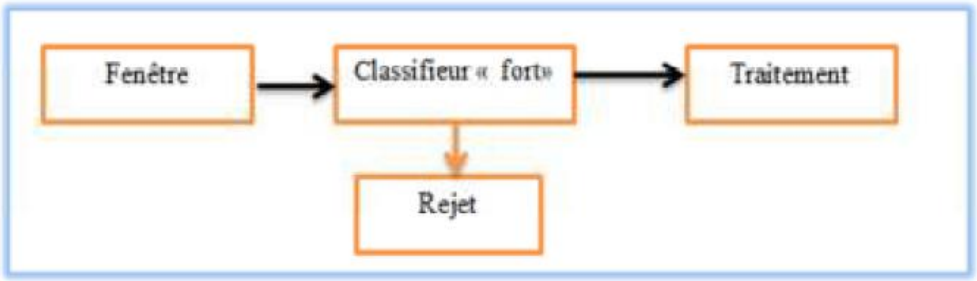


Figure 2.10 : Sélection par boosting.

2.2.6 Cascade de classifieurs :[1]

Une cascade de classifieurs est un arbre de décision dégénère dans laquelle, chaque étape est entrainée pour détecter un maximum d'objets intéressants tout en rejetant une certaine fraction des objets non-intéressants

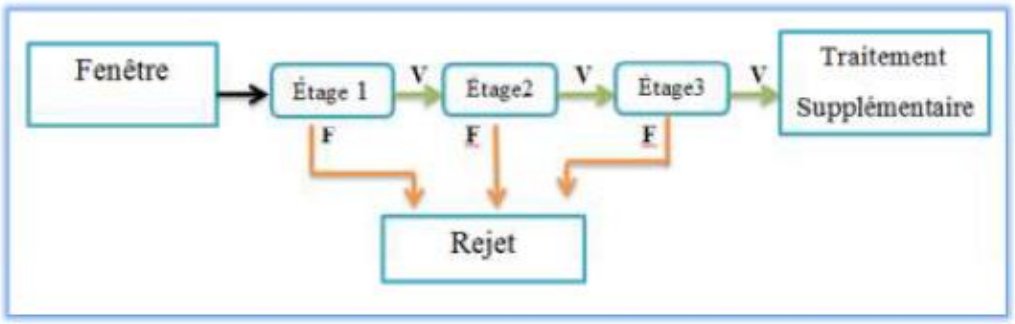


Figure 2.11 : Illustration de l'architecture de la cascade.

La décision ≪ fausse ≫ stoppe le calcul et impose le détecteur de retourner ≪ faux ≫. Une décision ≪ vraie ≫ passe l'image d'entrée au prochain classifie dans la cascade. Si tous les classifier vote vraie alors l'entrée est classifiée comme un vrai exemple. Si n'importe quel classifier vote ≪ faux ≫ alors le calcul se stoppe et l'entrée sera classifiée comme fausse.

En pratique, la cascade est constituée d'une succession d'étages, chacune étant formée d'un classifieur fort appris par AdaBoost. L'apprentissage du classifieur de l'étage *n* est réalise avec les exemples qui ont passe l'étage *n* - 1 ; ce classifieur doit donc faire face a un problème plus difficile : plus on monte dans les étages, plus les classifieur sont complexes.

Pour une cascade, le taux total de faux positifs est le produit du taux de faux positifs de chaque étape :

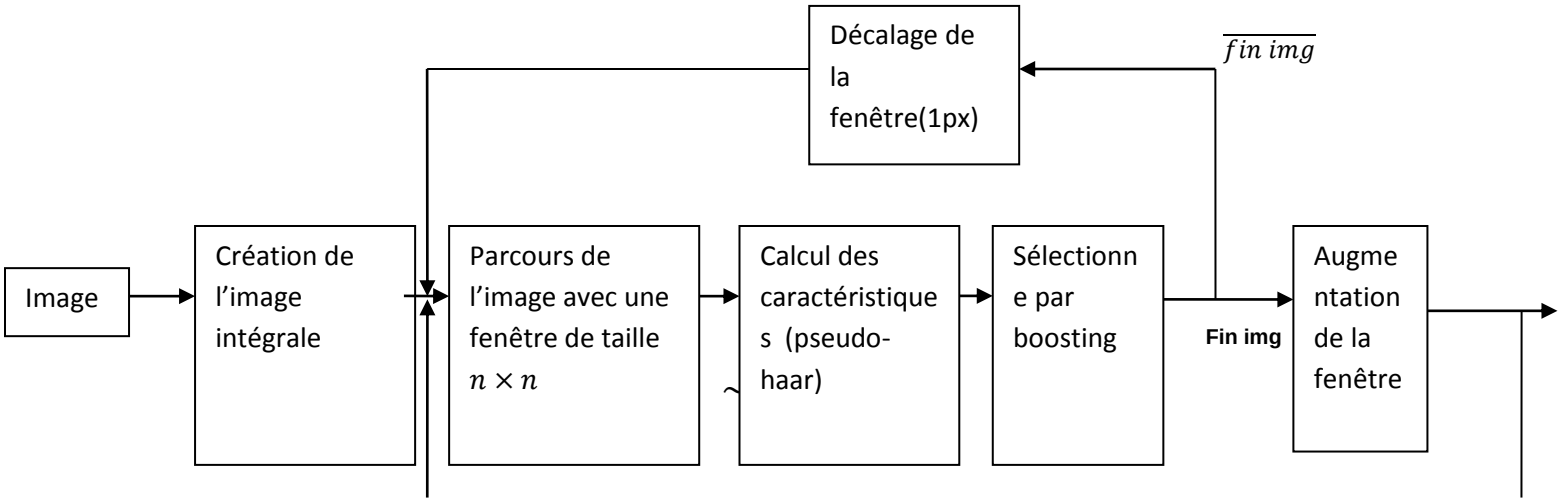
$F = \prod_{i=1}^{K} fi$ ..... (2.10)

K : Nombre d'étages i.

Et de même pour le taux total des détections correctes :

$D = \prod_{i=1}^{K} Di$…… (2.11)

2.2.7 schéma fonctionnel de l'algorithme :[1]



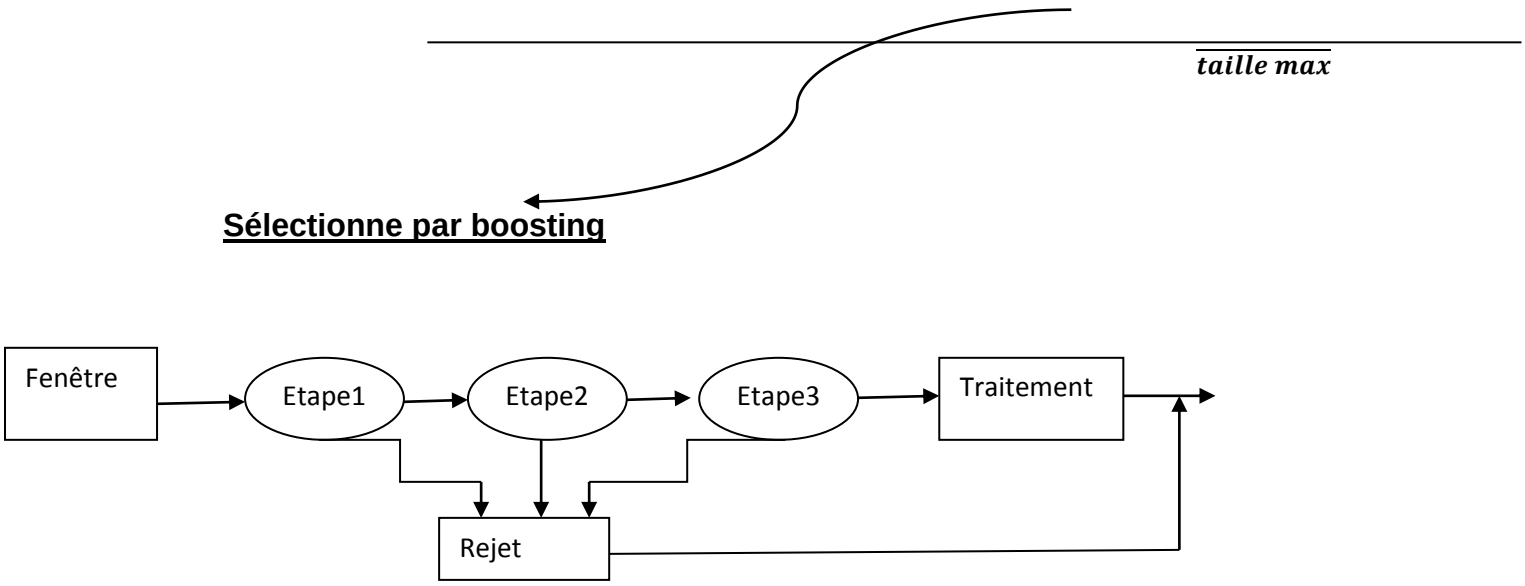


Figure 2.12 : schéma fonctionnel de l'algorithme viola-Jones

Le déroulement global de l'algorithme est le suivant : nous entrons d'abord l'image du système en nivaux de gris. La sélection de fonction Haar est utilisée pour détecter caractéristiques d'un visage .Une image intégrale est calculée pour fixer les traitements, et la fonction Adaboost est ensuite suivie pour bien localiser et former les fonctionnalités, et pour augmenter le temps le temps de traitement. Et enfin la cascade est utilisée pour directement distinguer si une fenêtre contient un visage au non.

2.3algorithme de détection et poursuite de visage KLT (kanade –Lucas –tomais) [18] :

L'algorithme Kanade Lucas Tomasi est utilisé pour suivi des fonctionnalités. C'est l'un des plus populaires. L'algorithme KLT a été introduit par Lucas et Kanade et leur travail a été plus tard étendu par Tomasi et Kanade. Cet algorithme est utilisé pour détecter les points caractéristiques dispersés qui ont assez de texture pour suivre les points requis .

L'algorithme Kanade-Lucas-Tomasi (KLT) est utilisé ici pour suivre les visages humains en continu dans une vidéo Cadre. Cette méthode est accomplie en trouvant les paramètres qui permettent de réduire la dissimilarité Mesurée entre des points caractéristiques liés à modèle de translation original.

Premièrement, dans cet algorithme, nous calculons le déplacement des points suivis d'une image à un autre cadre. À partir de ce calcul de déplacement, il est facile de calculer le mouvement de la tête. La fonctionnalité Les points d'un visage humain sont suivis à l'aide d'un flux optique. L'algorithme de suivi KLT suit le visage en deux étapes simples, il trouve d'abord les points caractéristiques tractables dans la première image, puis suit les caractéristiques détectées dans les images suivantes en utilisant le déplacement calculé.

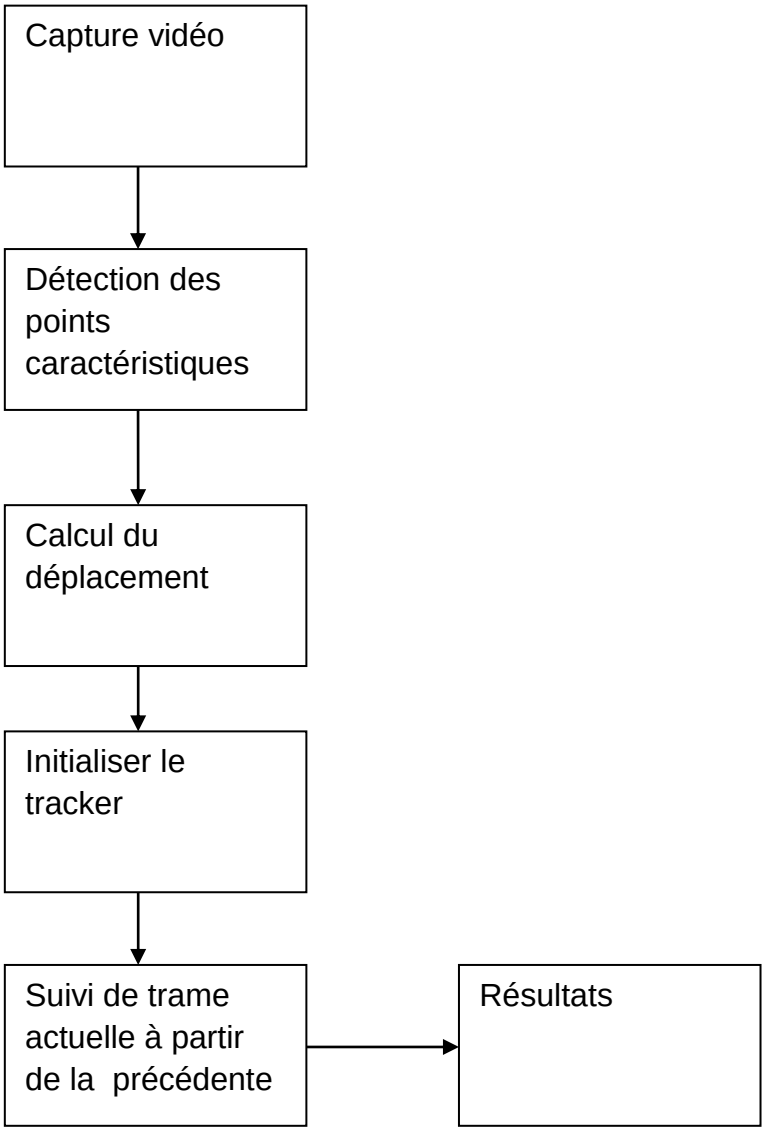


Figure 2.13 : organigramme de l'algorithme KLT.

Dans cet algorithme, il détecte d'abord les coins Harris dans

la première image. Et puis en utilisant le flux optique, il continue à détecter les points en calculant le mouvement des pixels à une autre image. Pour chaque mouvement de translation de l'image optique le flux est calculé. Les

coins Harris sont détectés en reliant les vecteurs de mouvement dans des images successives pour obtenir une piste pour chaque point Harris. Juste pour ne pas perdre la trace de la séquence vidéo, nous appliquons le détecteur harris tous les 10 à 15 cadres. Ce n'est rien d'autre que de s'assurer en vérifiant les cadres périodiquement. De cette façon, nouveaux et anciens points Harris sont suivis. nous considérons uniquement 2-D mouvement, c'est-à-dire mouvement de translation.

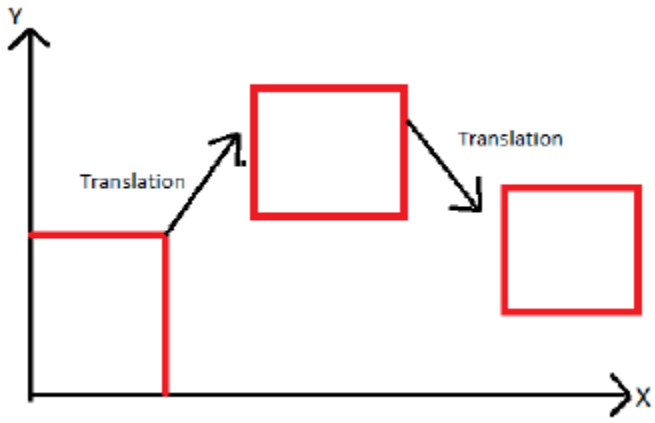


Figure 2.14: Translation mouvement de cadre.

2.4 Algorithme de détection et poursuite d'objets selon la couleur [19] :

Cette méthode est basée sur la couleur d'objets détectés.

On effectue la Lecture pixel par pixel de la valeur de celui-ci en rouge vert et bleu. Puis on test le pixel, Suivant la couleur à reconnaitre le rouge, le vert et le bleu doivent être supérieur ou inférieur au seuil imposé. S'ils respectent ces conditions alors on incrémente.

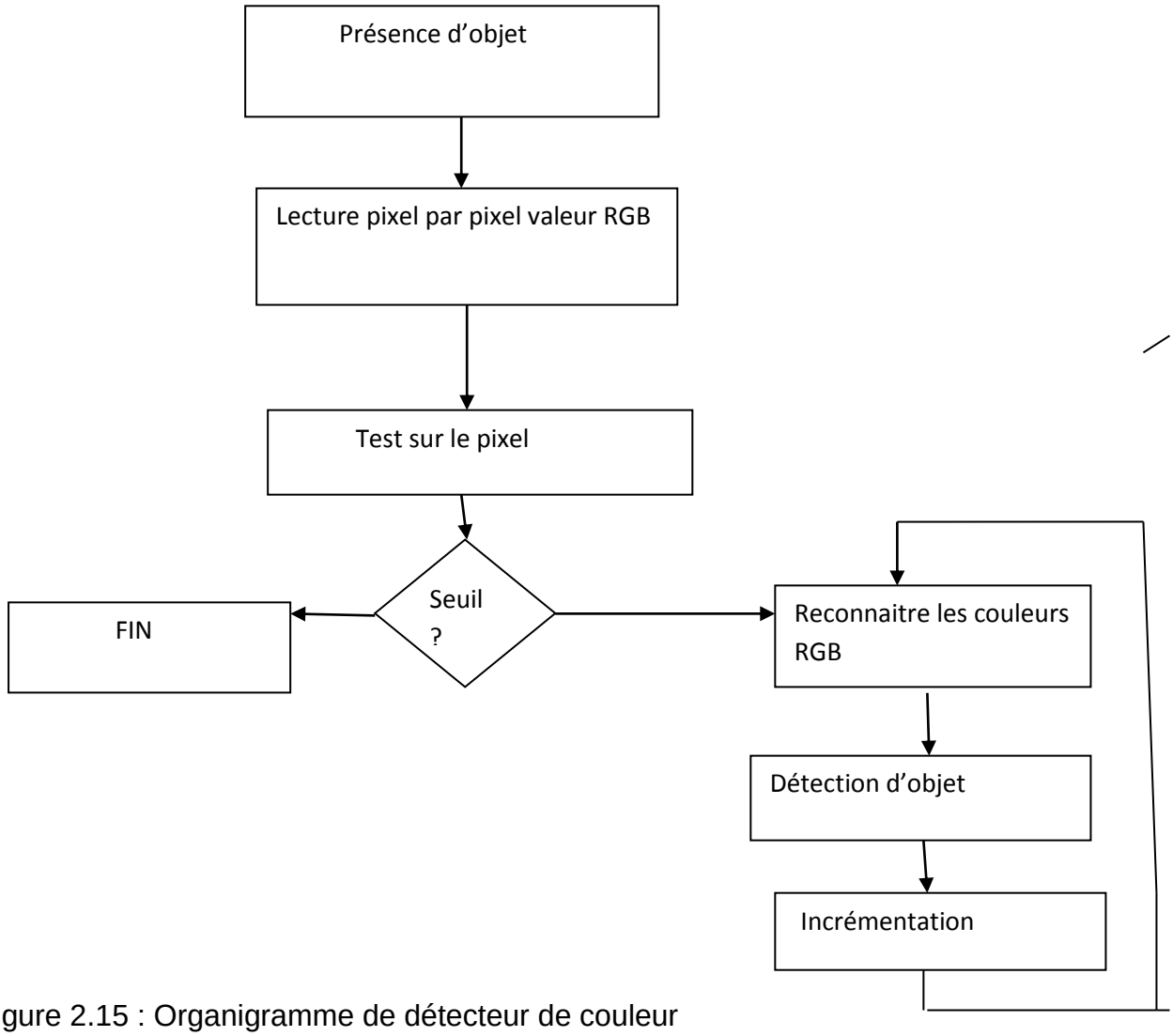


Figure 2.15 : Organigramme de détecteur de couleur

2.4.1Paramètres

Les paramètres principaux sont les différents seuils des couleurs (RGB).

2.4.2Avantage

C'est une méthode assez simple à mettre en œuvre, il suffit de reconnaitre une couleur qui ressort par rapport au reste de l'image

2.4.3Inconvénients

Si notre objet change de couleur au cours de la vidéo, la couleur à reconnaitre devra également être modifiée.

Il existe beaucoup de nuance de couleur (256^3) il faut alors avoir une grande base de données puis choisir la couleur ou définir a chaque fois la couleur souhaitée ce qui sous entend que ces valeurs soit connues.

2.5Conclusion :

Dans ce chapitre, nous avons étudié des algorithmes de détection et poursuite d'objets nous avons détaillé l'algorithme De Voila et Jones, ensuite nous avons présenté le fonctionnement et le principe de cette méthode, nous avons également étudié la méthode de KLT pour la poursuite de visage et d'objets et la méthode de poursuite d'objets selon la couleur

Dans le chapitre suivant nous montrons l'implémentation des ces méthodes et les résultats obtenus.

3.1 Introduction :

L'objectif de notre chapitre est de présenter l'évaluation pour nos méthodes implémentées de détection des objets en mouvement selon leur couleur et détection de visage et les résultats obtenu dans une scène vidéo capturée par une webcam et les vidéos enregistrée en utilisant le logiciel matlab

3.2 Enivrement de développement :

Dans cette section, nous présentons les environnements matériel et logiciel de notre travail.

3.2.1 Environnement matériel :

Pour effectuer notre projet , nous avons utilisé un PC (Samsung MSI-PC) ayant les caractéristiques aperçus dans le tableau suivant :

Processeur	Intel(R) Core(TM) i3-3110M CPU 2.40GHz
Mémoire installe (RAM)	4,00 Go
Type de Système d'exploitation	Windows7 64 bit
Capacité du disque dur	270 Go

3.2.2 Environnement logiciel :

3.2.2.1 Le langage MATLAB :

MATLAB est à la fois un langage de programmation et un environnement de développement développé et commercialisé par la société américaine MathWorks.
MATLAB est utilisé dans les domaines de l'éducation, de la recherche et de l'industrie pour le
calcul numérique mais aussi dans les phases de développement de projets.
MATLAB = MATrix LABoratory est largement utilisé dans l'industrie de l'ingénierie et des
milieux universitaires, en particulier au secteur de l'aérospatiale

MATLAB c'est un langage de haut niveau utilisant la structure de données matrice comme base
et possédant :
- Les structures de contrôle de tous les langages de haut niveau
- Les possibilités d'écrire des fonctions
- Les entrées sorties habituelles
- La programmation Orientée Objet est également possible.
Ce langage permet aussi bien de développer des petites applications de façon très rapide que de
complexes programmes d'application.

LE MATLAB :
-Permet de réaliser des simulations numériques basées sur des algorithmes d'analyse numérique, Peut donc être utilisé pour la résolution approchée d'équations différentielles, d'équations aux
dérivées partielles ou de systèmes linéaires,
-Permet la manipulation de matrice,
Afficher des courbes et des données,
- Mettre en oeuvre des algorithmes,
Créer des interfaces utilisateurs,
Peut s'interfacer avec d'autres langages comme le C, C++, Java, et Fortran,
Peut-être aussi considéré comme un langage de programmation adapté pour les problèmes
scientifiques,
Fonctionne dans plusieurs environnements tels que Windows, Unix, Macintosh.
Il existe deux modes de fonctionnement:
- Mode interactif (en ligne) : MATLAB exécute les instructions au fur et à mesure qu'elles
sont données par l'usager.
- Mode exécutif (programmation) : MATLAB exécute ligne par ligne un "fichier .m"
(programme en langage MATLAB).

Le mode en ligne permet d'obtenir des résultats simples qui ne sont pas sauvegardés.
Le mode programmation, quant à lui, permet de développer des applications très complexes.

Nous avons utilisé vision toolbox de matlab Computer Vision Toolbox fournit des algorithmes, des fonctions et des applications pour concevoir et tester des systèmes de vision par ordinateur, de vision 3D et de traitement vidéo. Vous pouvez effectuer la détection et le suivi d'objets, ainsi que la détection, l'extraction et la mise en correspondance des caractéristiques.

- vision.CascadeObjectDetector:Le détecteur d'objet en cascade utilise l'algorithme Viola-Jones pour détecter les visages, le nez, les yeux, la bouche ou le haut du corps des personnes.

3.3 implémentation du programme de détection viola-jones pour la détection de visage :

```
close all
clear all
clc
FDetect = vision.CascadeObjectDetector;
% pour détecter les visages, le nez, la bouche ou le haut du corps des personnes
% pour détecter le visageFDetect = vision.CascadeObjectDetector;
%Read the input image
I=imread('unnamed.jpg');
figure(1),imshow(I)
title('originale photo');
BB =step(FDetect,I);
figure(2),
imshow(I);hold on
for i=1:size(BB,1)
    rectangle('Position',BB(i,:),'Linewidth',6,'Linestyle','-','Edgecolor','b');
end
title('face detection');
```

Figure 3.1 : programme de détection de visage par viola-jone

3.3.1 implémentation du programme de détection viola-jones pour la détection de nez :

```
% Mergethresold
% pour éviter la détection multiple autour d'un objet, le 'Seuil de fusion'%La valeur% peut être remplacée
% pour détecter le nez
NoseDetect = vision.CascadeObjectDetector('Nose','MergeThreshold',20);
BB=step(NoseDetect,I);
figure(3),
imshow(I);hold on
for i=1:size(BB,1)
    rectangle('Position',BB(i,:),'Linewidth',3,'Linestyle','-','Edgecolor','b');
end
title('Nose detection');
hold off;
```

Figure 3.2 : programme de détection de nez par viola-jones

3.3.2 Implémentation du programme de détection viola-jones pour la détection de bouche :

```
MouthDetect = vision.CascadeObjectDetector('Mouth','MergeThreshold',60);
BB=step(MouthDetect,I);
figure(4),
imshow(I);hold on
for i=1:size(BB,1)
    rectangle('Position',BB(i,:),'Linewidth',3,'Linestyle','-','Edgecolor','r');
end
title('Mouth detection');
hold off;
```

Figure3.3 : programme de détection de bouche par viola-jones

3.3.3 implémentation du programme de détection viola-jones pour la détection des yeux :

```
EyeDetect = vision.CascadeObjectDetector('EyePairBig');
BB=step(EyeDetect,I);
figure,imshow(I)
figure(5),
imshow(I);hold on
for i=1:size(BB,1)
    rectangle('Position',BB(i,:),'Linewidth',3,'Linestyle','-','Edgecolor','r');
end
title('Eye detection');
hold off;
```

Figure 3.4 : programme de détection des yeux par viola-jones

.

Nous avons créé le détecteur d'objets en cascade pour détecter le visage, le nez, les yeux, la bouche ou le haut du corps, puis lire les images entrantes. Lorsque le visage est détecté, un rectangle de détection et visualisation sont créés, puis nous avons redimensionné l'image pour appliquer le «recadrage» après avoir appliqué le recadrage et afficher l'image redimensionnée.

3.4 Implémentation du programme de détection et poursuite de visage par viola-jones et KLT algorithme (kanade –lucas –tomasi) :

```
% Créez un objet détecteur en cascade.
faceDetector = vision.CascadeObjectDetector();
%Lisez une image vidéo et exécutez le détecteur de visage.
videoFileReader = vision.VideoFileReader('200902-141334.mpg');
videoFrame      = step(videoFileReader);
bbox            = step(faceDetector, videoFrame);
% Convertissez la première boîte en polygone.
%Ceci est nécessaire pour pouvoir visualiser la rotation de l'objet.
x = bbox(1, 1); y = bbox(1, 2); w = bbox(1, 3); h = bbox(1, 4);
bboxPolygon = [x, y, x+w, y, x+w, y+h, x, y+h];
%Dessinez le cadre de délimitation renvoyé autour du visage détecté.
videoFrame = insertShape(videoFrame, 'Polygon', bboxPolygon);

figure; imshow(videoFrame); title('Detected face');
%Détectez les points caractéristiques dans la région du visage.
points = detectMinEigenFeatures(rgb2gray(videoFrame), 'ROI', bbox);
% Affichez les points détectés.
figure, imshow(videoFrame), hold on, title('Detected features');
plot(points);
%Créez un suivi de points et activez la contrainte d'erreur bidirectionnelle pour
%le rendre plus robuste en présence de bruit et d'encombrement.
pointTracker = vision.PointTracker('MaxBidirectionalError', 2);
%Initialisez le tracker avec les emplacements des points initiaux et le
% image vidéo.
points = points.Location;
initialize(pointTracker, points, videoFrame);
videoPlayer  = vision.VideoPlayer('Position',...
    [100 100 [size(videoFrame, 2), size(videoFrame, 1)]+30]);
```

Figure3.5: programme de détection et poursuite de visage par klt algorithme et viola-jones

Nous avons utilisé l'algorithme de détection violajones avec l'algorithme de poursuite KLT pour détecter et suivi le visage .

Pour détecter le visage on utilise l'objets système vision.cascadeobjectdetector pour détecter l'emplacement d'un visage dans une image vidéo .pour suivre le visage on utilise l'algorithme KLT ce programme ne détecte le visage qu'une seul fois puis l'algorithme KLT suit le visage sur les images vidéo .

KLT suit un ensemble de points caractéristique sur les image vidéo une fois que la détection a localisé le visage l'étape suivant de programme identifier les point caractéristique .ensuite on utilise l'objets système vision.tracker pour le suivre pour chaque point de l'image précédent le suivi des point tente trouver le point correspondent dans l'image actuelle .ensuite la fonction géométrique transforme est utilisé pour estimer la translation , retation et l'échelle entre les anciens points et les nouveaux point. Aprés. on a crée un objets lecteur vidéo pour afficher les images vidéo, enfin racke les point

d’une image à l’autre et utilisé la fonction estimation geometric transform pour estimer les mouvement de visage

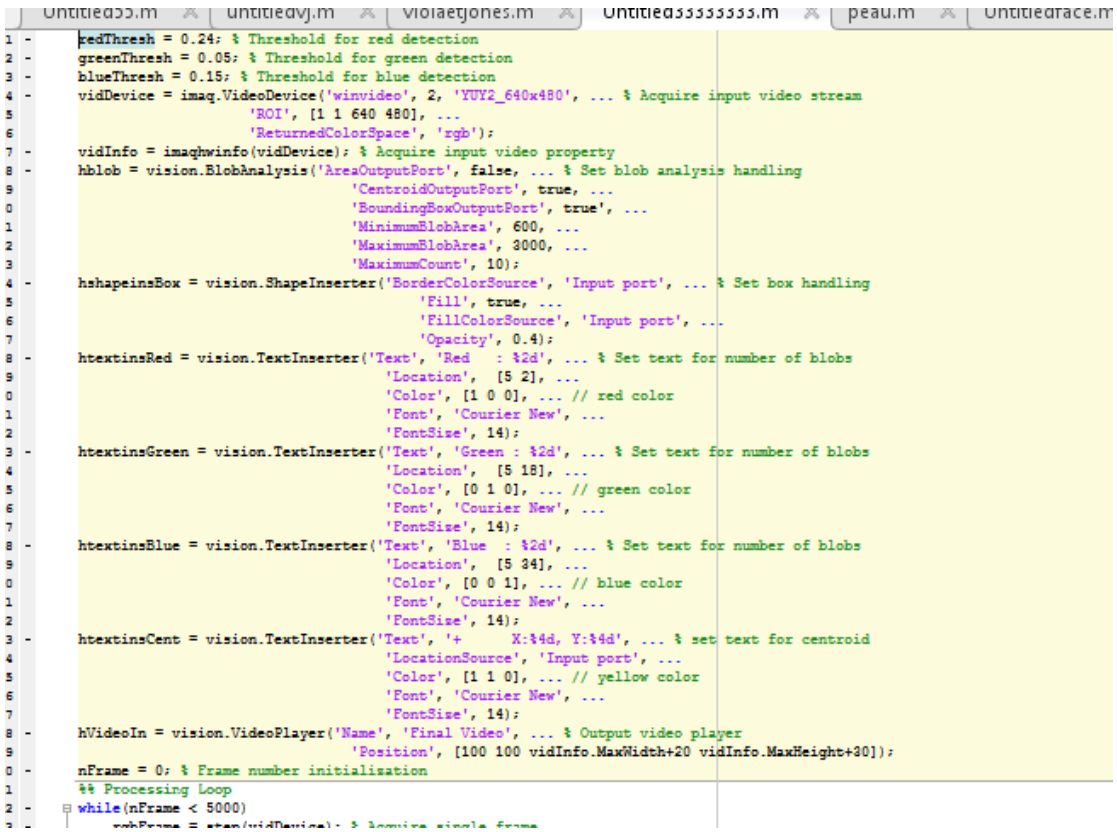
3.5 Implémentation de programme de détection et poursuite d’objets on blanc :

```
thresh = 0.75; % la seuil de détection de couleur blanc
vidDevice = imaq.VideoDevice('winvideo', 2, 'YUY2_640x480', ... % Acquérir un flux vidéo d'entrée
                    'ROI', [1 1 640 480], ...
                    'ReturnedColorSpace', 'rgb');
vidInfo = imaqhwinfo(vidDevice); %Acquérir la propriété vidéo d'entrée
hblob = vision.BlobAnalysis('AreaOutputPort', false, ... % Définir la gestion de l'analyse des objets blob
                                'CentroidOutputPort', true, ...
                                'BoundingBoxOutputPort', true', ...
                                'MinimumBlobArea', 400, ...
                                'MaximumCount', 50);
hshapeinsWhiteBox = vision.ShapeInserter('BorderColor', 'Custom', ...
                                         'CustomBorderColor', [1 0 0]); %Définir la gestion des boîtes blanches
htextins = vision.TextInserter('Text', 'number of white object (s): %2d', ... %Définir le texte pour le nombre d'objets blob
                                    'Location',  [7 2], ...
                                    'Color', [1 1 1], ... // couleur blanche
                                    'Font', 'Courier New', ...
                                    'FontSize', 12);
htextinsCent = vision.TextInserter('Text', '+      X:%6.2f,  Y:%6.2f', ... % définir le texte pour le centre de gravité
                                    'LocationSource', 'Input port', ...
                                    'Color', [0 0 0], ... // couleur noir
                                    'FontSize', 12);
hVideoIn = vision.VideoPlayer('Name', 'Final Video', ... % Lecteur vidéo de sortie
                                'Position', [100 100 vidInfo.MaxWidth+20 vidInfo.MaxHeight+30]);
nFrame = 0; % Initialisation du numéro de trame
%%Boucle de traitement
while(nFrame < 3000)
    rgbFrame = step(vidDevice); % Acquérir une seule trame
```

Figure3.6 : programme de détection et poursuite d’objets on blanc

Un algorithme de détection de couleur identifie les pixels d'une image qui correspondent à une couleur ou à une gamme de couleurs spécifiée. La couleur des pixels détectés peut alors être modifiée pour les distinguer du reste de l'image.

3.6 Implémentation de programme de détection et poursuite d’objets selon la couleur rgb:



```
redThresh = 0.24; % Threshold for red detection
greenThresh = 0.05; % Threshold for green detection
blueThresh = 0.15; % Threshold for blue detection
vidDevice = imaq.VideoDevice('winvideo', 2, 'YUY2_640x480', ... % Acquire input video stream
                    'ROI', [1 1 640 480], ...
                    'ReturnedColorSpace', 'rgb');
vidInfo = imaqhwinfo(vidDevice); % Acquire input video property
hblob = vision.BlobAnalysis('AreaOutputPort', false, ... % Set blob analysis handling
                                'CentroidOutputPort', true, ...
                                'BoundingBoxOutputPort', true', ...
                                'MinimumBlobArea', 600, ...
                                'MaximumBlobArea', 3000, ...
                                'MaximumCount', 10);
hshapeinsBox = vision.ShapeInserter('BorderColorSource', 'Input port', ... % Set box handling
                                        'Fill', true, ...
                                        'FillColorSource', 'Input port', ...
                                        'Opacity', 0.4);
htextinsRed = vision.TextInserter('Text', 'Red   : %2d', ... % Set text for number of blobs
                                    'Location',  [5 2], ...
                                    'Color', [1 0 0], ... // red color
                                    'Font', 'Courier New', ...
                                    'FontSize', 14);
htextinsGreen = vision.TextInserter('Text', 'Green : %2d', ... % Set text for number of blobs
                                    'Location',  [5 18], ...
                                    'Color', [0 1 0], ... // green color
                                    'Font', 'Courier New', ...
                                    'FontSize', 14);
htextinsBlue = vision.TextInserter('Text', 'Blue  : %2d', ... % Set text for number of blobs
                                    'Location',  [5 34], ...
                                    'Color', [0 0 1], ... // blue color
                                    'Font', 'Courier New', ...
                                    'FontSize', 14);
htextinsCent = vision.TextInserter('Text', '+      X:%4d, Y:%4d', ... % set text for centroid
                                    'LocationSource', 'Input port', ...
                                    'Color', [1 1 0], ... // yellow color
                                    'Font', 'Courier New', ...
                                    'FontSize', 14);
hVideoIn = vision.VideoPlayer('Name', 'Final Video', ... % Output video player
                                    'Position', [100 100 vidInfo.MaxWidth+20 vidInfo.MaxHeight+30]);
nFrame = 0; % Frame number initialization
%% Processing Loop
while(nFrame < 5000)
```

Figure3.7 :programme de détection et poursuite d'objets selon les couleur rgb

cette méthode est basée sur la couleur d'objets détectés.

Ce code peut détecter et suivre les objets de couleur rouge, vert et bleu dans la vidéo LIVE. Dans ce code, les objets peuvent être segmentés à l'aide de la détection et de la segmentation des couleurs.

3.7 implémentation de programme de détection et poursuite visage selon la couleur de peau:

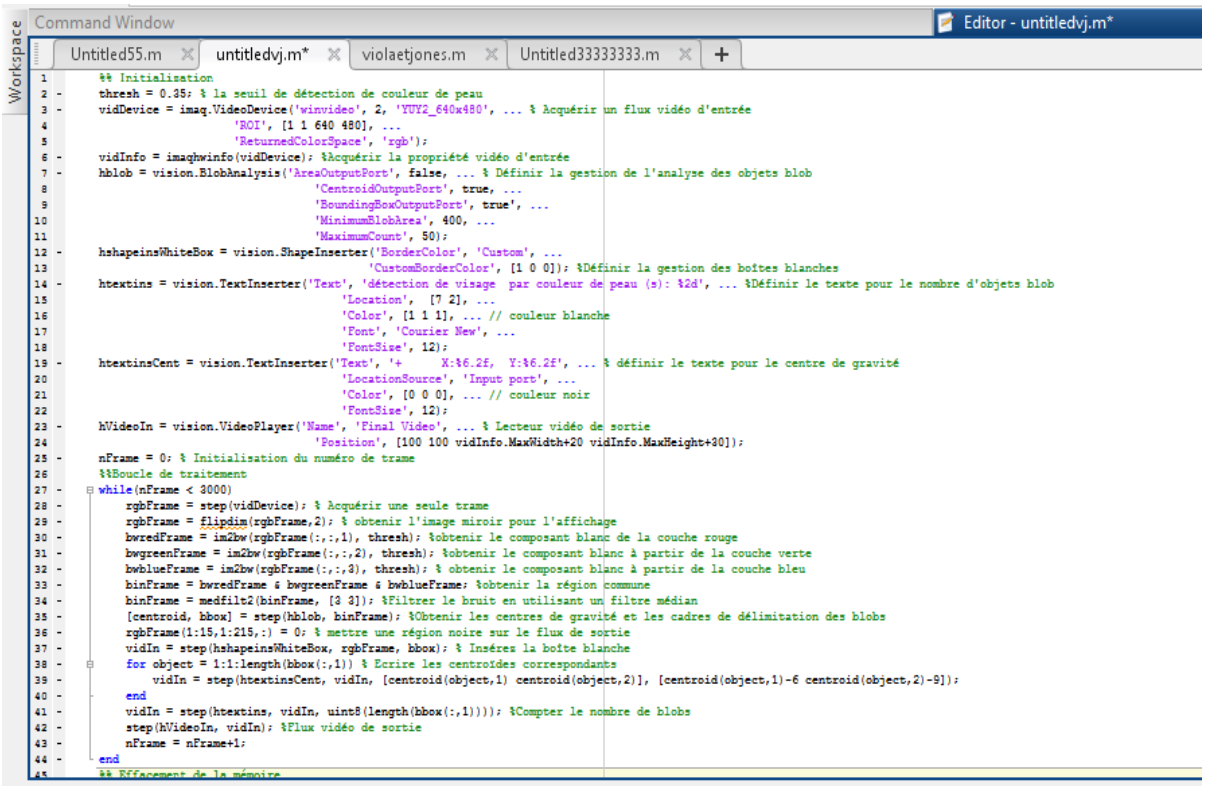


Figure3.8: programme de détection et poursuite d'objets selon la couleur de peau

Pour les visages la couleur de la peau de l'être humain est une caractéristique spécifique c'est pour ça elle a été utilisée pour la détection des visages. En effet la couleur de la peau est différente selon les personnes et leur origine (Africain, Européen, Asiatique...). la seuil de couleur de peau entre (0.35 et 0.45).

3.8 Résultat :

3.8.1 Résultat de la détection de visage par viola-jones :

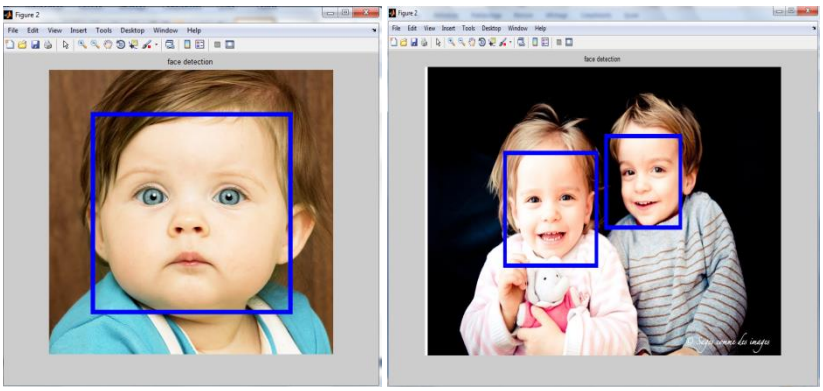


Figure3.9 : résultat de détection de visage par violajones

3.8.1.1 Résultat de la détection de nez par viola-jones :

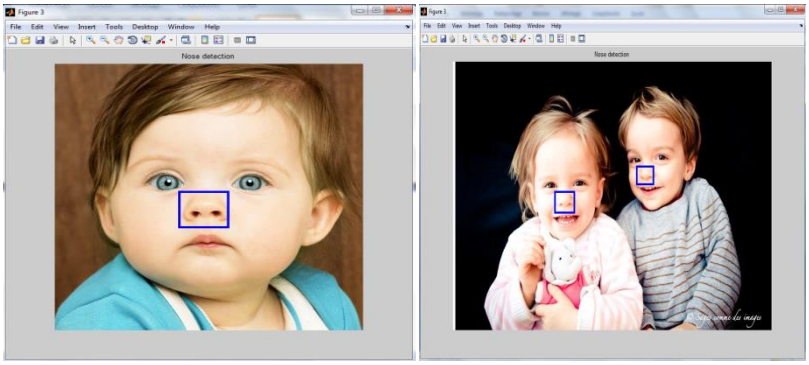


Figure3.10 : résultat de détection de nez par violajones

3.8.1.2 Résultat de la détection de bouche par viola-jones :

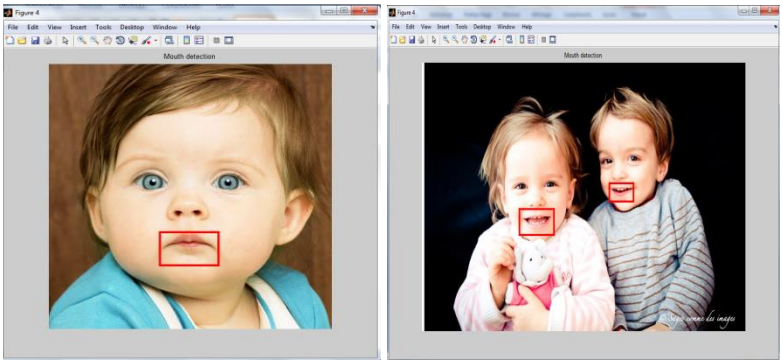


Figure3.11 : résultat de détection de bouche par violajones

3.8.1.3 Résultat de la détection des yeux par viola-jones :

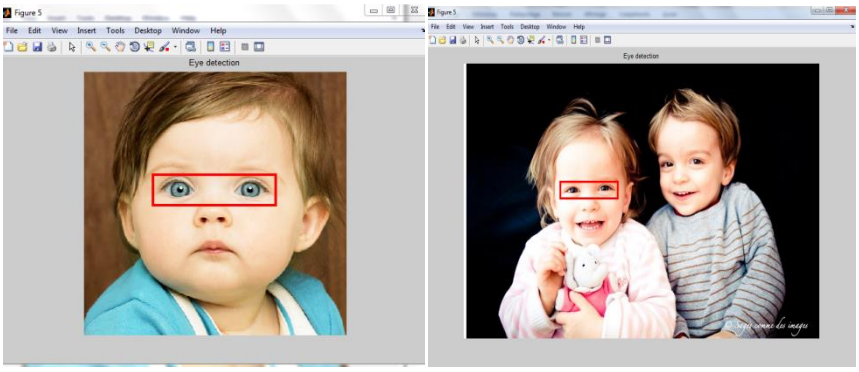
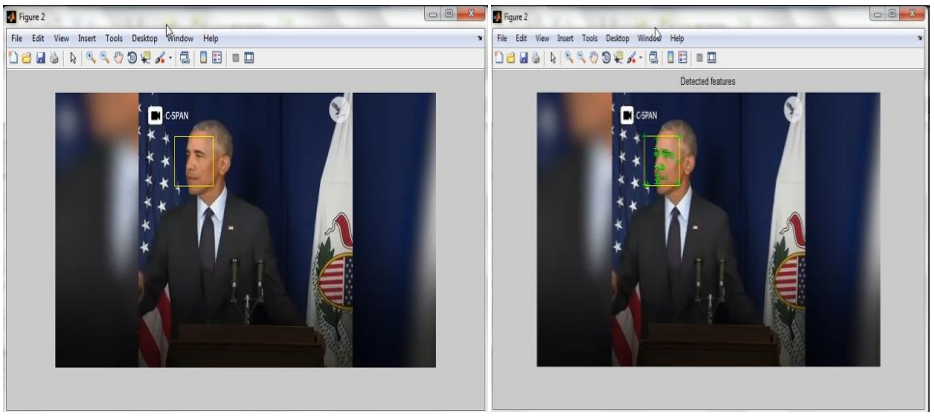


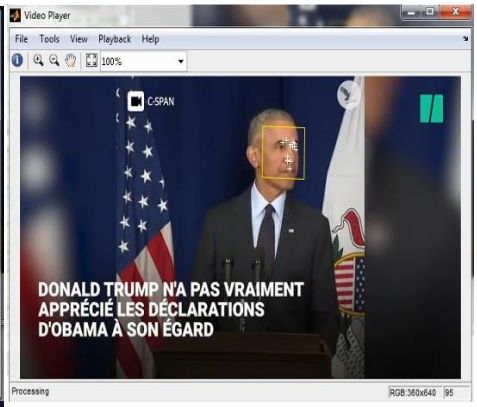
Figure3.12 : résultat de détection des yeux par violajones

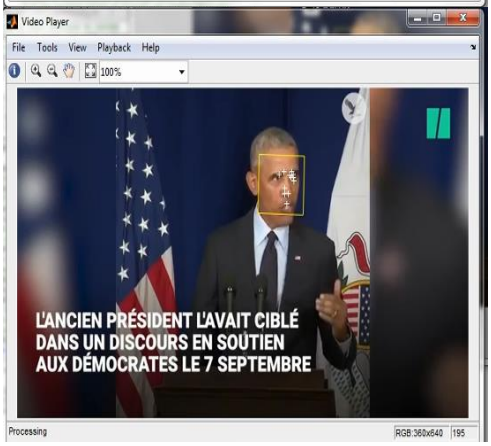
3.8.2 Résultat de la détection et poursuite de visage par viola-jones et KLT algorithme :

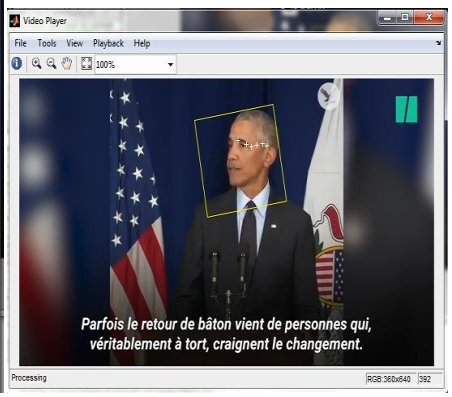
Dans cette étude nous utilisons l'algorithme de KLT. La séquence d'images suivante nous montre comment le visage est localisé sur chaque image




Video Player
File Tools View Playback Help
100%
C-SPAN
DONALD TRUMP N'A PAS VRAIMENT
APPRÉCIÉ LES DÉCLARATIONS
D'OBAMA À SON ÉGARD
Processing
RGB:360x640 55


Video Player
File Tools View Playback Help
100%
C-SPAN
DONALD TRUMP N'A PAS VRAIMENT
APPRÉCIÉ LES DÉCLARATIONS
D'OBAMA À SON ÉGARD
Processing
RGB:360x640 95


Video Player
File Tools View Playback Help
100%
L'ANCIEN PRÉSIDENT L'AVAIT CIBLÉ
DANS UN DISCOURS EN SOUTIEN
AUX DÉMOCRATES LE 7 SEPTEMBRE
Processing
RGB:360x640 195


Video Player
File Tools View Playback Help
100%
Parfois le retour de bâton vient de personnes qui,
véritablement à tort, craignent le changement.
Processing
RGB:360x640 392


Video Player
File Tools View Playback Help
100%
Processing
RGB:360x640 537


Video Player
File Tools View Playback Help
100%
par les puissants,
Processing
RGB:360x640 672


Video Player
File Tools View Playback Help
100%
et les privilégiés.
Processing
RGB:360x640 728

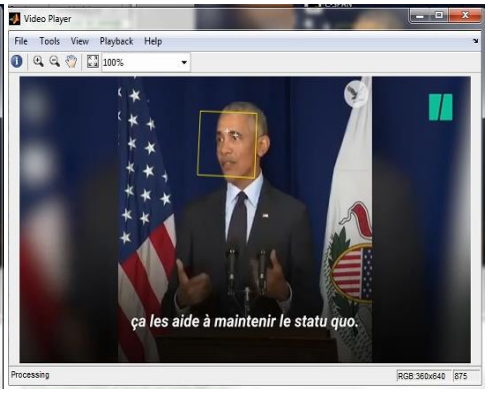
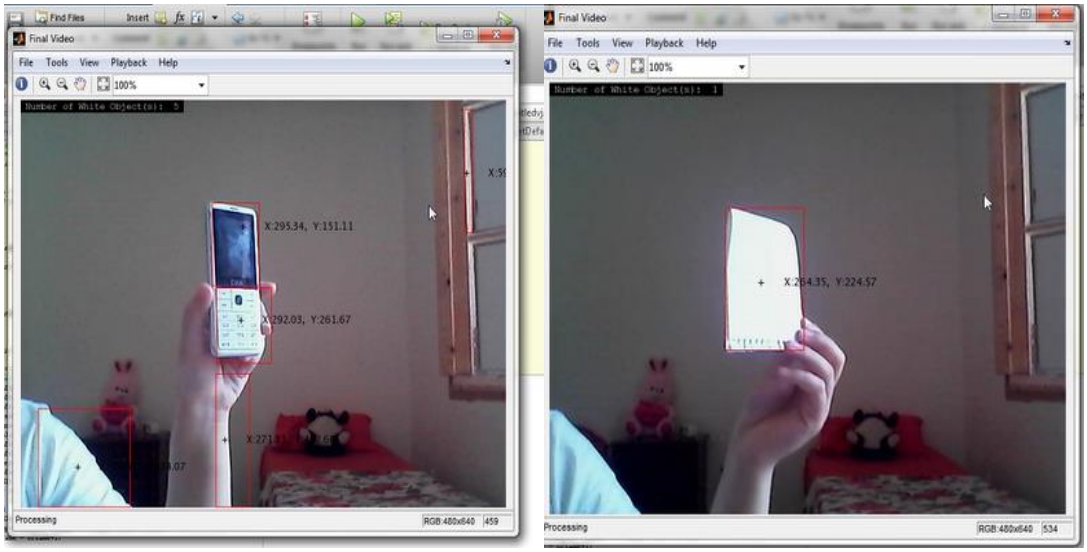

Video Player
File Tools View Playback Help
100%
ça les aide à maintenir le statu quo.
Processing
RGB:360x640 875

Figure3.13 : résultat de détection et poursuite de visage par violajones et klt algorithme

3.8.3 Résultat de la détection et poursuite d'objets selon la couleur blanc :

Nous avons dans cette image (**figure**) un objet de couleur blanc , nous sommes intéressé par la détection de l'objets qui une couleur blanc (avec rvb = **(255, 255, 255))**



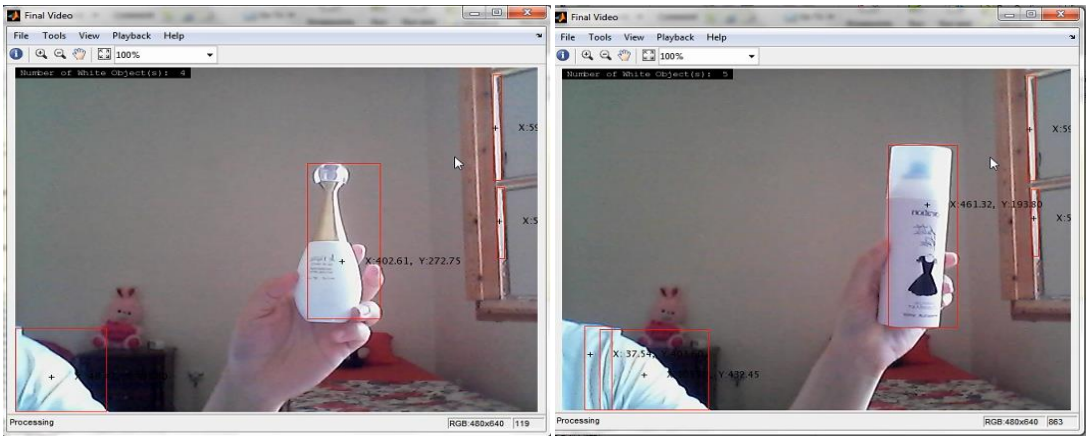
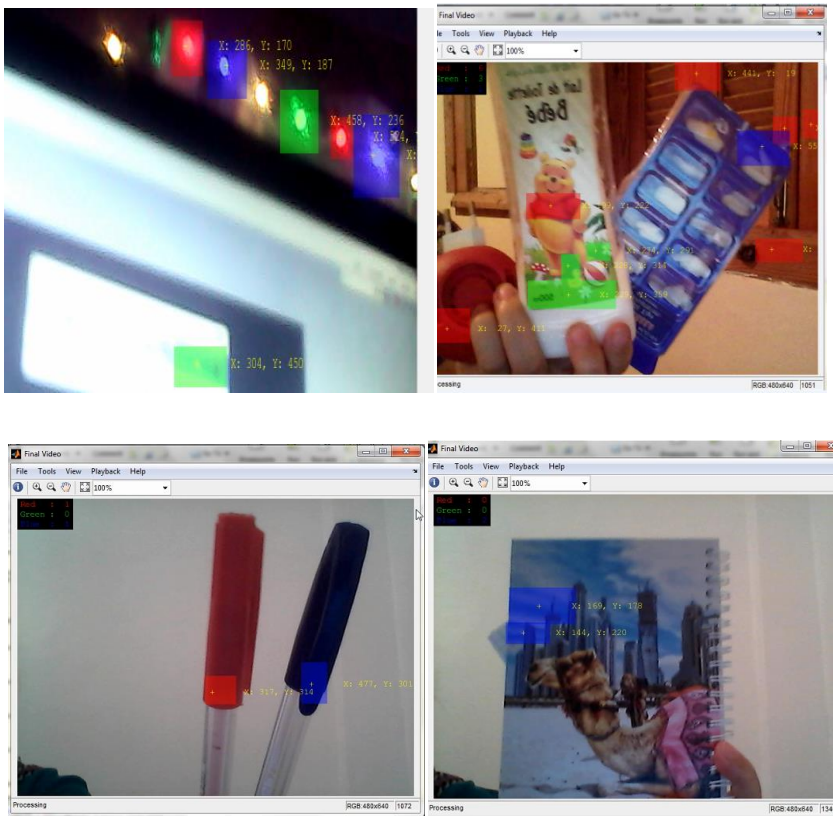


Figure3.14 : résultat de détection et poursuite d'objets on blanc

3.8.4 Résultat de la détection et poursuite d'objets selon les couleurs rgb :

Nous avons dans cette image (**figure**) plusieurs objets de couleur différents, nous sommes intéressé par la détection de l'objets qui une couleur rouge (avec rvb = **(255, 0, 0)), vert (avec rvb =(0 255 0)), bleu (avec rvb = 0 0 255)).**



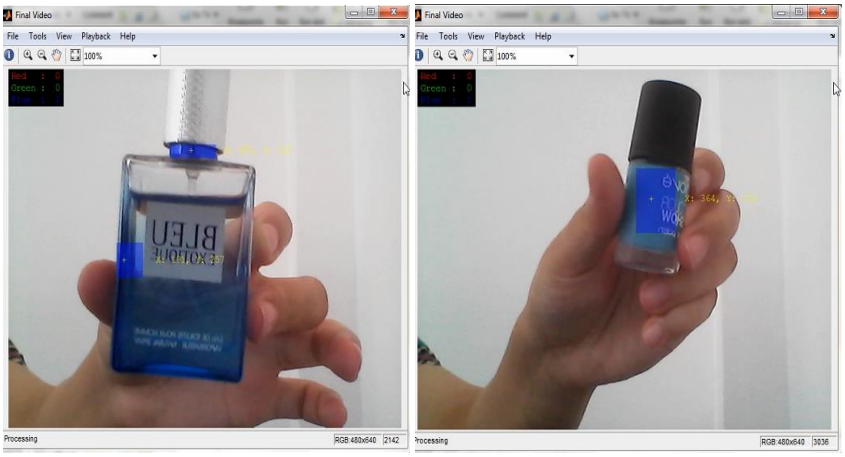
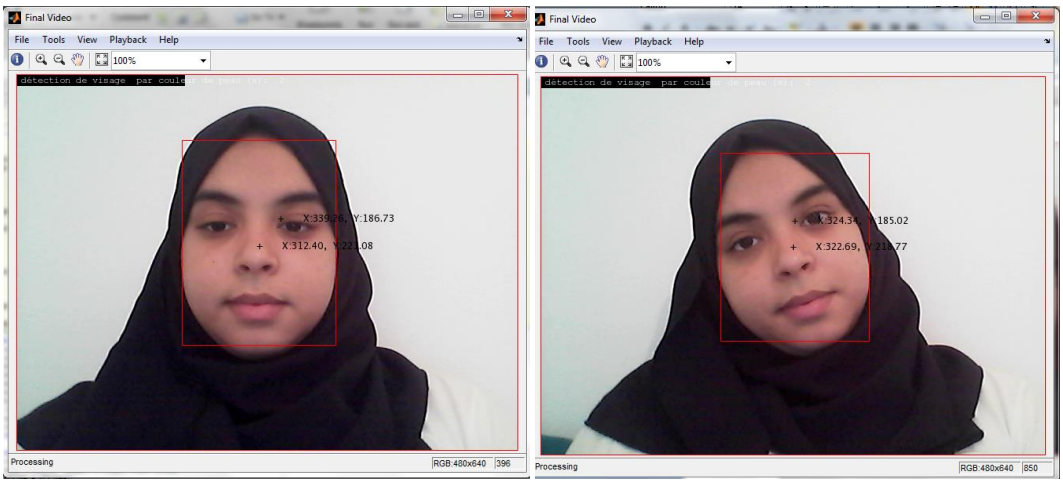


Figure3.15 : résultat de détection et poursuite d'objets en couleur vert , bleu , rouge

3.8.5 Résultat de la détection et poursuite d'objets selon la couleur de peau :



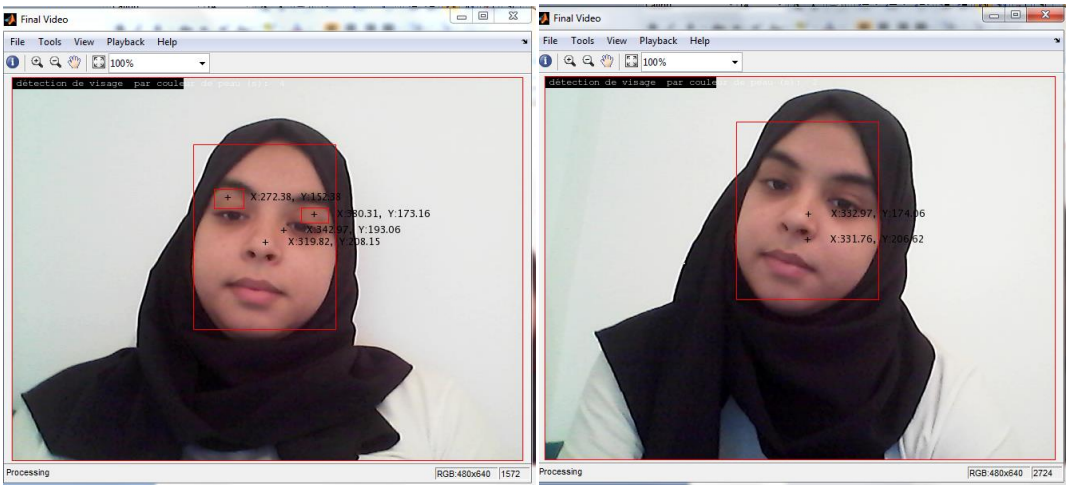


Figure 3.16: résultat de détection et poursuite de visage selon la couleur de peau.

3.8.6 discutions :

Nous avons traité dans notre projet le sujet de la détection et du suivi des objets, nous avons étudié de nombreux algorithmes, tels que l'algorithme de détection et poursuite de couleur ,KLT algorithme qui suivi les visages , Tous ces algorithmes étaient bons pour détecter des objets ou des visages, mais nous avons choisi l'algorithme de Viola Jones car il capables de détecter efficacement et en temps réel des objets dans une image et aussi facile à appliquer, caractérisé par facilité de calcul et rapidité.

Par contre, nous avons eu plusieurs problèmes :

3.8.6.1 Tests et résultat

3.8.6.1.1 Test de visage en face

Le programme détecte correctement les visages de l'image, soit contienne un ou plusieurs visages :

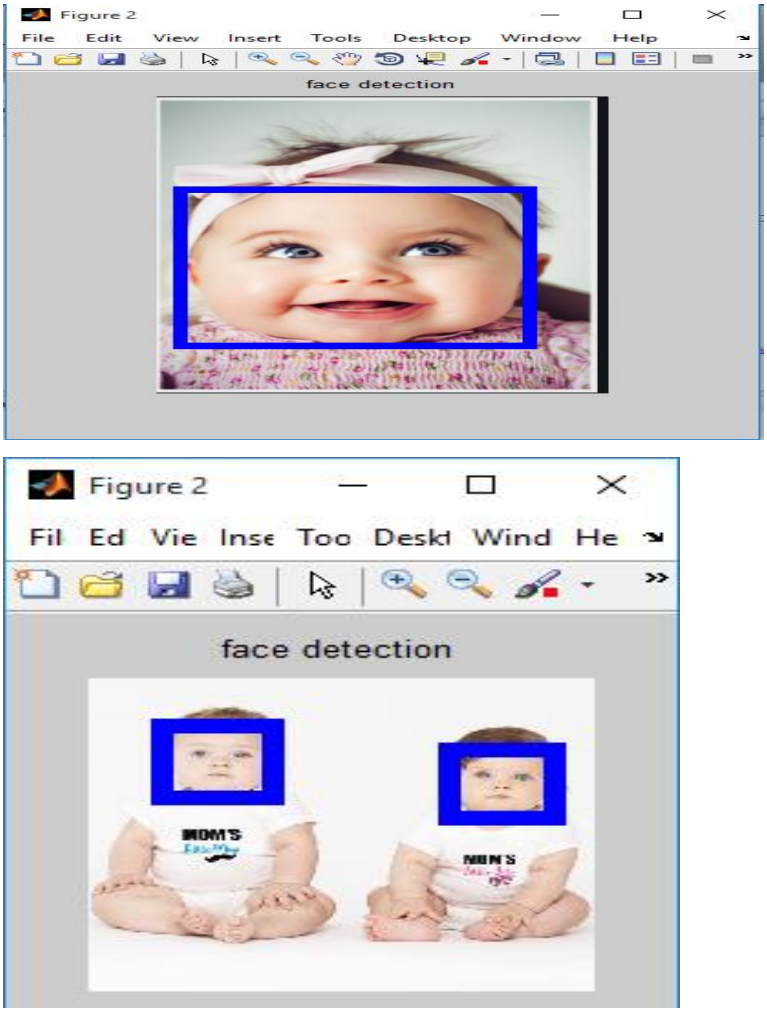


Figure 3.17 : Détection d'un ou plusieurs visages.

3.8.6.1.2Test de l'image contient des visages inclinés :

Le programme ne détecte rien et ne fonctionne pas quand des têtes présentées dans L'image sont inclinée latéralement ou verticalement.

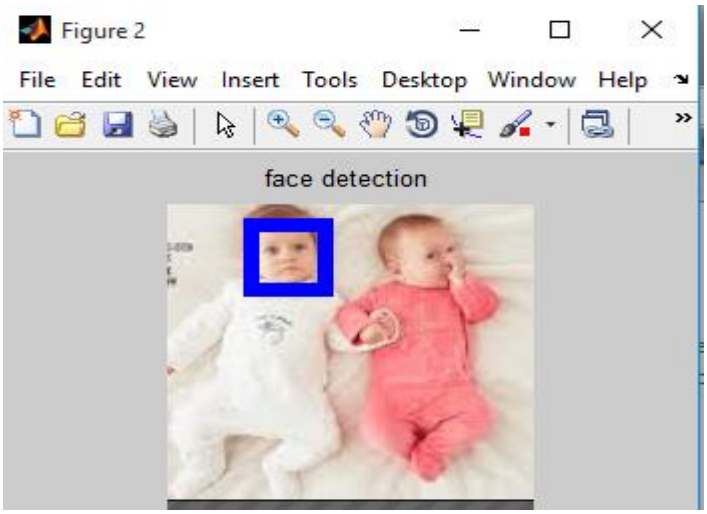


Figure 3.18 : Problème d'inclinaison des visages.

3.8.6.1.3 Test de l'image contient des visages Occlus :

Dans ce cas l'algorithme s'exécute mal si un visage est partiellement caché par un autre ou par un objet quelconque. Ce mal fonctionnement est dû à ce que la corrélation dans la partie cachée sera faible car elle ne correspond pas à un visage

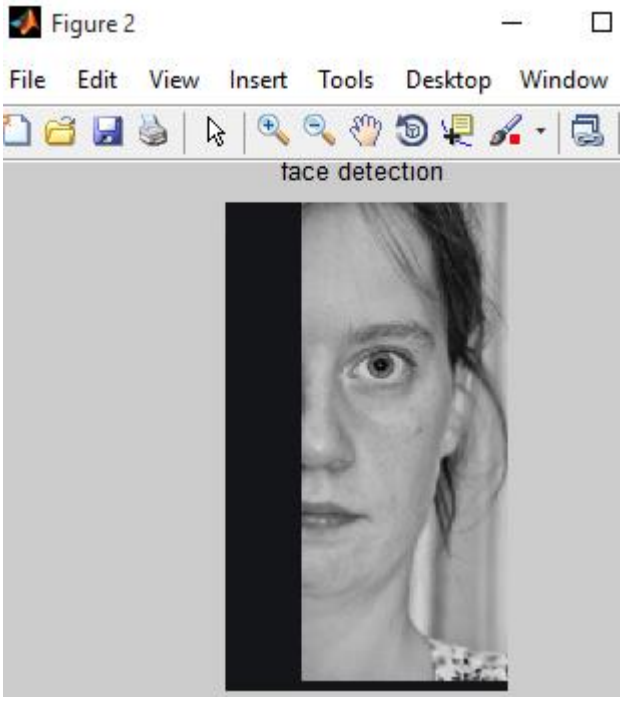


Figure 3.19 : résultat de détection des visages partiellement cachés.

3.8.6.1.4 Test sur des images à différentes conditions d'illumination :

À cause de l'existence de changement de l'éclairage, un même visage, avec la même expression faciale l'algorithme donne un mauvais résultat.

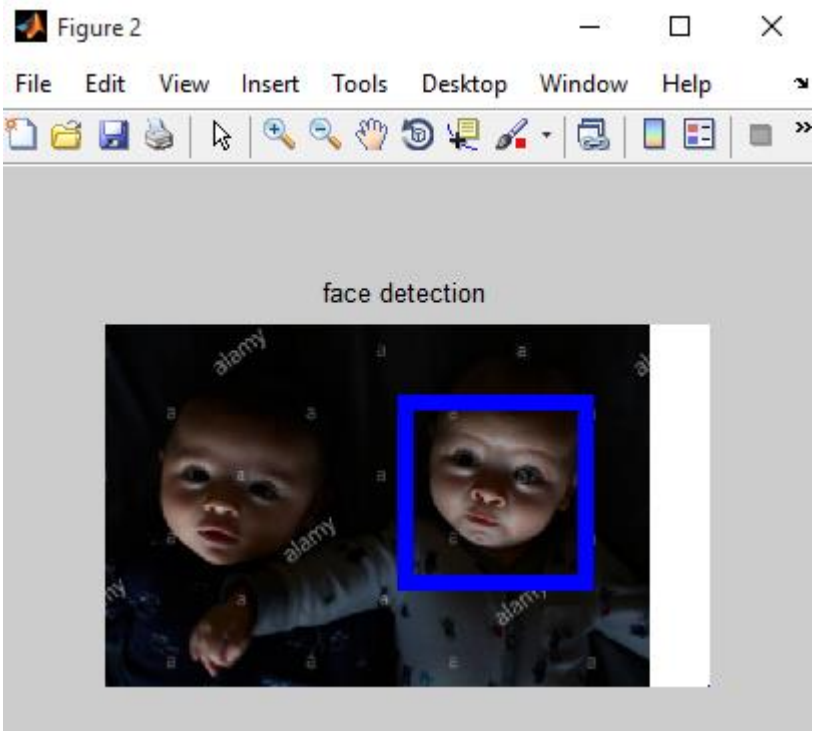


Figure 3.20 : Résultat de la détection des visages à différentes conditions d'illumination

3.8.6Conclusion :

Dans ce chapitre nous avons présente l'implémentation de les méthodes de détections et poursuites d'objets et de visage et les résultat obtenu. en utilisant le langage de programmation matlab Les tests ont été effectués sur les images originales non traitées. Durant notre étude, nous avons obtenu des résultats significatifs en utilisant un nombre important des images de différents types. Ces résultats ont été présentes et interprètes pour montrer l'efficacité de la méthode proposée.

Bibliographie

[1] CH. Berkane et A.Berkani ≪la détection des visage≫,mémoire de master ,université de labri ben M'hidi Oum el bouaghi,soutnence le 2014

[2] Jean –Charles Risch , Segmentation et détection d'objets en temps réel avec Tensorflow, société Elter (Société de services informatiques sp,2018

[3] https://bricabracinfo.fr/Telechargt/def_resol_pf.pdf

[4] https://www.cnrtl.fr/definition/vid%C3%A9o

[5] Melle MEDJAHED Fatiha, ≪ Détection et Suivi d'Objets en Mouvement Dans Une Séquence d'Images≫, mémoire de magister, Université des Sciences et de la Technologie d'Oran U. S. T. O, 2011/2012

[6] http://www.pc-code.com/base/numetlet/let/f/fps.php

[7] https://www.lirmm.fr/~strauss/MasterVision/TP_enonces/TP-Tracking/PoursuiteDeCible.pdf

[8]http://www.webreview.dz/IMG/pdf/07fr-rist14-2.pdf

[9] http://morpheo.inrialpes.fr/people/Boyer/Teaching/M2PGI/c4.pdf?fbclid=IwAR1uCc0fSZtDpja50yiWVRPkd6RfCk2SbNcvZkaKJxqgcEa3fQRqpAAJk8E

[10] SOUDANI Ibtissam, ≪ Le suivi d'une cible dans une séquence d'images≫,Mémoire de master, Université Labri Ben M'hidi Oum EL Bouaghi, 2013 /2014

[11]M. Kachouane et S. Sahki , ≪ Détection et poursuite visuelle de piétons en temps réel pour robot de type voiture≫,mémoire de master, université Saad Dahlab de blida1, 2011-2012

[12] https://www.codeflow.site/fr/article/how-to-build-a-neural-network-to-recognize-handwritten-digits-with-tensorflow

[13] Ch. Bencheriet, A/H. Boualleg, H. Tebbikh B. Guerzize, W. Belguidoum, ≪Détection de Visages par Méthode Hybride Couleur de Peau et Template Matching≫, Université 8 mai 45 de Guelma BP 401, Guelma 24000, Algérie, 2007

[14] https://translate.google.com/translate?hl=fr HYPERLINK "https://translate.google.com/translate?hl=fr&sl=en&u=https://realpython.com/lessons/viola-jones-object-detection-framework/&prev=search"& HYPERLINK "https://translate.google.com/translate?hl=fr&sl=en&u=https://realpython.com/lessons/viola-jones-object-detection-framework/&prev=search"sl=en HYPERLINK "https://translate.google.com/translate?hl=fr&sl=en&u=https://realpython.com/lessons/viola-jones-object-detection-framework/&prev=search"& HYPERLINK "https://translate.google.com/translate?hl=fr&sl=en&u=https://realpython.com/lessons/viola-jones-object-detection-framework/&prev=search"u=https://realpython.com/lessons/viola-jones-object-detection-framework/ HYPERLINK "https://translate.google.com/translate?hl=fr&sl=en&u=https://realpython.com/lessons/viola-jones-object-detection-framework/&prev=search"& HYPERLINK "https://translate.google.com/translate?hl=fr&sl=en&u=https://realpython.com/lessons/viola-jones-object-detection-framework/&prev=search"prev=search

[15] https://fr.slideshare.net/kingslim/une-approche-multiagents-pour-la-dtection-de-contours

[16] L.Moussaoui et F.Fares << Suivi de cible >>,memoire de master ,Université Larbi Ben M'hidi -Oum EL Bouaghi- ,Soutenu Le 09- 06- 2015 .