UNIVERSITE DU QUEBEC

MÉMOIRE PRÉSENTÉ À

L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIERES

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE MATHÉMATIQUES ET INFORMATIQUE APPLIQUÉES

PAR

ABBEY ADJOVI MATIALE

DÉTECTION DE VISAGES PAR YOLOV8 ET YOLOV5

JANVIER 2025

REMERCIEMENTS

Nulle œuvre n'est plus exaltante que celle réalisée avec le soutien moral et financier des personnes qui nous sont proches.

Je tiens à exprimer ma plus profonde reconnaissance à :

En premier lieu notre gratitude au Tout-Puissant pour nous avoir accordé la force et la détermination ce modeste travail.

À mes deux directeurs Professeur Demagna Koffi du département de génie mécanique et Professeur Meunier François du département de mathématiques et informatique appliquées, Professeurs à l'Université du Québec à Trois-Rivières, qui n'ont ménagé aucun effort pour me guider durant ce travail par leur entière disponibilité malgré leurs occupations en me procurant de meilleures idées et motivations qui ont été un tonus de mon avancement. Je vous exprime ma profonde gratitude.

À mon père, ma mère, mes frères et sœurs, en particulier à, M. Yawo APETCHO, mes amis (es) bien-aimés. Et aussi à tous ceux qui de près ou de loin m'ont apporté leurs aides.

À mes chers amis et collègues de l'université qui ne cessent également d'exceller dans leurs divers domaines d'études par leur dévouement vis-à-vis du travail afin d'acquérir de meilleurs résultats scolaires et clôturer l'année en toute beauté.

Mille mercis.

RÉSUMÉ

Avec les progrès continus de la puissance de calcul et la disponibilité abondante des données, l'apprentissage en profondeur est devenu largement utilisé dans diverses disciplines, avec un accent particulier sur la vision par ordinateur. Le domaine de la détection d'objets, une facette de la vision, a connu une profonde transformation grâce aux progrès des algorithmes d'apprentissage en profondeur. Ces progrès ont donné lieu à diverses applications qui s'appuient entre autres des CNN (Convolutional neural network) pour leur mise en œuvre. Grâce à cette évolution rapide, les modèles de détection d'objets en temps réel ont réalisé des progrès remarquables en termes de vitesse de détection et de précision. Notre objectif est d'étudier et de mettre en œuvre des algorithmes de détection d'objets d'apprentissage en profondeur pour détecter et compter les occurrences de visages dans une image. Notre étude s'est principalement centrée sur l'algorithme de détection d'objets en temps réel YOLO (YoloV5 et YOLOv8), une méthode largement reconnue et approuvée dans les applications pratiques.

Et pour identifier et compter le nombre de visages dans notre système, nous avons entraîné le modèle YOLO en utilisant les images prises par notre caméra et le web scraping pour récupérer les données. Nous avons obtenu des résultats très intéressants, atteignant un taux de précision de plus de 75 %. L'évaluation globale souligne l'avantage significatif de l'algorithme de détection d'objets YOLO en temps réel par rapport à de nombreux modèles alternatifs.

Mots clés : Réseaux de neurones convolutifs (CNN), Détection d’objets, Apprentissage en profondeur, YOLO.

ABSTRACT

With continual advances in computing power and the abundant availability of data, deep learning is now widely used in a variety of disciplines, with a particular focus on computer vision. The field of object detection, a facet of vision, has been profoundly transformed by advances in deep learning algorithms. These advances have led to various CNN-based applications for their implementation. Thanks to this rapid evolution, real-time object detection models have made significant progress in terms of detection, speed and accuracy. In this research, our aim is to study and implement deep learning object detection algorithms to detect and count faces in images and videos. Our study focused mainly on the Yolo real-time object detection algorithm, a generally recognized and accepted method in practical applications. And to identify and count the number of faces in our system we trained the Yolo model using the image taken by our camera and web scraping to extract the data. We achieved very good results, with an accuracy rate of 75%. The overall evaluation emphasizes the considerable advantages of the Yolo real-time object detection algorithm compared with several alternative models.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

Chapitre 1- INTRODUCTION

1.1 Généralités

Grâce à la vision par ordinateur et à l'apprentissage en profondeur, l'intelligence artificielle a connu un essor remarquable ces dernières années. Les algorithmes d'apprentissage profond dans la détection d'objet sont très utilisés dans les recherches et tâches industrielles. D'où la possibilité de reconnaissance des objets sur une image.

De nos jours et dans la plupart des endroits, la vidéosurveillance est l'une des solutions de sécurité utilisant des techniques de traitement d'images pour la détection et la reconnaissance d'un objet ou d'un visage dans une séquence d'images.

Dans ce mémoire, notre objectif principal sera la détection des visages dans les images prises et ensuite de pouvoir comparer ces performances de détection avec les versions de Yolo V5 et V8 (You Only Look Once). Parmi de nombreux algorithmes d'apprentissages profond nous avons opté pour le modèle Yolo car il exploite les réseaux de neurones convolutifs, donne une meilleure précision de détection, permet d'obtenir les meilleurs résultats, pour sa rapidité de détection et adapté pour les systèmes de temps réel.

De ce fait, notre travail consistera à détecter les visages dans les images prises puis les comparer deux versions de Yolo : dont nous avons choisi YoloV8 et YoloV5.

1.2 Objectifs

Notre projet a pour but de comparer de façon globale deux modèles de Yolo (Yolov5 et Yolov8). Pour y arriver, les objectifs en sont :

- Déterminer la performance et la robustesse des deux modèles.
- Comparer les deux modèles avec les mêmes types de données.
- Comparer la précision et la vitesse de détection des deux modèles.

1.3 Problématique

Reconnu pour son efficacité, sa rapidité et son architecture légère, YoloV5 a été affiné pour fonctionner sur plusieurs plates-formes, flexible d'implémenter (modulaire, extensible, intégrable et portable). A l'inverse, YoloV8 intégrant de nouvelles fonctionnalités par son architecture pour améliorer la détection d'objet dans les situations plus complexes. D'où nous pouvons poser cette question : « quelles sont les améliorations et distinctions entre YoloV5 et YoloV8 en ce qui concerne la précision de détection d'objets et les performances de traitement d'images? »

Ce projet va permettre de comparer YoloV5 et YoloV8 en analysant les détails de performance en matière de vitesse de traitement, de détection d'objet et la robustesse du modèle.

1.4 Contenu du mémoire

Le premier chapitre présente le sujet du mémoire, son objectif, et la problématique. Ensuite le deuxième chapitre est consacré à l'état de l'art, les méthodes et techniques de détection. Par la suite le chapitre trois explore la détection faciale et les étapes de la détection. Le quatrième chapitre présente la conception et la réalisation de notre projet. Le chapitre cinq présente les bibliothèques utilisées, les interfaces, les comparaisons des

résultats et leurs discussions. Enfin le sixième chapitre présente la conclusion de notre projet.

Chapitre 2- ÉTAT DE L'ART

2.1 Introduction

L'objectif de cette recherche est de construire un système qui automatise l'analyse des informations contenues dans les images et plus spécifiquement pour réaliser la tâche de détection de visage. Dans ce contexte il est essentiel de disposer d'algorithmes à la fois efficaces et robustes. En effet, la capacité à détecter avec précision les visages est cruciale dans diverses applications. En réalité, l'importance de la détection des visages ne peut être surestimée lorsqu'il s'agit de tout système impliquant l'analyse des visages. L'objectif est de détecter toutes les zones d'une image qui représentent un visage, sans tenir compte de leur emplacement. Pour mieux comprendre les effets de la position, de l'orientation et de l'éclairage sur la perception visuelle, de nombreuses études ont été menées. Plus précisément, la recherche s'est concentrée sur ces facteurs et leur impact sur la façon dont les individus perçoivent les stimuli visuels. Il existe de nombreuses méthodes pour identifier les visages dans les images statiques qui ont été développées au fil du temps. La reconnaissance faciale a été effectuée pour la première fois dans les années 1970 par à l'université de Tokyo. La détection faciale est alors surtout utilisée à des fins de sécurité [1]. Certaines interfaces homme machine utilisent non seulement le clavier et la souris mais de nouvelles formes de sources d'information comme la reconnaissance des visages. Dans ce chapitre, nous présenterons l'état de l'art sur les techniques de la détection faciale.

2.2 Définitions des concepts

Le processus de détection des visages consiste à détecter les visages humains et à localiser leur emplacement dans une image numérique. Cette technique est une variante de la détection d'objets, avec un accent particulier sur la localisation et le comptage et l'identification des visages dans une image [2]. Les experts en vision par ordinateur ont manifesté un grand intérêt pour cette méthode, comme en témoignent les nombreux articles scientifiques, conférences et brevets consacrés à son étude [2]. L'étude de la détection de visage a également contribué au développement de techniques plus générales de détection d'objets, grâce à un corpus de recherche conséquent dans ce domaine [2]. La vidéosurveillance peut grandement bénéficier de l'utilisation de la technologie de détection de visage, compte tenu de ses nombreuses applications directes. Les domaines de la biométrie, de la robotique, du contrôle des interfaces homme-machine, de la photographie et de l'indexation d'images sont étroitement liés et interdépendants. Les autres fonctionnalités incluent la recherche d'images en fonction du contenu, ainsi que des vidéos.

2.3 Les méthodes de détection de visage

Plusieurs méthodes ont été développées pour la détection de visage et sont subdivisées en quatre groupes :

2.3.1 Méthodes basées sur des caractéristiques invariantes

L'objectif de cette technique est d'identifier les caractéristiques structurelles d'un visage, indépendamment de toute autre variable telle que figures des points caractéristiques du contenu du visage. L'apparence de l'objet peut varier en fonction de sa position, de l'éclairage ou de l'angle sous lequel il est observé. Un inconvénient majeur de l'utilisation de cette approche est qu'elle peut réduire considérablement la qualité des images. Les performances de l'algorithme peuvent être entravées par l'éclairage, le bruit ou l'occlusion, ce qui entraîne des problèmes de performance. Le visage possède certaines caractéristiques et attributs immuables qui sont considérés comme ses propriétés ou caractéristiques inhérentes. Les principales caractéristiques incluent :

2.3.1.1 Méthodes basées sur la couleur de la peau

La caractéristique spécifique de la couleur de la peau humaine est attribuée aux visages. Elle est souvent utilisée comme facteur dans la technologie de reconnaissance faciale. En effet, ce facteur est considéré comme une caractéristique distinctive qui peut aider à identifier la présence d'un individu. Cependant, cette pratique a suscité une vaste controverse en raison de problèmes de préjugés raciaux et de discrimination. Il est important de considérer les conséquences et les implications potentielles de l'utilisation d'une telle technologie. Il est important de reconnaître et d'apprécier la diversité au sein et entre les cultures. Il est également important de reconnaître et de traiter toute inégalité ou discrimination pouvant exister en raison de différences culturelles [49].

2.3.1.2 Méthodes basées sur les caractéristiques du visage

L'approche employée par cette technique implique l'utilisation de plans de commande appelés " Canny detector", en conjonction avec des heuristiques. Elle ne conserve que les groupes de points caractéristiques qui ont une forme elliptique. La bordure qui sépare l'arrière-plan du visage correspond à une ellipse en raison de sa forme caractéristique. La figure 2-1 permet d'observer les points caractéristiques d'un visage déduit par l'approche développée par Smith [48]. On dit que la formation de cette entité provient des points de discontinuité trouvés dans la fonction de luminance.

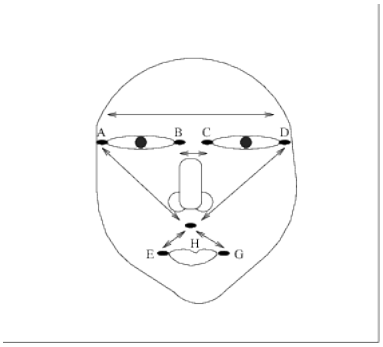

Figure 2-1 Points caractéristiques du visage [48]

2.3.2 Méthodes basées sur l'apparence

Les techniques d'analyse statistique sont à la base de la méthode basée sur l'apparence. Les traits distinctifs entre les visages et les non-visages sont remarquables. Il existe plusieurs méthodes pour l'approche donnée, chacune étant basée sur un principe distinct. Une caractéristique distincte du visage ou un trait particulier du visage, peut être

identifié avec précision. L'interprétation produite peut être formulée en termes de probabilité [50].

2.3.3 Méthodes basées sur les connaissances acquises

Dans les méthodologies qui s'appuient sur les connaissances, il existe un certain degré de connaissances préalables impliquées (connaissances contextuelles, connaissances statistiques, etc). Sur la base de la géométrie faciale, certains individus communément appelés gouvernants utilisent ces techniques. Ces caractéristiques sont toutes interconnectées et leur placement les uns par rapport aux autres détermine l'aspect général du visage. Il est également important de prendre en compte la forme du visage, par exemple s'il est rond ou anguleux, car cela peut affecter la façon dont les traits sont perçus. De plus, les expressions faciales peuvent modifier l'apparence de ces traits et transmettre différentes émotions. La combinaison de tous ces facteurs contribue à l'unicité du visage de chaque individu. Certaines règles régissent les distances et les positions relatives entre les sourcils, le nez et la bouche. La symétrie des yeux et des sourcils sont des traits qui les distinguent et doivent être reconnus [51].

2.3.4 Méthodes basées sur la mise en correspondance

Un motif facial (texture de la peau, distance entre les yeux, contour des yeux, etc) commun est utilisé dans les techniques d'appariement. La face avant du modèle est généralement déterminée soit par une prédéfinition manuelle, soit par un paramétrage. Ce processus consiste à analyser le degré d'association entre deux ensembles de données. Les

valeurs résultantes aident à identifier les similitudes et les différences entre elles, et à déterminer dans quelle mesure elles sont liées. L'image d'entrée permet une identification indépendante des caractéristiques faciales telles que le contour du visage, des yeux, du nez et de la bouche. Bien que la mise en œuvre de cette approche soit simple, sa capacité de détection et d'identification est insuffisante. Ils en existent qui peuvent être utilisés pour relever ces défis, comme l'utilisation de modèles 3D ou l'utilisation d'algorithmes d'apprentissage automatique pour générer des images de visage plus diversifiées et réalistes. Malgré la difficulté de la tâche, la capacité à générer des images de visage réalistes et diversifiées a de nombreuses applications potentielles dans des domaines tels que la réalité virtuelle, le divertissement et l'application de la loi. Les modèles multi-échelles, multi résolutions, déformables et les sous-modèles sont tous des composants importants [52].

2.4 Techniques de détection de visage

2.4.1 Technique d’analyse en composantes principales

L’analyse des composantes principales (ACP) est une technique de réduction de dimensions qui consiste à convertir un ensemble de variables en un ensemble de combinaisons linéaires permettant la maximisation de la validité des composantes projetées à partir des données initiales. Cette technique permet de transformer un espace de données en un autre espace de dimension inférieure tout en minimisant la perte d’information.

Le domaine de la statistique est engagé dans la recherche active de la réduction de la dimensionnalité des données. Ce problème revêt une importance significative dans divers domaines, notamment la prévision, la classification des documents, la bio-informatique, la reconnaissance d'objets et la modélisation de processus technologiques complexes. Il est courant de rencontrer des ensembles de données comportant des milliers de composantes. Même si toutes les composantes peuvent être importantes pour certains problèmes, il ne s'agit généralement que d'un petit sous-ensemble de composantes pertinentes pour des concepts cibles spécifiques. L'ACP implique la transformation d'un ensemble de données initial, constitué d'échantillons vectoriels, en un nouvel ensemble d'échantillons vectoriels avec des dimensions dérivées. Le concept sous-jacent peut être résumé comme suit : un ensemble d'échantillons vectoriels à n dimensions, noté X = x1, x2, x3, x4, …, xn, doit être converti en un autre ensemble, Y = y1, y2., y3, y4, …, yn, avec la même dimensionnalité. Cependant, les vecteurs y possèdent la particularité que la majorité de leurs informations sont concentrées dans les premières dimensions. Par conséquent, nous sommes en mesure de réduire l'ensemble de données à un plus petit nombre de dimensions, ce qui entraîne une perte minimale d'informations [40]. En générale, l'ACP est une technique statistique qui permet de réduire le nombre de varaible dans un ensemble de données tout en gardant le plus important de l'information. C'est particulièrement utile pour visualiser, faire le prétraitement des données et aider les performances des machines a apprendre automatiquement.

2.4.2 Technique SVD (Décomposition en Valeurs Singulières)

L'ère informatique a été grandement influencée par la factorisation matricielle importante connue sous le nom de décomposition en valeurs singulières (SVD). Cette technique mathématique sert de base à un système de détection faciale très efficace, permettant le calcul des visages propres (Eigenfaces). Les Eigenfaces, à leur tour, offrent une méthode simplifiée et efficace pour représenter des images dans le domaine de la détection faciale

L'utilité de la SVD est évidente dans une variété de tâches. Explorons quelques exemples. Une application courante implique des situations dans lesquelles la matrice de données A ressemble beaucoup à une matrice de bas rang. Dans de tels cas, il devient utile d'identifier une matrice de bas rang qui fournit une approximation fiable de la matrice de données d'origine. En utilisant la décomposition spectrale de A, nous pouvons dériver une matrice B de rang k qui offre la meilleure approximation possible de A. Ce processus peut être répété pour toute valeur souhaitée de k. Contrairement à la décomposition spectrale plus couramment utilisée en algèbre linéaire, la décomposition en valeurs singulières englobe toutes les matrices, qu'elles soient rectangulaires ou carrées. Il convient de noter que la familiarité avec les vecteurs propres et les valeurs propres n'est pas supposée ici, mais ceux qui sont familiers reconnaîtront la nécessité de certaines conditions sur la matrice pour garantir l'orthogonalité des vecteurs propres. D'autre part, les colonnes de V dans la décomposition en valeurs singulières, appelées vecteurs singuliers droits de A, forment toujours un ensemble orthogonal sans aucune hypothèse sur A. De même, les colonnes de U, appelées vecteurs singuliers gauches, constituent également un ensemble orthogonal.

Elle est très importante car elle aide a trouvé les principales variations des données sans faire le calcul clair de la matrice de covariance. La SVD donne donc un moyen stable rapide et mathématiquement rigoureux pour réduire les dimensions des données en gardant le plus important de leur structure.

2.4.3 Technique SVM (Machine à vecteurs de Support)

L'ensemble d'algorithmes machine connu sous le nom de SVM est utilisé dans l'apprentissage automatique pour résoudre divers problèmes, notamment la classification d'images pour la détection faciale. Cette approche est réputée pour sa remarquable adaptabilité et sa nature conviviale, la rendant accessible même à ceux qui découvrent le domaine de l'apprentissage automatique. La figure 2-2 permet d'observer les deux classes distinctes (points bleus et points rouges) [41]. L'objectif principal des SVM est d'établir deux groupes de données distincts, efficacement séparés l'un de l'autre par une frontière qui maximise la distance entre les deux groupes.

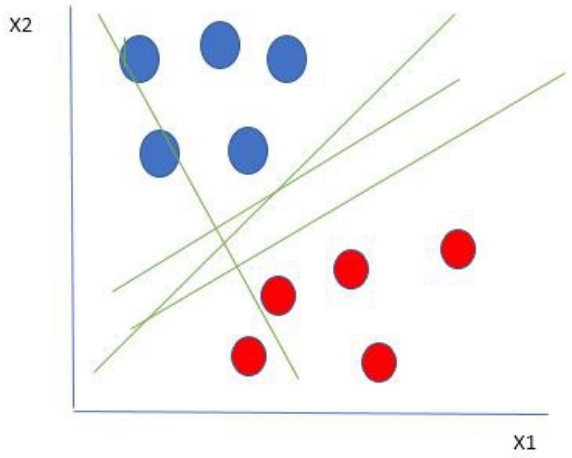

Figure 2-2 Classification SVM **[41]**

2.5 Les attributs du visage

2.5.1 L'iris

Le concept d'utilisation de l'iris comme moyen d'identification des individus a été initialement proposé par l'ophtalmologiste Frank Burch en 1936 [47]. En 1987, les ophtalmologistes Aran Safir et Leonard Flom ont poussé cette idée encore plus loin en obtenant un brevet. Ces spécialistes ont ensuite fait appel à l'expertise de John Daugman, qui enseignait à l'époque à l'Université Harvard, pour développer des algorithmes d'identification de l'iris. Les algorithmes brevetés de Daugman, basés sur les ondes de Gabo, servent de base à tous les systèmes d'identification de l'iris. Contrairement à d'autres méthodes d'identification, l'identification de l'iris englobe un plus grand nombre de paramètres, ce qui entraîne un niveau de fiabilité qui dépasse la simple identification et s'oriente vers l'authentification [42].

John Daugman a développé une méthode de caractérisation des iris, qui constitue actuellement la manière la plus efficace de les différencier. Même les vrais jumeaux ont des caractéristiques d'iris uniques qui permettent une identification facile. Les calculs de Daugman ont déterminé que la probabilité de trouver deux iris suffisamment identiques est incroyablement faible, à 1 sur 10^{72}.

2.5.2 La rétine

Le film photographique de l'œil est appelé rétine. La rétine, située à la partie postérieure de l'œil, est constituée de plusieurs couches de cellules (cellules bipolaires, cellules ganglionnaires, photorécepteurs et les cellules gliales). Le facteur distinctif entre deux rétines réside dans les veines qui les traversent. Ces veines possèdent un motif cohérent et distinctif qui diffère non seulement entre les individus mais également entre les yeux gauches et droit. Les modèles résultants dérivés de ces modèles héritent de la même stabilité. Cette stabilité inhérente empêche les fraudeurs de reproduire une rétine, garantissant ainsi un niveau de sécurité exceptionnellement élevé.

2.5.3 Le visage

Le fait d'utiliser les traits du visage pour reconnaître les individus est une pratique instinctive et innée. Parmi les diverses caractéristiques biométriques utilisées par les humains pour l'identification personnelle, les images faciales sont sans aucun doute les plus répandues et les plus largement utilisées.

En utilisant une caméra, il devient possible d'enregistrer les contours du visage d'une personne et de discerner des caractéristiques spécifiques. Le positionnement de l'individu

par rapport au dispositif varie en fonction du système, allant du placement direct face à l'avant à la capture du mouvement à une distance prédéterminée. Les informations biométriques acquises sont ensuite croisées avec le fichier de référence.

Au cours des années 1970, la technologie de la détection faciale reposait largement sur des caractéristiques faciales quantifiables telles que la distance entre les yeux, la forme des sourcils, la position des lèvres et du menton, ainsi que d'autres caractéristiques similaires. Cependant, les progrès des techniques de traitement d'images depuis les années 1990 ont permis l'utilisation de diverses technologies tirant parti de ces découvertes. Plus récemment, les réseaux de neurones ont joué un rôle important dans l'amélioration des capacités de détection faciale.

Différentes méthodes existent pour analyser et détecter les visages. Cependant, ce qui est particulièrement intéressant dans ces techniques, c'est qu'elles se concentrent sur les éléments du visage les moins sujets à altération, tels que les larges zones supérieures des orbites, les régions entourant les pommettes, les côtés de la bouche et d'autres zones similaires. Cela permet d'ignorer les changements tels que les coiffures. Toutes ces techniques impliquent le processus d'identification d'individus parmi des bases de données contenant des milliers, voire des centaines de milliers de personnes.

2.5.4 Les attributs du visage

- **Morphologie**

La détection et l'identification du visage a comme objectif est d'isoler des segments ressemblant à des yeux, car ils sont considérés comme les caractéristiques les plus

durables et les plus marquantes du visage humain. Ces segments correspondent aux contours des yeux et sont extraits selon l'approche suivante : D'abord identifier ces pixels présentant un contraste local prononcé en intensité, identifier parmi ces pixels les segments susceptibles de faire partie des yeux, utiliser des combinaisons géométriques pour identifier les composants restants du visage potentiel, confirmation grâce à l'utilisation d'un réseau neuronal [43].

- **Couleur de la peau**

La détection des visages s'appuie aussi historiquement sur les caractéristiques distinctes de la couleur de la peau humaine. La couleur de la peau varie selon les individus, en fonction de leur origine ethnique (africaine, européenne, asiatique, etc.). La divergence la plus significative réside dans la plage d'intensités lumineuses plutôt que dans la chrominance [44].

- **La texture**

La qualité unique de la texture de la peau, des cheveux, du visage humain permet de différencier les visages des autres objets. Une technique a été conçue par des chercheurs qui permet d'identifier des visages dans une image en utilisant uniquement la texture [46]. Le calcul de la texture est réalisé en employant des attributs de second ordre sur des sous-images mesurant 16 à 17 pixels. Au sein de cette méthodologie, l'analyse intègre trois catégories distinctes de caractéristiques : la peau, les cheveux et les autres éléments du visage [43].

- **Les contours**

L'analyse des images repose en grande partie sur la détection des contours, qui fournit non seulement des informations cruciales sur l'image, mais aide également à identifier et à décrire les formes présentes dans celle-ci. Une quantité importante de recherche est consacrée au développement de méthodes d'extraction automatique des contours des traits du visage.

Les traits du visage qui restent constants sont les yeux, les sourcils, les lèvres et le menton. Pour y parvenir, un modèle paramétrique précis est établi pour englober toutes les variations et distorsions imaginables. Dans la première étape, la phase d'initialisation, les points clés du visage tels que les coins de la bouche et des yeux sont identifiés et utilisés comme points de référence de départ pour chaque modèle individuel du visage d'une personne. Passant à la phase d'évolution, chaque modèle subit une déformation pour s'aligner sur les contours spécifiques des traits du visage analysés. Ce processus de déformation consiste à maximiser le flux des gradients, que ce soit en termes de luminance

ou de chrominance, le long des courbes définies par chaque modèle. En définissant ces modèles, une contrainte de régularisation est naturellement incorporée pour garantir le maintien des contours souhaités. [43]

- **Les pigmentations**

Les pigmentations et des séquences linéaires partagent une orientation commune. Le modèle facial utilisé dans l'étude comprenait deux points sombres représentant les yeux et trois points clairs représentant les pommettes et le nez. Les lignes du visage se distinguaient par des traits affichant une orientation similaire [43].

Ces caractéristiques spécifiques jouent un rôle essentiel dans la réalisation d'une discrimination faciale précise. En utilisant ces détails complexes, les réseaux neuronaux effectuent efficacement des tâches de la détection faciale.

2.6 Extraction de caractéristiques du visage

L'objectif est d'identifier et d'extraire les traits du visage. La capacité de permettre de différencier chaque visage tout en restant résilient aux changements est très important. Il comprend les caractéristiques uniques du visage, telles que la forme des yeux, du nez et de la bouche, ainsi que toutes les marques distinctives telles que les cicatrices ou les taches de naissance.

2.6.1 Les caractéristiques d’apparences

La reconnaissance et la synthèse des expressions ont été au centre des travaux de Franck et de Al. Le sujet de cette étude est l'utilisation de modèles actifs en relation avec les expressions faciales. Plus précisément, ce modèle est utilisé dans une nouvelle application. L'apparence joue un rôle essentiel dans l'analyse et la synthèse des visages expressifs. La reconnaissance des expressions faciales consiste à identifier les émotions véhiculées par les traits du visage d'une personne. Les six expressions universelles qui sont largement reconnues par les chercheurs dans le domaine sont prises en considération au cours de ce processus. À la suite de la description des modèles qui suscitent des expressions de joie, de colère, de peur, de dégoût, de tristesse et de surprise, une méthode est suggérée. Après avoir analysé l'expression faciale d'une personne, il est possible d'en créer une artificiellement par synthèse. Afin de modéliser avec précision un même visage, deux approches distinctes ont été proposées. Ces approches sont conçues pour fournir des représentations précises de la structure faciale étudiée. L'interaction humaine dépend fortement des signaux non verbaux, et les expressions faciales sont un élément crucial de cette communication. Avec l'utilisation des muscles faciaux, nous pouvons transmettre une pléthore d'émotions, de pensées et d'intentions sans avoir besoin de les vocaliser. Un domaine qui en a intrigué plus d'un est la reconnaissance des expressions faciales, et l'Active Appearance Model (AAM) est un modèle exemplaire qui a suscité un intérêt considérable en raison de son potentiel dans ce domaine. La démonstration montre comment les expressions faciales peuvent apparaître automatiquement.

2.6.2 Normalisation géométrique

Lors de l'utilisation d'un algorithme de reconnaissance qui repose sur la réduction de l'espace, il est crucial de prendre en compte la normalisation géométrique des images faciales. Ce processus de normalisation consiste à extraire la zone du visage de l'image d'origine, suivie d'une rotation du visage afin d'aligner l'axe de l'œil avec l'axe horizontal. Enfin, une réduction est appliquée par rapport à la distance entre des yeux. En suivant ce processus, une image faciale est obtenue avec la distance correcte entre les centres des yeux. Les dimensions de l'image faciale sont déterminées par la distance requise entre les centres des yeux.

2.7 La classification

L'identificateur ultime des expressions faciales n'est obtenu que par une évaluation méthodique et automatisée. Les expressions faciales peuvent être reconnues grâce à l'extraction de caractéristiques (zone d'intérêt, caractéristiques basées sur mouvements, caractéristiques géométriques. etc). Bien que tous les classificateurs ne le fassent pas, certains systèmes classent directement les expressions. De nombreux classifieur utilisant les caractéristiques énumérées précédemment, ont été appliqués à la reconnaissance d'expression tels que: Régression logistique multinomiale (Multinomial Regression Logistic, MRL), Modèles de markov cachés (Hidden Markov Model, HMM), Réseaux bayésiens (Bayesian Network, BN), Analyse discriminante linéaire (Linear discriminant analysis, LDA), K-plus proche voisin (K Nearest Neighbor, KNN), Réseaux de neurone

(Neural Networks, NN), Machines à vecteurs de support (Support Vector Machine, SVM) [3].

2.8 Apprentissage en profondeur

Le terme "apprentissage en profondeur" fait référence à un ensemble complet de méthodes et d'approches. Des avancées significatives dans le domaine du renseignement ont été réalisées grâce à l'intégration de l'apprentissage automatique. Les progrès de l'apprentissage en profondeur ont été rendus possibles principalement grâce à l'amélioration des capacités informatiques et la création de bases de données volumineuses « Big Data » [4]. De nouveaux algorithmes pour le Deep Learning n'ont émergé qu'à la suite de ses échecs (manque de données annotées, difficulté à entraîner de grands modèles, etc). Dans un effort pour améliorer la création d'algorithmes conventionnels pour diverses tâches, l'apprentissage automatique est utilisé pour résoudre un large éventail de problèmes d'intelligence artificielle (IA). L'intelligence artificielle a la capacité de gérer de grandes quantités de données, y compris les mégadonnées, et est conçue pour résoudre tout type de problème tout en extrayant automatiquement les caractéristiques pertinentes. Donc les réseaux de neurones sont utilisés par l'apprentissage en profondeur. Il classifie les visages par la collecte des visages étiquetées qui sont ensuite prétraiter. Ces images prétraitées exploitent le réseaux neuronaux convolutionnels pour extraire des caractéristiques telles que le nez, la bouche, les yeux.

2.9 Conclusion

En examinant le processus de détection faciale, nous avons acquis des connaissances et des enseignements précieux, notamment les applications pratiques de la technologie de détection faciale et un examen complet des diverses méthodes et systèmes d'identification des visages humains. Le prochain chapitre approfondira le sujet de la détection faciale.

Chapitre 3- DETECTION FACIALE

3.1 Introduction

Au cours des dix dernières années, des subdivisions spécifiques de l'intelligence artificielle ont fait des progrès notables, en particulier la reconnaissance faciale. Ce domaine a suscité l'intérêt de plusieurs domaines et a connu des améliorations significatives. En conséquence, de nombreuses applications prometteuses font surface dans le domaine de l'analyse d'images en raison de sa vaste applicabilité. Généralement, la reconnaissance faciale repose sur le concept de détection de visage. Des algorithmes complexes sont utilisés pour traiter les images contenant des visages. Les algorithmes conçus pour la reconnaissance ou l'identification d'individus se concentrent principalement sur la zone faciale d'une image. Cela est possible en raison de certains traits mutables, tels que l'orientation, la posture, le positionnement, l'expression faciale, le teint, la présence ou l'absence de lunettes, les différences de traits du visage, l'éclairage et la résolution de l'image. Les techniques reposant sur la biométrie sont largement adoptées, souvent célébrées dans la culture populaire à travers des films et des émissions de télévision. Cependant, l'augmentation du vol d'identité a entraîné une demande pressante de technologie biométrique dans divers contextes qui nécessitent des mesures de sécurité renforcées, telles que l'accès restreint aux lieux sensibles et la surveillance des aéroports. Le traitement d'image et la vision par ordinateur utilisent souvent la détection de cible, une technique permettant d'identifier des objets spécifiques tels que des bâtiments, des arbres, des voitures et même des visages, s'ils existent dans l'image. L'aspect clé de la

détection de cible est la détection de ces objets, qui est l'une des principales interactions impliquées.

3.2 Historique

Les origines de la reconnaissance faciale remontent aux années 1950 et 1960, mais les recherches sur la reconnaissance faciale automatique sont généralement reconnues comme ayant commencé dans les années 1970. C'est alors que la publication "Identification of human faces" de Goldstein et al. a présenté une première tentative, quoique rudimentaire, d'identification des visages. L'approche suggérait d'étudier 21 caractéristiques subjectives différentes du visage, y compris l'épaisseur des lèvres et la couleur des cheveux, comme moyen d'identification. L'année 1987 a marqué la sortie du travail fondateur de Sirovich et Kirby, "Une procédure de faible dimension pour la caractérisation des visages humains » [33]. Cela a ensuite été suivi par "Face Recognition Using Eigenfaces » [34] de Turk et Pentland en 1991. En outre, la recherche sur la reconnaissance faciale a considérablement augmenté dès le début des années 1990, grâce aux progrès du matériel et à l'importance toujours croissante d'applications. Le principal inconvénient de l'utilisation de photographies comme moyen d'identification était l'incapacité d'éliminer la subjectivité.

3.3 Définition

Le processus de reconnaissance faciale implique l'examen de visages humains dans des images dans le but de déterminer leur identité. Cela peut être accompli grâce à la

vérification, qui est utilisée lorsque nous cherchons à déterminer si un visage particulier est présent dans une base de données. Alternativement, la reconnaissance est utilisée lorsque la base de données prend la forme d'images étiquetées et que nous souhaitons identifier la personne figurant dans l'image par son nom.

3.4 Méthodes de détection

Divers scientifiques ont proposé de nombreuses catégorisations de techniques de détection de visage. Cependant, les classifications de techniques de détection qui ont été mises en œuvre sont celles développées par Ahuja, Kriegman et Yan [45]. La catégorisation des méthodes est divisée en quatre groupes distincts, chaque groupe étant composé d'algorithmes de détection pouvant appartenir à plus d'une catégorie.

3.4.1 Algorithme basé sur les connaissances

La détection des visages par ces systèmes dépend de règles qui sont formulées à travers les connaissances humaines. Ces règles (position et taille des caractéristiques, symétrie du visage, etc)se concentrent principalement sur les caractéristiques physiques visibles du visage humain, telles que la présence d'une bouche, d'un nez et de deux yeux, qui sont disposés selon un schéma spécifique avec de légères variations d'une personne à l'autre. Le principal inconvénient de cette approche particulière découle du défi de créer une règle normalisée. Ce défi se pose en raison de la complexité morphologique potentielle de certains individus, ce qui peut rendre l'application de l'algorithme basé sur

les connaissances incertaines. En raison de ces limitations, les chercheurs ont poursuivi des approches alternatives pour compléter cette méthode et améliorer son efficacité.

3.4.1.1 Technique basée sur les fonctionnalités

Le processus consiste à identifier la région faciale et à isoler ses caractéristiques (nez, bouche, yeux, etc). Initialement, le processus fonctionne comme un classificateur, distinguant les zones faciales et non faciales. Cette technique vise à améliorer la qualité et la précision des informations faciales.

3.4.1.2 Correspondance des modèles de visage

Cette approche implique l'utilisation de modèles (Eigenface, modèle de visage 3D Active Appearance, etc) de visage qui ont été prédéterminés, dans le but d'établir des connexions entre les images d'entrée et ces modèles pour localiser et identifier les visages. L'étape suivante est la dissection du visage humain, où chaque élément crucial, comme la bouche, le nez et les contours du visage, est mis en valeur. Bien que cette méthode soit l'une des plus simples à exécuter, elle n'est pas sans limites, notamment dans la détection de visage. Néanmoins, il existe des procédures alternatives qui ont été développées pour remédier à ces insuffisances.

3.4.2 Algorithme basé sur l’apparence

Le fondement de cette approche réside dans l'utilisation d'images faciales à des fins de test, ce qui permet de former le modèle avec une plus grande efficacité. En

comparaison avec d'autres méthodes, il est considéré comme plus sophistiqué en raison de ses performances supérieures. Les méthodes basées sur l'apparence fonctionnent en identifiant des points significatifs (le contour des yeux, le nez, la bouche, etc) sur des images faciales à l'aide d'une combinaison d'analyse statique et de techniques d'apprentissage automatique (Support Vector Machine, FaceNet, etc). Cette approche est largement utilisée, notamment dans le cadre de la détection faciale, pour extraire les traits des visages.

3.4.3 Algorithme basé sur la distribution

Il existe plusieurs algorithmes qui aident à définir des classificateurs de la détection faciale, qui sont utilisés pour identifier des instances de la classe de motifs cibles à partir du motif d'image d'arrière-plan (texture, les traits géométriques, etc). Deux algorithmes courants sont l'analyse en composantes principales (Eigenfaces) et l'analyse discriminante de Fisher (Fisherfaces).

3.4.4 Réseaux de neurones

Les réseaux de neurone constituent une méthode d’apprentissage automatique venant de la façon dont notre cerveau agit. Leur capacité à modéliser les relations complexes et non linéaire. C’est un outil de choix pour les tâches de la classification, la régression, la reconnaissance des formes et etc. Il est constitué des neurones organisés en couches (une couche cachée, une couche d’entrée et une couche de sortie). Il peut apprendre à partir des données brutes (un régalage fin des hyperparamètres, un temps de

calcul important). La détection faciale, la détection d'émotions et la détection d'objets sont autant de domaines qui offrent une large gamme de résolution pour diverses opérations. Après avoir effectué une série de tests, ces technologies (YoloV5 et YoloV8) ont finalement été sélectionnées pour leur efficacité à analyser en profondeur les résultats.

3.4.5 Machines à vecteur de supports (SVM)

Typiquement, ce sont des classificateurs qui fonctionnent de manière linéaire. Leur capacité à dépasser les limites posées par les limites des exemples d'ensembles d'apprentissage et de plans de décision est remarquable.

3.4.6 Modèle de Markov caché

Le motif utilisé pour les algorithmes de détection se présente fréquemment sous la forme de traits du visage. En règle générale, il se manifeste par une ligne fine et fine ou un groupe de pixels. Ce modèle est fréquemment intégré à d'autres techniques afin de produire des algorithmes (algorithme de Viola-Jones, OpenCv Face Detector, etc) de détection efficace.

3.4.7 Approche théorique de l'information

Les champs aléatoires de Markov sont souvent utilisés pour la réalisation de modèles faciaux et leurs caractéristiques de corrélation. La discrimination entre les classes dans le processus de Markov est améliorée par la mise en œuvre de la technique de divergence Kullback-Liebler, qui est couramment utilisée dans la détection faciale.

3.4.8 Apprentissage inductif

Dans la détection de visages humains à l'aide de cette technique, plusieurs algorithmes sont utilisés, avec un accent particulier mis sur deux : C4.5 de Ross Quinlan et FIND-S de Tom M. Mitchell [35].

3.5 Étapes de la détection faciale

3.5.1 Détection de visage

L'objectif principal de la détection de visage est d'approximer la zone rectangulaire englobant le visage dans une image. Ceci est effectué pour extraire le visage et améliorer son utilité dans les tâches de détection. Pour améliorer la robustesse et la simplicité des systèmes de détection faciale, certains systèmes incluent l'alignement des visages.

3.5.2 Extraction des caractéristiques

Une fois l'étape de détection (segmentation) terminée, la prochaine étape cruciale est l'extraction des caractéristiques. C'est l'étape la plus fondamentale et essentielle de la détection faciale, où les composants physiologique (la peau du visage, les muscles faciaux, etc) du visage sont extraits. Pour s'assurer que deux individus, à l'exception des jumeaux identiques, n'est pas de combinaisons identiques de caractéristiques, diverses méthodes sont utilisées pour extraire ces caractéristiques.

3.6 Technologie de la détection de visage

L'identification de diverses technologies (détection de visage dans des images statiques, détection de visage en temps réel, etc) a conduit à une détection plus efficace et plus précise des visages humains. Ces avancées ont abouti à des processus plus fiables et sécurisés avec une plus grande précision. Les logiciels utilisés pour ces opérations possèdent souvent des fonctionnalités comparables, notamment les réseaux de neurones, MATLAB et OpenCv etc.

3.7 Conclusion

Après un aperçu concis des différentes approches de la détection faciale, les étapes séquentielles de ce processus ont été mises en évidence. Nous nous sommes ensuite concentrés sur les méthodes utilisées dans la détection des visages, avant de nous plonger dans les spécificités de la détection faciale. Le prochain chapitre portera sur la conception et réalisation qui nous conduira à parler de notre environnement de travail et des algorithmes choisi YoloV5 et YoloV8 pour effectuer la détection de visage.

Chapitre 4- CONCEPTION ET REALISATION

4.1 Introduction

Dans ce chapitre l'accent initial sera mis sur les ressources, Yolo qui est un système de détecteur temps réel d'objet dans une image utilisé à cause de sa robustesse, sa vitesse et sa précision remarquable et l'environnement utilisés pour le projet. Par la suite, ce modèle sera implémenté conformément à son architecture, et nous le formerons à l'aide d'images contenant des visages humains tirée du web scraping et des images prises par caméra. Enfin, le projet sera réalisé en utilisant Python comme langage de programmation, ainsi que diverses bibliothèques telles que Streamlit, Torch, Pil, et Opencv à des fins d'apprentissage et de classification, en plus de quelques techniques simples pour optimiser les performances du modèle.

4.2 Environnement de travail

4.2.1 Environnement matériel

Notre projet a été développé sur un ordinateur portable avec les caractéristiques : processeur CPU @1.70GHz, d'une mémoire vive de 16.0 Go, capacité disque dur 1 téra-octet sur un Windows 11 professionnel de 64 bits. Le projet a été implémenté dans Google Colab, un environnement basé sur le cloud. Cette plateforme a joué un rôle crucial en nous fournissant un GPU et une interface pour exécuter notre algorithme de détection de visages pré-entraîné. Google Colab (seulement pour l'entrainement) est particulièrement

bien adapté aux tâches d'apprentissage en profondeur car il simplifie le processus d'écriture et d'exécution du code Python via le navigateur. De plus, il offre la commodité de stocker des blocs-notes dans Google Drive et la possibilité de les charger à partir de GitHub.

4.2.2 Environnement logiciel

4.2.2.1 Présentation du langage Python

Python est un langage qui vise à aider les programmeurs à créer un code logique et précis pour des projets de petite et grande échelle, en utilisant des principes orientés objet et en offrant une plate-forme open source. Il a été créé par Guido Van Rossum au début des années 1980, et son trait distinctif est sa flexibilité. Il peut être utilisé dans un large éventail d'applications telles que le développement Web, l'intelligence artificielle, l'apprentissage automatique, les systèmes d'exploitation, le développement d'applications mobiles, les jeux vidéo et bien d'autres domaines. Le but de ce langage est d'améliorer l'efficacité des programmeurs grâce à la fourniture d'outils de haut niveau et d'une syntaxe conviviale. De plus, certains éducateurs apprécient ce langage en raison de sa séparation claire de la syntaxe des mécanismes de bas niveau. Cette fonctionnalité permet une introduction en douceur aux concepts fondamentaux de la programmation. Python est un langage de programmation doté de plusieurs bibliothèques d'apprentissage automatique (Scikit-learn, PyTorch, Keras, etc) et de puissants packages analytiques (NumPy, Matplotlib, etc). Ces facteurs nous ont obligés à choisir Python plutôt que d'autres

langages de programmation. Un autre avantage de Python est sa vaste communauté, qui offre une assistance via divers forums.

4.2.2.2 Présentation de la bibliographie Open CV

En 1999, Gary Bradsky a commencé le développement d'OpenCV pendant son mandat chez Intel. La première édition d'OpenCV a été publiée en 2000. OpenCV est une abréviation pour Open Source Computer Vision Library, et bien qu'elle soit écrite dans des versions optimisées de C et C++, il existe des interfaces supplémentaires qui sont proposées en Python, Java et C++. La communauté mondiale d'utilisateurs d'OpenCV est dynamique et en expansion. OpenCV-Python est l'API basée sur Python pour OpenCV. Il peut être conçu comme une enveloppe Python autour de la réalisation C++ d'OpenCV. Non seulement OpenCV-Python est rapide dans son exécution (grâce à sa base basée sur C/C++), mais c'est aussi un langage convivial et simple pour écrire et distribuer du code (grâce à la superposition Python qui est présente). Par conséquent, il s'agit d'une option optimale pour exécuter des programmes nécessitant une puissance de calcul importante.
Nous avons choisi OpenCV comme outil de traitement d'images en raison de ses performances et de sa polyvalence. OpenCV permettra lire les données d'entrée, prétraiter les images pour répondre aux spécifications Yolo et visualiser efficacement les résultats de détection en affichant les boîtes englobantes et des scores de confiance.

4.2.2.3 Streamlit

Une bibliothèque Python open source, Streamlit, fournit une méthode simple pour créer et diffuser des applications Web personnalisées visuellement attrayantes pour la

science des données et l'apprentissage automatique. Sa simplicité et sa rapidité dans la construction d'applications Web axées sur les données en font un outil particulièrement apprécié des scientifiques des données et des ingénieurs en apprentissage automatique [36]. Lorsque l'on parle de développement Web, le terme "session" se rapporte à la période pendant laquelle un utilisateur s'engage avec une application Web. Les informations conservées par l'application tout au long de cette interaction de l'utilisateur sont appelées état de session. Au sein de la plateforme Streamlit, l'état de session permet aux développeurs de conserver les données entre les exécutions, facilitant ainsi la création d'applications hautement interactives et fluides.

Nous avons utilisé cette bibliothèque car elle crée des visualisations interactives, facilite la configuration des paramètres de détection aux utilisateurs et fourni les résultats de la détection en temps réel.

4.2.2.4 Torch (PyTorch)

PyTorch est une bibliothèque d'apprentissage automatique qui est open source et est utilisée pour la création et la formation de modèles d'apprentissage en profondeur qui reposent sur des réseaux de neurones. Il est principalement développé par le groupe de recherche sur l'IA de Facebook. PyTorch est compatible avec C++ et Python, bien que l'interface Python soit plus raffinée. PyTorch est très apprécié dans les laboratoires de recherche et a acquis une immense popularité, soutenu par des entités importantes telles que Microsoft, Facebook... Malgré sa popularité, PyTorch n'est pas encore largement utilisé sur les serveurs de production, qui sont dominés par TensorFlow (supporté par

Google). Néanmoins, PyTorch se développe à un rythme rapide. Contrairement à d'autres cadres d'apprentissage en profondeur largement utilisés, tels que TensorFlow, qui utilisent des graphiques de calcul statiques, PyTorch utilise un calcul dynamique, offrant un degré de flexibilité plus élevé. PyTorch s'appuie fortement sur les concepts de base de Python, tels que les classes, les structures et les boucles conditionnelles, qui sont familiers à de nombreux utilisateurs et, en tant que tels, sont plus intuitifs à comprendre. Cela rend PyTorch nettement plus simple à utiliser que les cadres (frameworks) concurrents, comme TensorFlow, qui intègrent leur propre style de programmation.

Nous avons choisi PyTorch en raison de sa compatibilité transparente avec Yolo, de ses excellentes performances GPU et de sa polyvalence, qui vont permettre d'adapter et d'améliorer le modèle en fonction des exigences uniques de notre projet.

4.2.2.5 PIL (Python Image Library)

La Python Imaging Library, une extension de PIL, est le progiciel leader pour le traitement d'images dans le langage Python. Il fournit une large gamme d'outils pour le traitement d'image léger, y compris la possibilité de créer, de modifier et de stocker facilement des images. Bien que le soutien au projet PIL original ait cessé en 2011, un nouveau projet appelé Pillow est apparu comme un fork du projet original. Pillow intègre la prise en charge de Python3.x et a été présenté comme un remplacement de PIL à l'avenir. Cette bibliothèque propose une vaste gamme de formats de fichiers image, notamment BMP, PNG, JPEG et TIFF. De plus, la bibliothèque encourage activement les utilisateurs

à étendre ses capacités en créant de nouveaux décodeurs de fichiers pour les formats émergents.

Nous avons utilisé cette bibliothèque pour améliorer la précision et réduire le temps de traitement de la détection. Contrairement à OpenCV, PIL est très simple a utilisé, utile pour les opérations web basiques.

4.3 Deep Learning

Le sous-domaine de l'intelligence artificielle connu sous le nom d'apprentissage en profondeur utilise des réseaux de neurones pour relever des défis complexes au moyen d'architectures complexes comprenant diverses transformations non linéaires. Grâce à l'utilisation de ces techniques, des progrès notables et rapides ont été réalisés dans les domaines de l'analyse des signaux auditifs ou visuels. Plus précisément, l'apprentissage en profondeur a eu un impact considérable sur l'identification faciale, l'identification vocale, et la compréhension linguistique automatisée.

4.3.1 Présentation de YoloV5

YoloV5 est un algorithme conçu pour la détection d'objets en temps réel. Son but est d'identifier avec précision des objets dans un environnement donné. Dans le domaine des médias visuels, il existe une catégorie d'objets qui ont une signification unique. Ces objets font partie intégrante des vidéos, des flux en direct ou des images, contribuant à leur composition et à leurs impacts globaux. PyTorch, un framework d'apprentissage en profondeur disponible en tant que logiciel open source, a été utilisé dans le développement

de cet algorithme. Ce cadre a permis de former et de tester des ensembles de données personnalisés, et il est connu pour ses excellentes performances de détection. En 2020, l'équipe d'Ultralytics LLC a présenté le YoloV5, un modèle de réseau de détection de cible avancé qui surpasse son prédécesseur (YoloV4). En tant que modèle fonctionnant en une seule étape, YoloV5 effectue la tâche de classification et la mise en œuvre de la régression par boîte englobante (une technique utilisée dans les tâches de vision par ordinateur) en une étape améliorant considérablement la précision et l'efficacité du processus. Plus rapide que la plupart des réseaux de neurones comme le Faster R-CNN par exemple, ce qui en fait l'un des modèles de détection de cible les plus appréciés. Le modèle YoloV5 offre une solution pour réduire les besoins de calcul en éliminant l'étape de proposition de région et en redéfinissant la détection d'objets comme une tâche de régression dense (consiste à faire une prédiction pour chaque pixel). YoloV5 possède une vitesse d'inférence d'image impressionnante de 455 images par seconde (FPS), ce qui en a fait un choix populaire parmi de nombreux chercheurs en raison de cet avantage remarquable.

4.3.1.1 Architecture de YoloV5

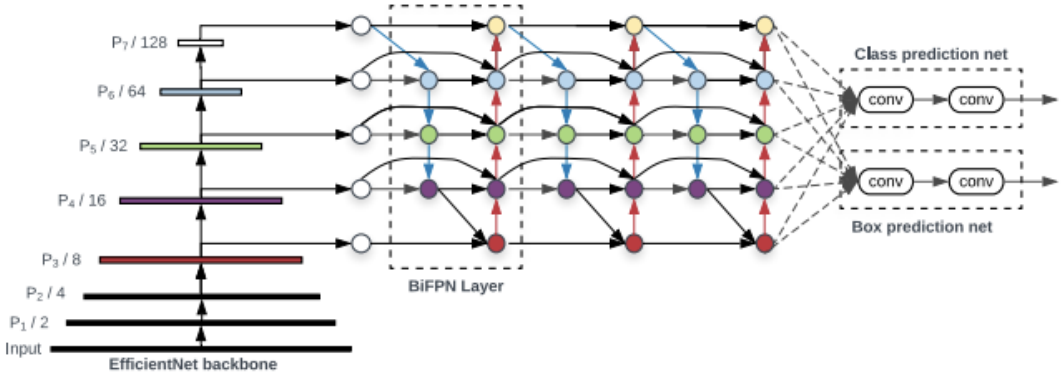

Figure 4-1 Architecture de YoloV5 [38]

La figure 4-1 présente l'architecture du réseau YoloV5 [38]. Elle se compose de trois parties principales :

- EfficientNet Backbone (épine dorsale) pour l'extraction de caractéristiques. Un réseau neuronal de nature convolutive est capable de rassembler et de modeler des caractéristiques d'image à différents niveaux de détail.
- Bi-FPN layer (cou) pour la fusion de caractéristiques (nez, bouche, yeux, etc). Le processus implique une séquence de niveaux qui facilitent le mélange et la fusion de diverses caractéristiques d'image afin de les préparer pour l'analyse et la prévision.
- Box prediction net (tête) produit les résultats de détection (score, emplacement, classe, taille.

YoloV5 utilise une approche CNN pour faire la détection en incorporant ses couches telles que les couches de détection, les couches de classification et les couches de régression.

4.3.1.2 Particularité de YoloV5

YoloV5 a été méticuleusement conçu pour surpasser ses itérations précédentes en termes de vitesse et de précision. En incorporant des techniques de vision par ordinateur de pointe, il est capable d'identifier rapidement et avec précision des objets en temps réel. De plus, sa capacité à détecter plusieurs objets simultanément en fait un atout inestimable pour la détection d'objets dans des images complexes. Outre sa vitesse et sa précision remarquables, la polyvalence et la convivialité de YoloV5 ne doivent pas être négligées. Il peut être formé sur des ensembles de données personnalisés, permettant aux utilisateurs de l'adapter à leurs besoins spécifiques. De plus, en tant que solution open source, les individus ont accès au code source, ce qui leur permet de le modifier en fonction de leurs besoins uniques. Le système YoloV5 possède un degré élevé d'évolutivité, lui permettant d'identifier efficacement des objets dans des images. De plus, il a la capacité de détecter des objets dans des images dynamiques en mouvement, ce qui le rend adapté aux applications en temps réel impliquant le suivi d'objets. Enfin, sa compatibilité avec plusieurs langages de programmation simplifie le processus d'intégration dans une gamme variée de projet.

4.3.2 Présentation de YoloV8

YoloV8 est la dernière version de Yolo lancé le 10 janvier 2023. S'appuyant sur les réalisations de ses prédécesseurs, le modèle YoloV8 de pointe intègre de nouveaux composants et améliorations pour améliorer ses capacités (une meilleure gestion des objets multi-échelle, des algorithmes de suppression plus intelligents, es techniques de

segmentation, etc), sa polyvalence et son efficacité. YoloV8 offre une assistance étendue pour un large éventail de projets d'IA visuelle, notamment l'identification, la division, l'estimation de la posture, la surveillance et la catégorisation. Cette extraordinaire adaptabilité permet aux utilisateurs d'exploiter tout le potentiel de YoloV8 dans diverses industries et applications. Basé sur les dernières avancées en matière d'apprentissage profond et de vision par ordinateur, YoloV8 offre une vitesse et une précision inégalées. Son architecture attrayante à bien des égards permet une utilisation polyvalente sur une gamme d'applications et une intégration sans effort avec diverses plates-formes matérielles, allant des appareils de périphérie aux API cloud. Une large sélection de modèles est disponible dans la série YoloV8, chacun étant adapté à des missions spécifiques de vision par ordinateur. Ces modèles ont été méticuleusement conçus pour répondre à une variété de besoins, allant de la détection d'objets de base à des tâches complexes telles que la segmentation d'instances, la détection de poses/points clés et la classification. La série YoloV8 comprend plusieurs composantes, chacune adaptée à des tâches spécifiques, garantissant une précision et des performances exceptionnelles. De plus, ces modèles s'intègrent parfaitement à divers modes opérationnels tels que l'inférence, la validation, la formation et l'exportation, permettant une utilisation sans effort tout au long des différentes étapes de déploiement et de développements. Utilisant des architectures de base et de Bi-FPN layer (cou) de pointe, YoloV8 améliore l'extraction de caractéristiques et les performances de détection. S'écartant des méthodes basées sur l'ancrage, YoloV8 implémente une tête (head) Ultralytics divisée, ce qui entraîne une précision améliorée et un processus de détection plus efficace. En donnant la priorité à

l'équilibre délicat entre vitesse et précision, YoloV8 s'avère être une solution polyvalente pour la détection d'objets dans divers domaines. De plus, YoloV8 propose une sélection de modèles pré-entraînés adaptés à différentes tâches et critères de performance, garantissant ainsi une recherche facile du modèle parfait pour vos besoins spécifiques.

À la pointe des avancées technologiques se trouve YoloV8. Bien que YoloV5 se vante de vitesse, de simplicité et de précision, il n'a jamais revendiqué le titre de meilleur au monde dans son domaine. Cependant, YoloV8 révolutionne quand même le domaine en surpassant tous les autres modèles existants tant en termes de vitesse que de précision. Le processus de configuration de YoloV8 est encore plus simple. Dans le cas de YoloV5, les utilisateurs devaient cloner manuellement le répertoire et configurer leur environnement. Bien que cette option existe toujours pour ceux qui préfèrent une approche pratique, l'installation de YoloV8 en tant que package pip permet un point de départ beaucoup plus simple. L'évolution de YoloV8 l'a transformé en une plateforme polyvalente. S'écartant d'une architecture singulière, YoloV8 s'adapte désormais à toutes les itérations de Yolo, même celles développées par des concurrents.

Dans le cadre YoloV8, il existe une gamme de modèles répondant à divers besoins de détection, de segmentation et de classification. Ces modèles peuvent être classés en cinq types distincts. Le YoloV8 Nano se distingue comme l’option la plus compacte et la plus efficace, offrant des résultats rapides. À l’autre extrémité du spectre, le YoloV8 Extra Large (YoloV8x) remporte la couronne pour être le modèle le plus lent mais aussi le plus précis de la gamme. YoloV8 comprend des modèles pré-entraînés qui englobent diverses fonctionnalités. Ces modèles se composent de points de contrôle de détection d'objets qui

ont été formés sur l'ensemble de données de détection Coco, en utilisant une résolution d'image de 640 X 640. De plus, il existe des points de contrôle de segmentation d'instance qui ont suivi une formation sur l'ensemble de données de détection Coco, avec un accent particulier sur l'analyse des données de segmentation Coco, toujours avec une résolution d'image de 640 X 640. De plus, YoloV8 propose des modèles de classification d'images qui ont été pré-entraînés sur l'ensemble de données ImageNet, en utilisant une résolution d'image de 224 X 224.

4.3.2.1 Architecture de YoloV8

Les utilisateurs ont la liberté d'adapter sans effort la structure et les paramètres de l'architecture YOLOV8 à leurs besoins spécifiques, la rendant hautement personnalisable. Tel que présenté dans la figure 4-3, les principaux composants de YOLOV8 sont constitués d'un réseau neuronal convolutif, qui peut être classé en plusieurs catégories : la colonne vertébrale et la tête.

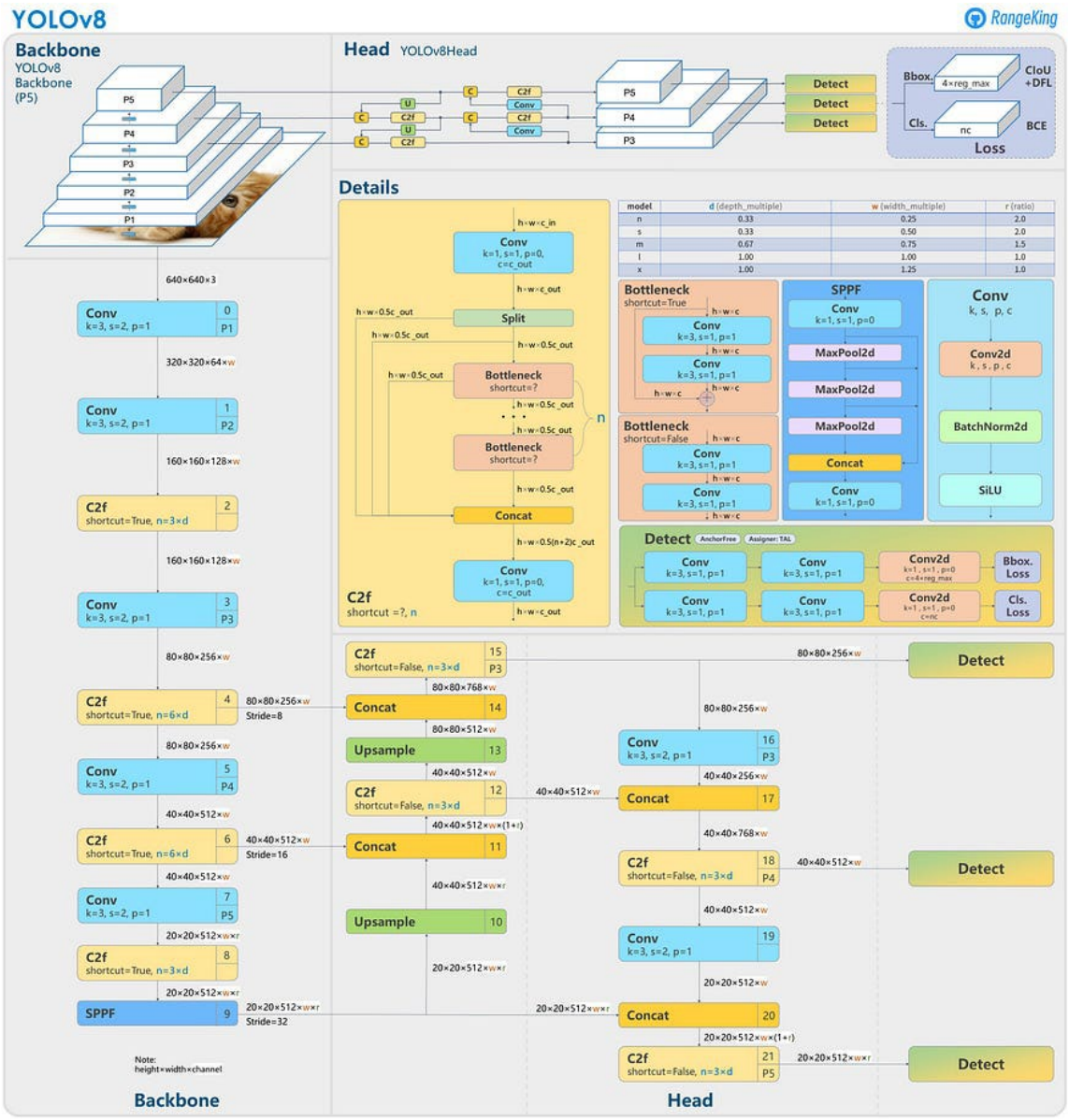

Figure 4-2 Architecture de YoloV8 [39]

La figure 4-2 présente l'architecture du réseau YoloV8 [39]. Elle se compose de deux parties principales :

- Head (tête) : comprend plusieurs couches convolutives, qui sont ensuite suivies d'une séquence de couches entièrement connectées.
- Backbone : pour l'extraction de caractéristiques. Un réseau neuronal de nature convolutive et capable de rassembler et de modeler des caractéristiques d'image à différents niveaux de détail.

4.3.2.2 Particularité de YoloV8

Le niveau de précision présenté par YoloV8 est vraiment impressionnant. Les développeurs ont la possibilité d'utiliser les nombreuses fonctionnalités pratiques qui accompagnent YoloV8, notamment son interface de ligne de commande facilement navigable et son package Python méticuleusement organisé.

Le domaine de la vision par ordinateur abrite une communauté dynamique et croissante centrée autour de Yolo et de sa dernière version, YoloV8. En conséquence, il existe une pléthore de personnes connaissant bien ce domaine, facilement disponibles pour offrir des conseils et un soutien si nécessaire. L'importance de YoloV8 réside dans ses caractéristiques remarquables. Contrairement à d'autres modèles qui répartissent les tâches sur plusieurs fichiers Python pour son exécution, YoloV8 simplifie le processus de formation en offrant une interface de ligne de commande. Cette fonctionnalité améliore l'expérience utilisateur en offrant une approche plus intuitive de la formation des modèles. De plus, YoloV8 est accompagné d'un package Python qui garantit une expérience de codage plus fluide par rapport aux modèles précédents. YoloV8 possède une capacité significative dans la détection d'objets à différentes échelles.

4.4 Comparaison entre YoloV5 et YoloV8

A travers les mises à jour apportées à YoloV8 qui est une version améliorée de YoloV5, YoloV8 donne une meilleure performance, vu les améliorations apportées sur son architecture, ses techniques d'entrainement et sur sa précision.

YoloV5 et YoloV8 se distinguent par leur efficacité dans la détection et la création de régions d'intérêt (ROI). Réputé pour sa rapidité et sa conception légère, YoloV5 génère efficacement des cadres de délimitation autour des objets identifiés, atteignant une précision mAP@0,5 à plus de 70 %, ce qui le rend idéal pour les tâches de détection rapides et simples. Néanmoins, sa limitation aux ROI rectangulaires peut ne pas répondre aux besoins des applications qui nécessitent une localisation précise. En revanche, YoloV8 bénéficie d'une précision améliorée (mAP@0,5 à plus de 80 %) et intègre des fonctionnalités avancées telles que la segmentation native, permettant la création de masques ROI précis qui sont particulièrement bénéfiques pour les objets complexes ou imbriqués. Bien qu'il soit légèrement plus exigeant en ressources, YoloV8 offre une polyvalence accrue grâce à des optimisations conçues pour les tâches de détection multi-échelles, ce qui permet d'obtenir des ROIs plus détaillés qui conviennent parfaitement aux applications modernes.

4.5 Conception et implémentation

4.5.1 Description de l'approche

Après avoir préparé et trié notre base de données d'images, nous avons décidé d'adopter l'approche Yolo pour entraîner notre modèle de détection faciale. Yolo, qui

signifie "You Only Look Once", est une architecture populaire pour la détection d'objets en temps réel, et ses dernières versions sont particulièrement performantes et efficaces.

Avant d'entraîner notre modèle, nous avons dû annoter soigneusement (entourant les zones d'intérêts en format Yolo qui est le format txt) les images de notre base de données avec labelImg un outil open source qui crée des bounding boxes."Afin de standardiser les coordonnées, nous extrayons les valeurs de pixels de x et y, qui servent de points de référence pour le centre du cadre de délimitation le long des axes x et y. Par la suite, nous normalisons la valeur x en la divisant par la largeur de l'image, et la valeur y en la divisant par la hauteur de l'image. Cela nous permet de représenter avec précision les dimensions du cadre de délimitation. Cela signifie que nous avons identifié et marqué les visages sur chaque image pour que notre modèle puisse apprendre à les reconnaître. Les ROIs sont traités automatiquement grâce au pipeline de Yolo intégré dans l'entrainement.

Une fois que nos modèles Yolo sont entraînés, nous avons utilisé Streamlit pour le déployer. Streamlit est un cadre de travail open-source qui permet de créer rapidement des applications Web destinées à la science des données. Ainsi, grâce à Streamlit, nos modèles de détection faciale basé sur Yolo ont pu être mis en œuvre de manière interactive et conviviale. Cette combinaison a permis à notre système de reconnaître et d'identifier les visages de manière efficace et en temps réel. Nous avons adopté YoloV5 et YoloV8 en raison de leur rapidité et de leur efficacité pour la détection d'objets, ce qui est crucial pour notre application en variant les paramètres de chaque modèle et en comparant les résultats. Yolo a été conçu pour détecter les objets et leur emplacement dans une image avec une grande précision, ce qui en fait un excellent choix pour notre projet.

4.5.2 Préparation des données

YoloV5 et YoloV8 ont la même préparation de données. Nous avons entrainé les modèles YoloV5 et YoloV8 avec les mêmes paramètres ensuite nous avons comparé les résultats de détections avec les modèles entrainés afin de déterminer entre YoloV5 et YoloV8 lequel est plus performant avec une meilleure précision de détection. Voici une série d'étapes générales que nous avons suivies pour préparer nos données pour l'entraînement de nos modèles.

4.5.2.1 Collecte de données

Premièrement, cette tâche consiste à rassembler un ensemble d'images de visages. En l'absence de bases de données publiques accessibles sur Internet, et nous avons choisi d'opter pour le web scraping et certaines images à partir d'une caméra. il a été utilisé des images variées et une base de données avec une taille d'image de 230 images; cela a permis d'augmenter la précision, la performance et la robustesse des modèles.

4.5.2.2 Augmentation des données

YoloV5 et YoloV8 soutiennent une variété de stratégies d'augmentation de données pour favoriser l'optimisation des résultats de détection, notamment l'inversion horizontale, le découpage aléatoire, l'ajout aléatoire de bruit coloré, entre autres données.
L'augmentation se fait automatiquement lors de l'entraînement avec les paramètres présentés dans la figure 4-3.

```
"fl_gamma": (False, 0.0, 2.0),  # focal loss gamma (efficientDet default gamma=1.5)
"hsv_h": (True, 0.0, 0.1),  # image HSV-Hue augmentation (fraction)
"hsv_s": (True, 0.0, 0.9),  # image HSV-Saturation augmentation (fraction)
"hsv_v": (True, 0.0, 0.9),  # image HSV-Value augmentation (fraction)
"degrees": (True, 0.0, 45.0),  # image rotation (+/- deg)
"translate": (True, 0.0, 0.9),  # image translation (+/- fraction)
"scale": (True, 0.0, 0.9),  # image scale (+/- gain)
"shear": (True, 0.0, 10.0),  # image shear (+/- deg)
"perspective": (True, 0.0, 0.001),  # image perspective (+/- fraction), range 0-0.001
"flipud": (True, 0.0, 1.0),  # image flip up-down (probability)
"fliplr": (True, 0.0, 1.0),  # image flip left-right (probability)
"mosaic": (True, 0.0, 1.0),  # image mixup (probability)
"mixup": (True, 0.0, 1.0),  # image mixup (probability)
"copy_paste": (True, 0.0, 1.0),
```

Figure 4-3 Paramètres utilisés pour l'augmentation des données

- fl_gamma : permet d'équilibrer les classes qui seront déséquilibrées.
- hsv_h, hsv_s, hsv_v : permettent d'améliorer et d'augmenter les couleurs présentes dans les images.
- degrees, flipud, fliph : permettent d'améliorer et d'augmenter l'orientation des objets.
- translate : permet d'améliorer et d'augmenter les positions des objets.
- scale : permet d'améliorer et d'augmenter les tailles des objets.
- shear : permet d'améliorer et d'augmenter les formes des objets.
- perspective : permet d'améliorer et d'augmenter les perspectives des objets.
- mosaic, mixup, copy_paste : permet d'améliorer et d'augmente la variété des images et des objets.

L’utilisation de HSV améliore la fiabilité et la précision de la détection en minimisant les effets des variations d’éclairage et de couleur.

Figure 4-4 Exemple d'augmentation d'image

4.5.2.3 Annotation des données

Il est nécessaire d'annoter les images pour délimiter la localisation des visages. Dans une image comme celle présentée à la figure 4-5, chaque visage doit être circonscrit par une boîte englobante et recevoir une étiquette appropriée. Habituellement, ces annotations sont sauvegardées dans des fichiers texte distincts, ou dans un fichier CSV ou JSON. Il existe plusieurs outils d'annotation d'images à notre disposition, tels que

LabelImg, VGG Image Annotator (VIA), ou encore le Visual Object Tagging Tool (VoTT) de Microsoft. Cependant, dans notre situation, nous avons privilégié l'utilisation de **LabelImg** pour sa simplicité d'usage.

Figure 4-5 Image à annoter

4.5.2.4 Format des annotations

Pour YoloV5 et YoloV8, nos annotations doivent respecter le format Yolo, ce qui implique que chaque boîte englobante soit définie par cinq valeurs distinctes :

- L'identifiant de classe (pour ce contexte, nous avons une seule classe, ainsi l'identifiant serait 0).

- Les coordonnées du centre de la boîte englobante sur l'axe X, normalisées par rapport à la largeur de l'image.
- Les coordonnées du centre de la boîte englobante sur l'axe Y, normalisées par rapport à la hauteur de l'image.
- La largeur de la boîte englobante, normalisée en fonction de la largeur de l'image.
- La hauteur de la boîte englobante, normalisée en fonction de la hauteur de l'image.

Ces valeurs sont normalisées en divisant par la largeur et la hauteur de l'image, respectivement, de sorte qu'elles se situent entre 0 et 1. Ces cinq valeurs doivent être séparées par des espaces et conservées dans un fichier texte correspondant à chaque image. Ce fichier texte devrait porter le même nom que l'image associée et être placé dans le même répertoire.

Organisation des fichiers : Nous avons structuré nos images et fichiers d'annotation selon une organisation spécifique de dossiers :/dataset, /images, /train, /valid, /labels.

Les images et les fichiers d'annotation pour l'ensemble d'entraînement vont dans les dossiers train et ceux pour l'ensemble de validation vont dans les dossiers val (**dataset/images/valid**).

Fichier de configuration YAML : YoloV5 et YoloV8 emploient un fichier YAML pour définir les chemins vers les ensembles d'entraînement et de validation. Nous avons donc créé un fichier YAML qui se présente de la manière suivante :

train: /dataset/images/train permet au modèle d'utiliser ces images pour détecter.

val: /dataset/images/valid permet au modèle d'évaluer sa performance lors de l'entrainement.

nc: 1 nombre des classes.

names: ['face'] liste des noms des classes ici on a la classe face.

Un fichier data.yamlfichier est créé dans le répertoire racine de notre ensemble de données, et un fichier data.yamlfichier qui décrit l'ensemble de données, les classes et d'autres informations nécessaires.

Après avoir complété ces étapes, nos données sont désormais préparées et prêtes à être utilisées pour l'entraînement de notre modèle de détection de visages.

Les figures 4-6, 4-7 et 4-8 permettent de visualiser les résultats d'annotation des trois visages présents.

Figure 4-6 Image annotée Visage 1

Figure 4-7 Image annotée Visage 2

Figure 4-8 Image annotée Visage 3

4.5.3 Entraînement du modèle

Dès que nos données sont correctement préparées, nous sommes en mesure de lancer l'entraînement de notre modèle de détection de visages basé sur YoloV5 et YoloV8. Voici les étapes fondamentales à suivre :

4.5.3.1 Installer les dépendances

Nous nous sommes assurés d'avoir installé toutes les dépendances nécessaires pour exécuter YoloV5 et YoloV8. Celles-ci comprennent Python, PyTorch et d'autres paquets Python tels que numpy et matplotlib. Ces dépendances ont été installées avec pip ou conda.

4.5.3.2 Cloner le dépôt YOLOv5

Ensuite, le code source de YoloV5 et YoloV8 a été cloné à partir du dépôt GitHub. Voici comment faire ce clonage en ligne de commande :

git clone https://github.com/ultralytics/yolov5.git

cd yolov5

pip install -r requirements.txt

https://github.com /yolov8

cd yolov8

pip install -r requirements.txt

4.5.3.3 Configurer le fichier YAML

Avant de commencer l'entraînement, nous avons configuré le fichier YAML pour notre modèle. Il y a deux fichiers YAML importants à noter.

Le premier de ces fichiers YAML de notre modèle, spécifie l'architecture du modèle que nous utilisons. YoloV5 et YoloV8 comprennent plusieurs architectures de modèles par défaut ou variantes (YoloV5 : YoloV5s, YoloV5m, YoloV5l, et YoloV5x, YoloV8 : YoloV8m, YoloV8l et YoloV8x). Nous avons choisi l'un d'entre eux en fonction de la précision et de la vitesse d'exécution que nous voulons pour nos modèles (Yolov5l et YoloV8l).

Le deuxième est le fichier YAML de nos données, qui indique à YoloV5 ou à YoloV8 où trouver nos données d'entraînement et de validation.

4.5.3.4 Surveillance avec l'entraînement

Pendant l'entraînement, YoloV5 et YoloV8 afficheront des informations sur les performances de notre modèle. Nous pouvons voir les courbes de perte, les précisions et d'autres métriques.

4.5.3.5 Évaluation du modèle

Après l'entraînement, les performances de notre modèle sont évaluées. YoloV5 et YoloV8 fournissent des mesures d'évaluation comme la précision, le rappel, le score F1 et la moyenne de la précision moyenne (mAP) pour évaluer le modèle.

4.5.3.6 Utilisation du modèle entraîné

Une fois l'entraînement achevé, le modèle est prêt à être utilisé pour la détection de visages. Les poids de notre modèle, une fois formé, sont enregistrés dans le répertoire runs/train/. Dans notre situation spécifique, nous avons choisi de former notre modèle sur Colab pour bénéficier d'une meilleure capacité de calcul et de rapidité. Par la suite, nous avons récupéré le modèle entraîné localement via téléchargement direct ou sauvegarder sur google drive pour procéder à la détection stockée dans un fichier .pt.

4.5.3.7 Chargement du modèle entraîné

Nous avons chargé notre modèle entraîné. Nous avons utilisé la fonction (detect.py) de chargement fourni par Yolo. Nous aurons besoin du chemin du fichier de poids de notre modèle runs/train/exp/weights/best.pt qui est le chemin par défaut.

4.5.3.8 Préparation de l'image

Préparez les images sur lesquelles nous souhaitons effectuer la détection. Nous avons récupéré ses images via le web scraping. L'automatisation de la collecte d'images via le web scraping a permis la création d'un ensemble de données diversifié et étendu pour la formation du modèle de détection. Cela peut impliquer de charger l'image dans le programme et de la prétraiter si nécessaire.

4.5.3.9 Détection d'objets

Maintenant que nos modèles sont entraînés et évalués, nous pouvons les utiliser pour détecter des objets dans de nouvelles images. Pour cela, nous devons exécuter la commande de détection sur notre image avec notre modèle entraîné.

4.5.3.10 Exécution de la détection

Nous avons utilisé le modèle pour effectuer la détection sur notre image ou vidéo. Nous pouvons le faire en utilisant une fonction de détection fournie par Yolo (detect.py) spécialement pour YoloV5 et (‘path/to/best.pt’) spécialement pour YoloV8 avec le chemin des images ou vidéos. Cela générera des résultats de détection comme ceux présentés à la figure 4-9.

Figure 4-9 Résultats de la détection

4.5.3.11 Analyse des résultats

Nous avons examiné les résultats de la détection."À partir de ces résultats nous pouvons affiner notre modèle si nécessaire, en ajustant les paramètres ou en fournissant plus de données d'entraînement. Pour évaluer si un modèle est bien entraîné, plusieurs métriques sont utilisées :

- Accuracy (Précision globale) : Le taux de précision est déterminé en comparant le nombre de prédictions correctes au nombre total de prédictions effectuées. Ce calcul consiste à additionner les résultats positifs et les résultats négatifs, puis à diviser cette somme par le nombre total de cas.
- Matrice de confusion : Cet outil permet d'évaluer visuellement l'efficacité d'un modèle en présentant des vrais positifs, des vrais négatifs, des faux positifs et des faux négatifs. Le but de la matrice est d'identifier des scénarios particuliers dans lesquels le modèle génère des résultats inexacts.
- Recall (Rappel) : La précision du modèle dans l'identification correcte des vrais positifs est mesurée par cette métrique. Cela s'avère particulièrement utile dans les situations où les conséquences des faux négatifs sont importantes.
- Precision (Précision) : La mesure en question est cruciale dans les situations où les conséquences des résultats faussement positifs sont importantes, car elle révèle la proportion de prédictions exactes parmi celles jugées positives.
- F1-scores : Lorsqu'on recherche un équilibre harmonieux entre précision et rappel, la mesure consolidée de leur moyenne harmonique s'avère être précieuse.

F1 = 2 * Precision*Rappel / Precision+Rappel [44].

4.6 Conclusion

Ce chapitre a permis de présenter les environnement (matériel et logiciels) qui ont permis à la réalisation, la présentation de YoloV5 et YoloV8, les étapes de la conception et de l'implémentation. En fin de compte, notre système de détection faciale a été conçu en utilisant l'architecture de YoloV5 et YoloV8 et en mettant en œuvre des techniques d'apprentissage en profondeur. Dans le prochain chapitre portera sur la présentation des tests et des résultats ainsi que l'analyse des résultats obtenus ainsi que leurs analyses.

Chapitre 5- RÉSULTATS

5.1 Introduction

Cette application utilise les modèles entraînés YoloV5 et YoloV8 personnalisés pour détecter les visages dans les images téléchargées par l'utilisateur via l'interface utilisateur. Les sections suivantes décrivent en détail le développement de l'interface, le chargement du modèle, le chargement des images, l'exécution du modèle sur les images et la présentation des résultats ainsi que leurs analyses.

5.2 Importation des bibliothèques

Pour pouvoir exécuter le modèle nous avons besoin d'importer les bibliothèques. Le code importe les bibliothèques nécessaires pour la manipulation d'images, la détection de visages et l'interface utilisateur voire figure 5-1 :

streamlit : pour créer l'interface utilisateur de l'application web.

Torch : pour travailler avec le modèle YOLOV5.

PIL : pour ouvrir et manipuler des images.

cv2 : Cette bibliothèque permet de traiter des images et des vidéos.

PIL : Cette bibliothèque permet de traiter des images.

tempfile : Cette bibliothèque permet de créer des fichiers temporaires.

ultralytics : Cette bibliothèque contient le code de YoloV8.

numpy : Cette bibliothèque permet de manipuler des tableaux de données.

pandas : Cette bibliothèque permet de manipuler des données tabulaires.

os : Cette bibliothèque permet d'interagir avec le système d'exploitation (utiliser pour la gestion des fichiers et de répertoire)

```
import streamlit as st
import cv2
from PIL import Image
import tempfile
from ultralytics import YOLO
import numpy as np
import pandas as pd
import os
from ultralytics.utils.plotting import Annotator, colors
import datetime
import torch
from PIL import Image, ImageDraw, ImageFont
```

Figure 5-1 Importation des bibliothèques

5.3 Développement de l'interface

Le développement de l'interface est pratiquement le même avec YoloV8. Le développement est décomposé en trois parties (le chargement du modèle, le chargement des images, l'exécution du modèle sur les images) voir respectivement la figure 5-2, la figure 5-3 et la figure 5-4.

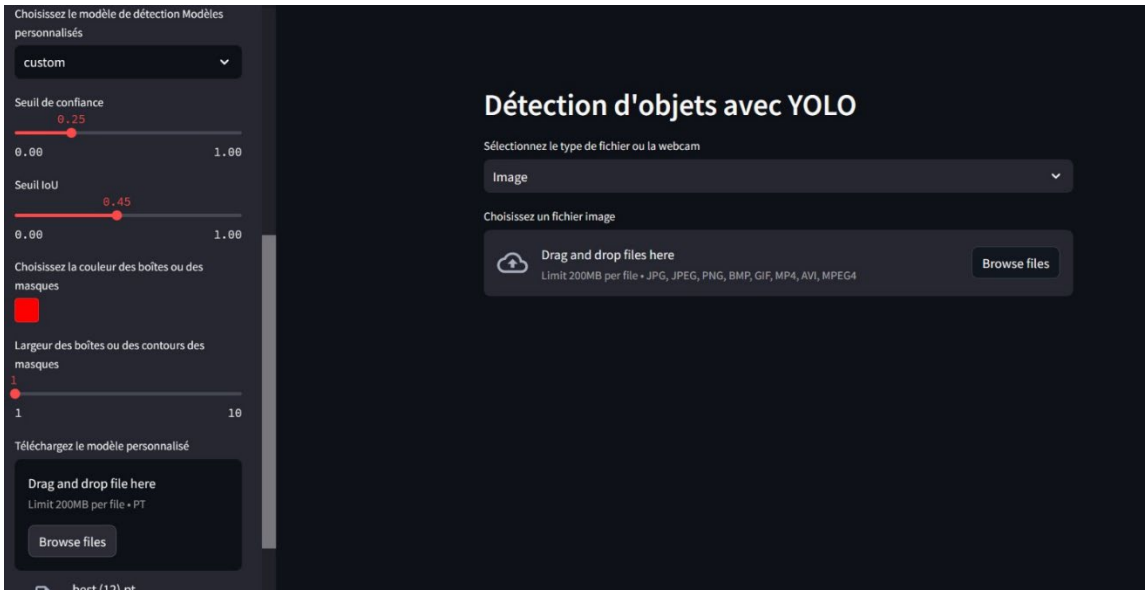

Figure 5-2 Interface de l'application

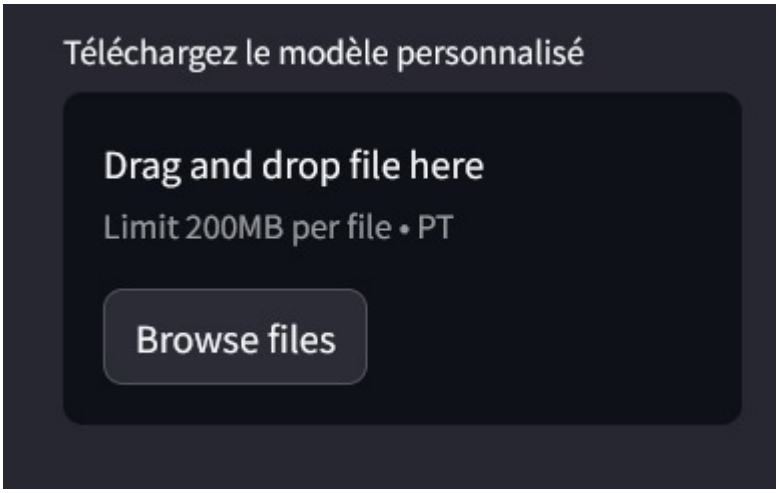

Figure 5-3 Chargement du modèle

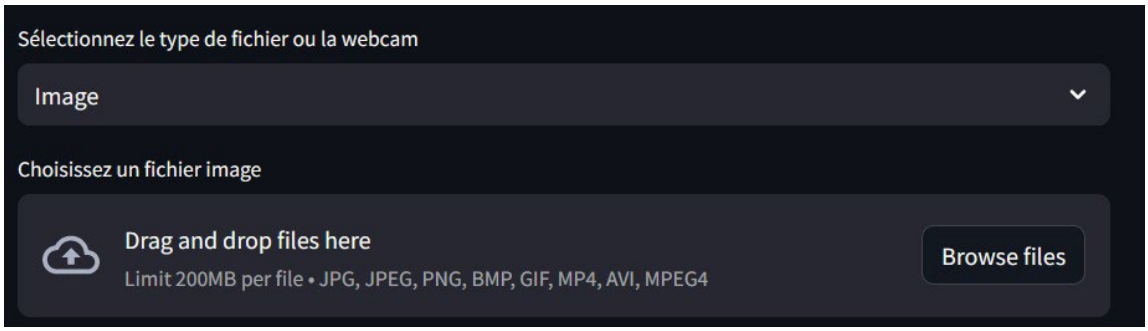

Figure 5-4 Chargement des images ou webcam

La figure 5-4 montre l'interface de chargement de l'image affichant les différents types de format acceptés.

Le développement de l'interface utilisateur est une étape cruciale pour rendre l'application accessible et conviviale. La bibliothèque Streamlit est utilisée pour créer une interface web intuitive. Le code commence par définir le titre de la page avec la commande st.title, puis il crée un bouton permettant aux utilisateurs de télécharger des images en utilisant st.file_uploader. Les utilisateurs peuvent télécharger des images aux formats jpg, jpeg ou png etc. Un bouton est créé pour chaque image téléchargée, permettant à l'utilisateur d'initier la détection de visages pour cette image spécifique.

5.4 Chargement des images

Figure 5-5 Image test 1

Figure 5-6 Image test 2

Une fois que l'utilisateur a téléchargé une ou plusieurs images, le code utilise la bibliothèque PIL pour lire et afficher chaque fichier image. Ces images sont ensuite affichées sur la page web avec la fonction st.image, montrant à l'utilisateur l'aperçu de l'image originale avant la détection. Voire un exemple d'image test dans la figure 5-6.

5.5 Exécution du modèle

Pour la détection de visages, le code utilise un modèle YoloV5 ou YoloV8 personnalisé. La fonction load_model est utilisée pour charger le modèle à partir d'un chemin à spécifié. Ce chemin est ensuite appliqué sur l'image pour détecter les visages. Pour ce faire l'utilisateur applique le bouton (voir figure 5.7) de détection sur l'image

Figure 5-7 Bouton détecter visages

Lorsque l'utilisateur clique sur le bouton (voir figure 5-7) de détection pour une image donnée, la fonction run_detection est appelée avec l'image (comme l'exemple de l'image result.png) et le modèle chargé comme arguments (confidence threshold, image size, batch size, class filter, etc). Cette fonction utilise le modèle YoloV5 ou YoloV8 personnalisé pour détecter les visages dans l'image.

5.6 Présentation du résultat 1

On observe les résultats de détection des visages dans l'image test1 (5-8 et 5-9) et test2 (voir figure 5-10 et 5-11) et le nombre de visages sur l'image.

5.6.1 Résultat de détection de YoloV5 et YoloV8

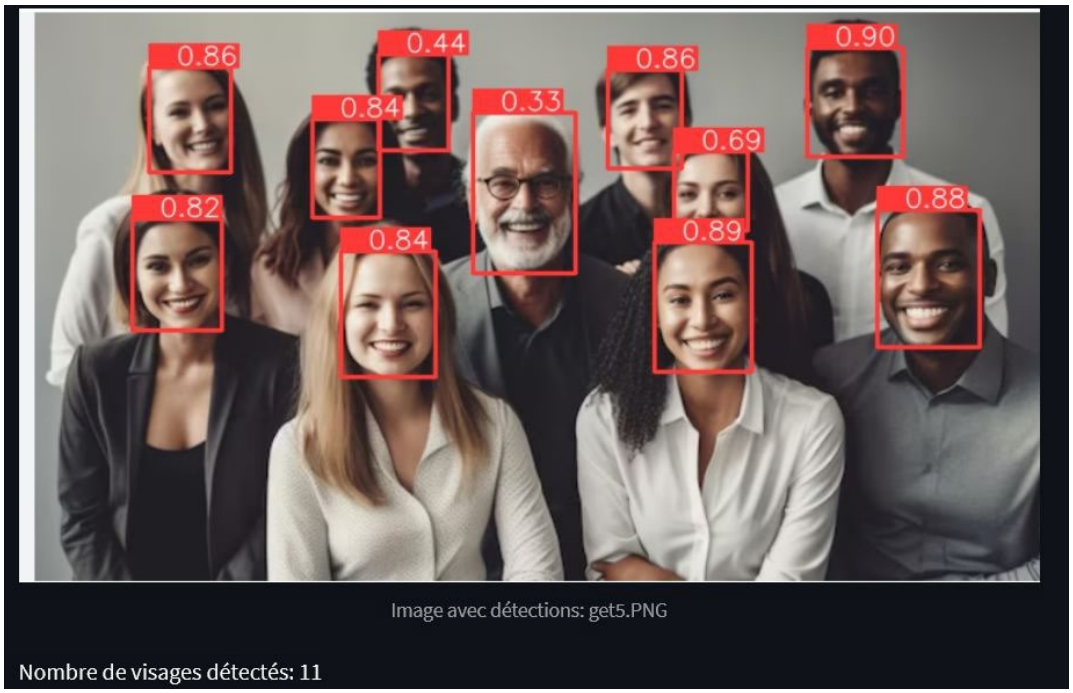

Figure 5-8 Image test 1 résultat YoloV5

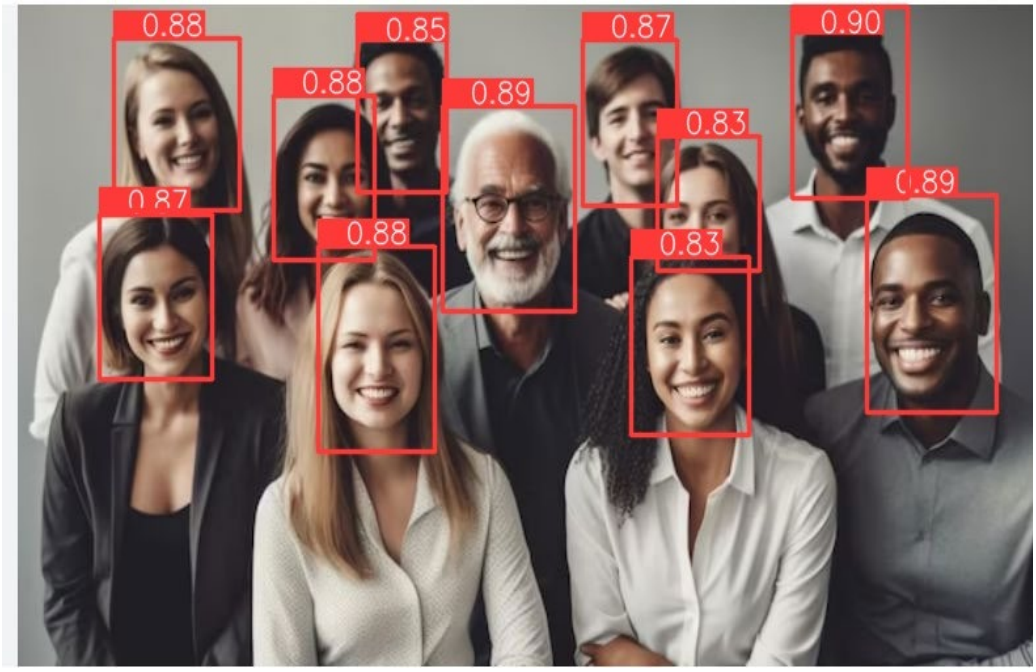

Figure 5-9 Image test 1 résultat YoloV8

Figure 5-10 Image test 2 résultat YoloV5

Figure 5-11 Image test 2 résultat YoloV8

Après l'exécution du modèle, l'image résultante comporte des boîtes englobantes autour des visages détectés et sont affichés à l'utilisateur avec la fonction st.imagel; le nombre de visages détectés est également calculé et affiché à l'utilisateur avec la commande st.write. Les résultats sont présentés d'une manière qui permet à l'utilisateur de comprendre clairement ce qui a été détecté et où dans l'image. Cela fait de cette application un outil puissant et accessible pour la détection de visages, que ce soit pour des professionnels de la vision par ordinateur ou des utilisateurs occasionnels intéressés par la technologie.

5.6.2 Discussion

Nous avons comparé les résultats de détecteur de visage des modèles YoloV5 et YoloV8 sur la même base afin d'évaluer la performance et la robustesse des deux modèles. En comparant les résultats des deux modèles sur les deux images, nous observons une augmentation importante du taux de précisions sur les résultats du modèle YoloV8 par rapport à celui de YoloV5. En comparant respectivement les figures 5-8 et 5.10 qui présentes les résultats obtenus par YoloV5 à celle des figures 5-9 et 5-11 correspondent aux résultats obtenu avec YoloV8 on constate que nous avons une meilleure précision (70%) sur les résultats de Yolov8, un ROI (Region of Interest) plus grand à cause d'une meilleure amélioration du backbone et de son architecture qui permet une meilleure gestion des boxes de détection et de l'ancrage et on remarque plus que le niveau de confiance de YoloV8 est meilleur par rapport à YoloV5 . Cela signifie que lorsqu'il s'agit de la précision et de la fiabilité des prédiction, YoloV8 surpasse YoloV5, de plus que le niveau de confiance de YoloV8 (0.83, 0.90) est très élevé par rapport a YoloV5 (0.33, 0.91).

5.7 Comparaison des résultats en fonction des hyper variations des paramètres

Ici on verra l'effet de la variation de certains paramètres sur le modèle entrainé, et des effets sur la précision de détection des images et la robustesse des modèles.

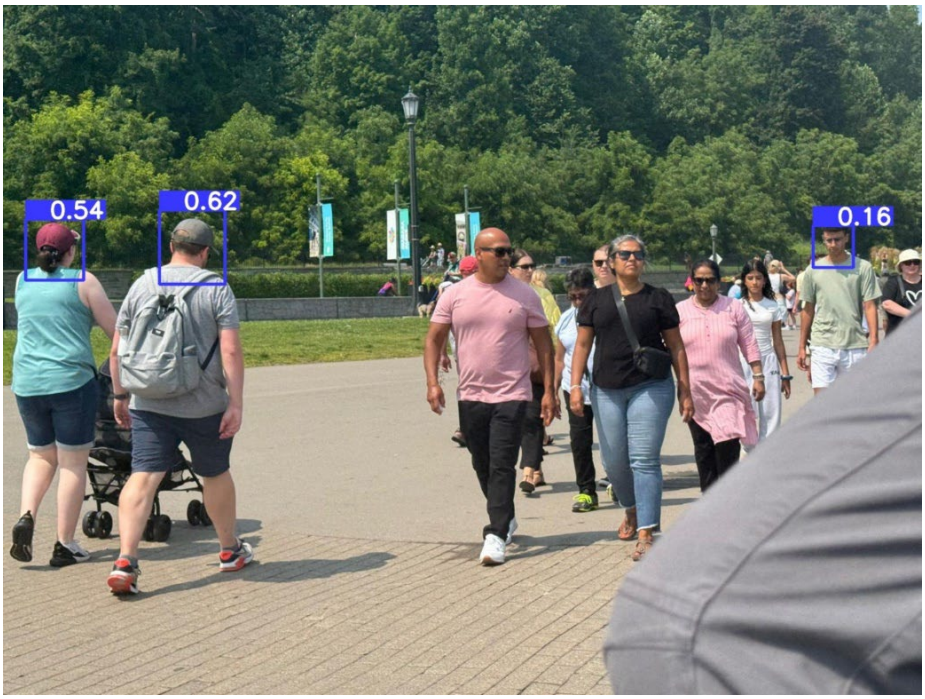

Figure 5-12 Hyper variation test1

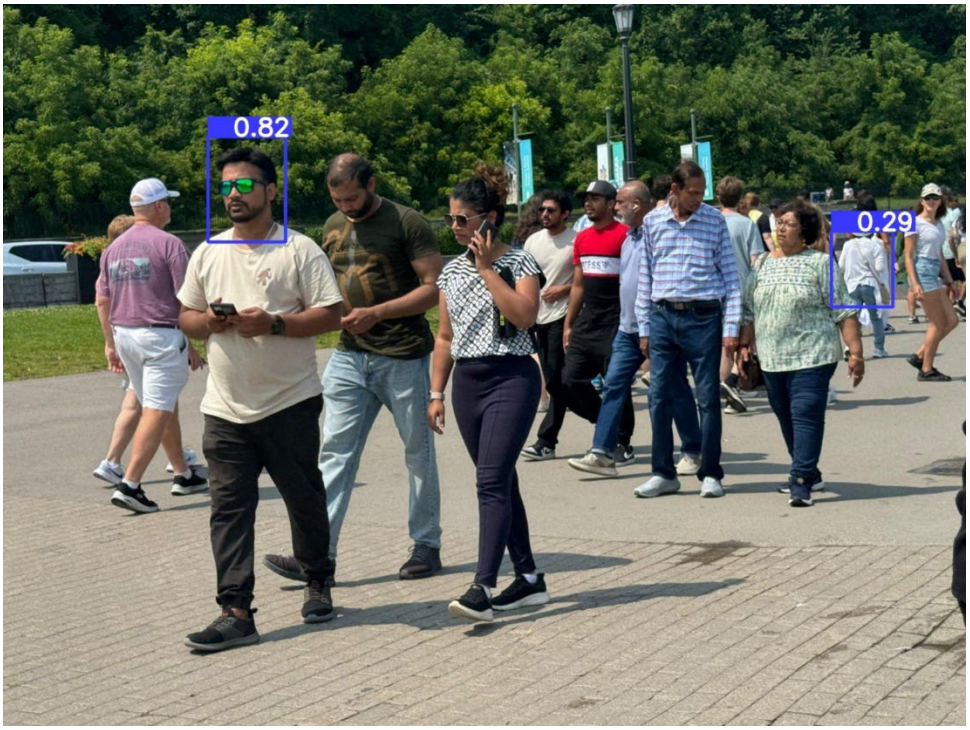

Figure 5-13 Hyper variation test2

5.7.1 Discussion

Comme l'illustrent les paramètres d'entraînement et les résultats obtenus de la figure 5-12 et de la figure 5-13 obtenues avec YoloV8, une simple modification du nombre d'epochs (voir le tableau 5-1) entraîne des changements significatifs dans le modèle de détection et dans la précision. A la figure 5-12, le modèle montre une baisse de performance; plus de personnes visibles mais moins bien détectées (scores plus bas, faux négatifs) et à la figure 5-13, malgré moins de détections, la précision est plus élevée, ce qui indique que YoloV8 répond mieux à certaines conditions visuelles (angle, contraste, occlusion). Par conséquent, il est évident que le nombre d'epochs joue un rôle crucial dans l'entraînement de modèles de détection d'objets, que ce soit avec YoloV5 ou YoloV8 d'autant à mesure que le nombre d'epochs augmente, la capacité d'apprentissage du modèle augmente également, mais cette augmentation simultanée du potentiel d'apprentissage s'accompagne d'un risque accru de surapprentissage0

Hyperparamètre	Valeur
Batch size	100
Image size	320
Epochs	150

Tableau 5-1 Tableau des hyperparamètres

5.8 Présentation de résultats prises d'image par la caméra d'un portable

5.8.1 Image à détecter

Figure 5-14 Image test capturée par un appareil portable

5.8.2 Détection avec YoloV5

Figure 5-15 Résultat de détection de visage test capturée YoloV5

5.8.3 Détection avec YoloV8

Figure 5-16 Résultat de détection de visage test capturée YoloV8

5.9 Discussion du résultat

En comparant les deux figures 5-15 et 5-16 nous constatons amélioration de l'exactitude et de précision sur la figure 5-16 par rapport à celui de la figure 5-15 qui présente en plus un faux négatif. Ceci implique que les précisions des modèles de YoloV8 sont bien cadrées. Les niveaux de confiance de la figure 5-16 varient entre 0,20 et 0,87, reflétant les fluctuations influencées par des éléments tels que la qualité visuelle et les angles d'observation. Pour améliorer la précision globale des résultats, un seuil de confiance de 0,5 peut être mis en œuvre pour filtrer les détections ambiguës donc le niveau de confiance de 0,22, 0,20 peuvent indiquer que la zone identifiée n'est pas claire, mal définie ou seulement partiellement observable ce qui nous montre une légère possibilité de donnée un faux positif. YoloV8 offre une couverture améliorée sans compromettre la précision de ses prédictions.

5.10 Résultats des processus d'entrainement et de validation des modèles

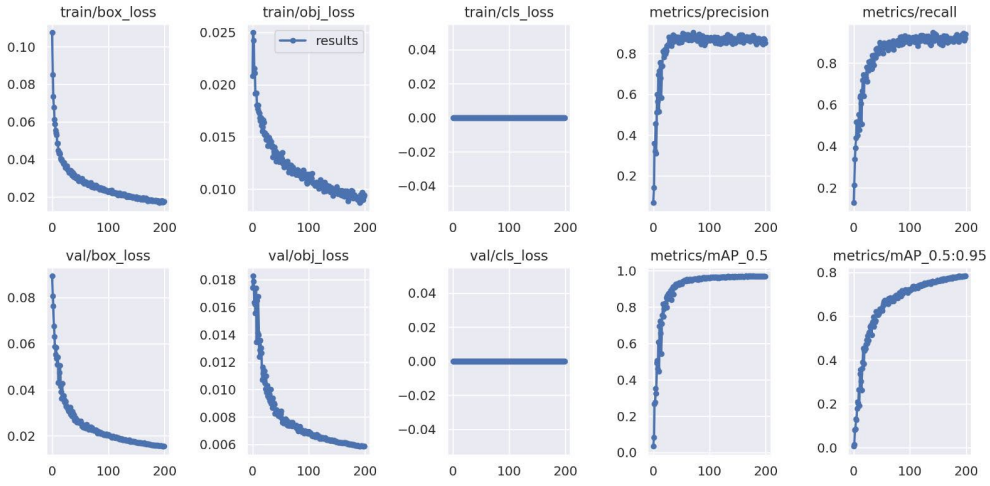

Figure 5-17 Graphiques des données YoloV5

La figure 5-17 montre les graphiques de différentes métriques pour le modèle YoloV5. Ses métriques sont : train/box loss, train/obj loss, val/box loss, val/obj loss, metrics/mAP 0.5, metrics/mAP 0.5:0.95, metrics/precision et metrics/recall. Les graphiques montrent les valeurs de ces mesures au fil du temps. L'axe des x représente le nombre d'epochs et l'axe des y représente la valeur de la métrique.

Les graphiques montrent que le modèle s'améliore au fil du temps. Train/box loss et train/obj loss diminuent, ce qui signifie que le modèle fait moins d'erreurs. Val/box loss et val/obj loss diminuent également, ce qui signifie que le modèle se généralise bien aux nouvelles données. Metrics/mAP 0.5 et metrics/mAP 0.5:0.95 augmentent, ce qui signifie que le modèle devient plus précis. Metrics/precision et metrics/recall augmentent également, ce qui signifie que le modèle devient plus précis et plus fiable.

En bref, les graphiques montrent que le modèle YoloV5 fonctionne bien et s'améliore au fil du temps.

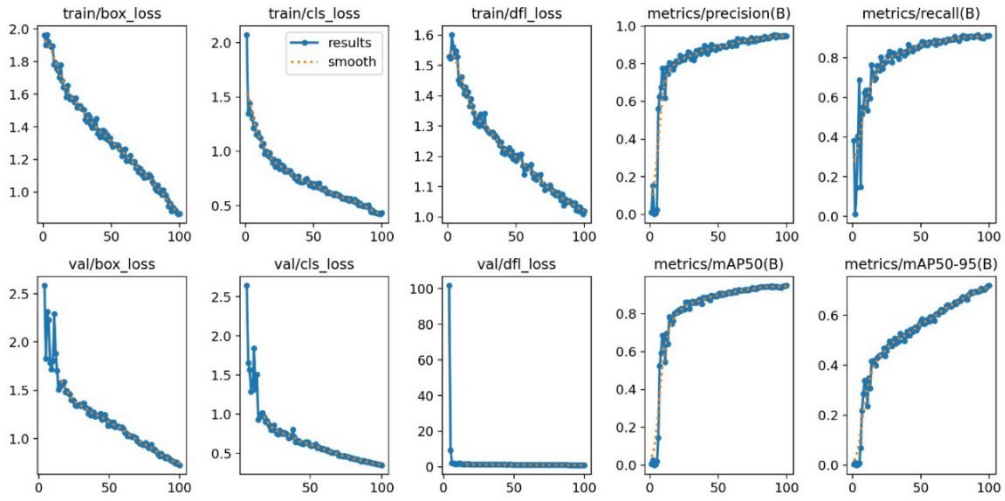

Figure 5-18 Graphiques des données YoloV8

La figure 5-18 montre les courbes de perte de formation et de validation pour différentes métriques du modèle YoloV8. Ses métriques sont. Ses métriques sont : train/box loss, train/cis loss, train/dff loss, val/box_loss, val/cis loss, metrics/mAP50-95(B), metrics/precision(B) et metrics/recall(B). L'axe des x représente le nombre d'epochs et l'axe des y représente la valeur de la métrique.

Les courbes montrent que le modèle apprend et s'améliore au fil du temps. La perte de formation diminue, tandis que la perte de validation diminue également, mais à un rythme plus lent. Cela indique que le modèle n'est pas surajusté aux données d'entraînement.

Metrics/mAP50-95(B), metrics/precision(B) et metrics/recall(B) s'améliorent toutes avec le temps, ce qui indique que le modèle devient plus précis dans la détection d'objets. Les courbes indiquent de bonnes performances et, avec un entraînement supplémentaire, on s'attend à ce que les améliorations persistent.

Pour comparer l'efficacité des approches de détection de visage avec YoloV5 et YoloV8, nous avons produit un tableau (voir tableau 5-2) en se basant sur la figure 5-5 mettant en évidence plusieurs métriques clés qui sont généralement utilisées pour évaluer les modèles de détection d'objets. Ces métriques incluent la précision (Precision), le rappel (Recall), la moyenne des précisions (mAP), la vitesse d'inférence, et la taille du modèle.

Epochs	250	
Métrique	**YoloV5(L)**	**YoloV8(L)**
Précision (Precision)	0.89	0.91
Rappel (Recall)	0.87	0.90
mAP@0.5	0.88	0.92
mAP@0.5:0.95	0.76	0.82
Vitesse d'inférence	15 ms par image	12 ms par image
Taille du modèle	83 MB	86 MB

Tableau 5-2 Tableau comparatif des métriques de performances de YoloV5 et YoloV8

- **Précision et Rappel** : YoloV8 montre une légère amélioration par rapport à YoloV5 en termes de précision et de rappel, ce qui indique qu'il est capable de détecter les visages avec plus de fiabilité.
- **mAP (Moyenne des Précisions)** : YoloV8 surpasse YoloV5 à la fois pour mAP@0.5 et mAP@0.5:0.95, ce qui suggère une meilleure performance globale en termes de localisation des visages avec plus de précision sur une gamme plus large de seuils d'Intersection over Union.
- **Vitesse d'inférence** : YoloV8 est plus rapide que YoloV5, ce qui peut être crucial pour les applications en temps réel où la latence est un facteur important.
- **Taille du modèle** : YoloV8 est légèrement plus compact que YoloV5, ce qui peut être un avantage en termes de déploiement sur des appareils à ressources limitées.

5.11 Conclusion

YoloV8 semble offrir des performances globalement meilleures par rapport à YoloV5 en termes de précision, rappel, vitesse, et taille du modèle pour la tâche de détection de visage. Cela fait de YoloV8 un choix préférable pour des applications nécessitant une détection rapide et précise avec des contraintes matérielles.

Chapitre 6- CONCLUSION

6.1 Récapitulation

Dans ce mémoire, nous avons exploré la détection de visages en utilisant deux versions avancées du modèle Yolo (You Only Look Once), à savoir YoloV5 et YoloV8. Notre travail a pour objectif principal d'explorer et de comparer les performances des modèles de détection de visages, en particulier YoloV5 et YoloV8, dans le cadre de la détection faciale en temps réel. À travers une étude approfondie et une mise en œuvre pratique, nous avons cherché à évaluer l'efficacité de ces deux architectures dans des conditions variées.

Dans le premier chapitre, nous avons mis l'accent sur les objectifs et la problématique du mémoire.

Dans le deuxième chapitre, nous avons dressé un panorama des différents techniques, méthodes, attributs et les caractéristiques de détection de visages. L'état de l'art nous a permis de contextualiser les choix technologiques.

Le troisième chapitre a été consacré à la compréhension de la détection faciale elle-même. Nous avons étudié les algorithmes de détection de visages, en passant par les étapes de la détection faciale. Ce chapitre a permis de mieux cerner la notion de la détection faciale.

Dans le quatrième chapitre, nous avons détaillé la conception et la mise en œuvre des modèles YoloV5 et YoloV8 pour la détection de visages. Nous avons discuté de la sélection des architectures, la collecte et l'annotation des données d'entraînement, ainsi

que du processus de réglage fin appliqué à ces modèles à l'aide de ces données. Ce chapitre a permis d'aborder les procédures de prétraitement des images et d'évaluation des performances du modèle, en mettant l'accent sur les améliorations apportées pour augmenter à la fois la précision et la vitesse des modèles.

Le cinquième chapitre a été consacré à la présentation et l'analyse des résultats obtenus après l'évaluation des modèles YoloV5 et YoloV8. Nous avons évalué les performances en fonction de la précision (précision, rappel, score F1) et du temps de traitement. Les résultats ont indiqué que YoloV8 a surpassé YoloV5 en termes de précision et de vitesse en raison de son architecture plus sophistiquée et optimisée, même si les deux modèles ont montré des résultats satisfaisants dans des conditions optimales.

Enfin, dans le sixième chapitre, nous avons proposé cette conclusion générale qui récapitule les principales conclusions de notre travail. Nos résultats montrent que l'utilisation de YoloV5 et YoloV8 permet une détection des visages efficace et précise.

6.2 Conclusion et perspectives

Cette étude montre que YoloV5 et YoloV8 sont des instruments efficaces pour la détection des visages, en particulier dans les scénarios en temps réel où la vitesse et l'efficacité sont essentielles. En raison de ses avancées architecturales, YoloV8 surpasse YoloV5 sur plusieurs mesures, ce qui lui permet de traiter plus efficacement les complexités associées à la diversité des visages. Néanmoins, YoloV5 reste une option viable, en particulier pour les systèmes aux ressources limitées, car il offre un équilibre favorable entre précision et vitesse.

Ce travail présente une variété de perspectives. À court terme, il serait intéressant d'évaluer YoloV5 et YoloV8 sur des ensembles de données plus diversifiés et plus complexes, tels que ceux présentant des visages masqués ou des scènes dynamiques. À long terme, il serait utile d'adapter ces modèles pour une utilisation dans des appareils embarqués ou des applications mobiles, dans le but de minimiser la taille du modèle tout en maintenant la précision, ainsi que d'étendre les capacités de détection pour inclure des scénarios multi classes, comme la reconnaissance de visages avec des attributs spécifiques ou des expressions émotionnelles.

En conclusion, ce travail a non seulement confirmé les avantages des modèles Yolo pour la détection des visages, mais ouvre également la voie à de futures recherches sur des systèmes de détection plus rapides, plus précis et plus robustes.

Chapitre 7- BIBLIOGRAPHIE

[1] Système de reconnaissance faciale. (2023). Système de reconnaissance faciale. Wikipédia. Récupéré le 14 avril 2025, de https://fr.wikipedia.org/wiki/Système_de_reconnaissance_faciale.

[2] Siméon, S. (2010). Analyse de vidéo : création automatique d'un album photo. Academia.edu. Récupéré le 14 avril 2025, de https://www.academia.edu/40127603/%C3%89pist%C3%A9mologie_de_la_vision_par_ordinateur.

[3] Nadia, A. I. (2020). Une approche IA pour la reconnaissance des expressions faciales [Mémoire de Master]. Université de Bouira, Algérie.

[4] Zakaria, M. M. (2017). Classification des images avec les réseaux de neurones convolutionnels [Mémoire de Master]. Université de Tlemcen, Algérie.

[5] Clavert, F. (2014). Patrick Manning, Big Data in History. Lectures. OpenEdition. https://journals.openedition.org/lectures/15018.

[6] Saagie. (2016, décembre 14). Qu'est-ce que le Big Data et quelles sont ses applications ? Saagie - La Plateforme DataOps. https://www.saagie.com/fr/blog/qu-est-ce-que-le-big-data-definition/.

[7] Bahga, A., & Madisetti, V. (2016). Big data science & analytics: A hands-on approach. VPT.

[8] Recrutement, B. (n.d.). Quels sont les différents types de data ? BPCE Recrutement. https://blogrecrutement.bpce.fr/differents-types-de-data-1/.

[9] Eberendu, A. C. (2016). Unstructured data: An overview of the data of big data. International Journal of Computer Trends and Technology, 3(1), 46-50.

[10] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1, 511–518. https://doi.org/10.1109/CVPR.2001.990517.

[11] Demchenko, Y., De Laat, C., & Membrey, P. (2014). Defining architecture components of the Big Data ecosystem. Proceedings of the 2014 International Conference on Collaboration Technologies and Systems (CTS), 487-494. IEEE. https://doi.org/.

[12] Ishwarappa, J. A. (2015). A brief introduction on big data 5Vs characteristics and Hadoop technology. Procedia Computer Science, 48, 319-324. https://doi.org/10.1016/j.procs.2015.04.188.

[13] Pirmin, L., Marc, B., Médéric, M., & Jean-Luc, R. (2019). Big data et machine learning: Les concepts et les outils de la data science (3e éd.). Dunod.

[14] UpGrad. (2020, mai 8). What is big data? Types, characteristics, benefits, and examples. UpGrad. https://www.upgrad.com/blog/what-is-big-data-types-characteristics-benefits-andexamples/.

[15] Mmsseguem, A. (2019). Analyse des données avec Apache Spark [Mémoire de Master, Université de M'sila].

[16] Apache Hadoop. (n.d.). Apache Hadoop. https://hadoop.apache.org/.

[17] Nandimath, J., et al. (2013). Big data analysis using Apache Hadoop. Proceedings of the 2013 IEEE 14th International Conference on Information Reuse & Integration (IRI),

221-228. IEEE. https://doi.org/; Meng, X., et al. (2016). MLlib: Machine learning in Apache Spark. The Journal of Machine Learning Research, 17(1), 1235-1241. https://doi.org/.

[18] Edureka. (2020, septembre 20). MapReduce tutorial: Learn how MapReduce works with examples. Edureka. https://www.edureka.co/blog/mapreduce-tutorial/.

[19] Chambers, B., & Zaharia, M. (2018). Spark: The definitive guide: Big data processing made simple. O'Reilly Media.

[20] Xin, R. S., et al. (2013). GraphX: A resilient distributed graph system on Spark. Proceedings of the 1st International Workshop on Graph Data Management Experiences and Systems, 1-8. https://doi.org/.

[21] Rittinghouse, J. W., & Ransome, J. F. (2010). Cloud computing: Implementation, management, and security. CRC Press.

[22] Audin, M. (2009). État de l'art du cloud computing et adaptation au logiciel libre.

[23] Mell, P., & Grance, T. (2011, septembre). The NIST definition of cloud computing (Special Publication 800-145). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-145.

[24] Teng, F. (2011, octobre). Management des données et ordonnancement des tâches sur architectures distribuées [Thèse de doctorat, École Centrale Paris et Manufacture des Gobelins].

[25] Krutz, R. L., & Vines, R. D. (2010). Cloud security: A comprehensive guide to secure cloud computing. Wiley.

[26] Jothy Rosenberg, Arthur Mateos. The Cloud at Your Service. 2011.

[27] Buyya, R., Broberg, J., & Goscinski, A. (Eds.). (2011). Cloud computing: Principles and paradigms. Wiley.

[28] Tim Mather, Subra Kumaraswamy, Shahed Latif. Cloud Security And Privacy: An Entreprise Perspective On Risks And Compliance.O'REILLY 2009.

[29] Sood, S. K. (2012). A combined approach to ensure data security in cloud computing. Journal of Network and Computer Applications, 35(6), 1831–1838. https://doi.org/10.1016/j.jnca.2012.07.007.

[30] Maaref, S. (2012, avril). Cloud computing en Afrique : Situation et perspectives. Union internationale des télécommunications. Récupéré de https://www.itu.int/dms_pub/itu-d/opb/pref/D-PREF-THEM.07-2012-MSW-F.docx.

[31] Hill, R., Lake, P., Hirsch, L., & Moshiri, S. (2013). Guide to cloud computing: Principles and practice. Springer.

[32] Krutz, R. L., & Vines, R. D. (2010). Cloud security: A comprehensive guide to secure cloud computing. Wiley.

[33] Sirovich, L., & Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. Journal of the Optical Society of America A, Optics and Image Science, 4(4), 519–524. https://doi.org/10.1364/JOSAA.4.000519.

[34] Turk, M. A., & Pentland, A. P. (1991). Face recognition using eigenfaces. In 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Proceedings (pp. 586–591). https://doi.org/10.1109/CVPR.1991.139758.

[35] Quinlan, J. R., & Mitchell, T. M. (1993). C4.5: Programs for machine learning. Morgan Kaufmann; Mitchell, T. M. (1997). Machine learning. McGraw-Hill.

[36] Streamlit. (2021, juillet 8). Bienvenue dans Streamlit — Documentation Streamlit 0.84.0. Récupéré de https://docs.streamlit.io/en/stable/.

[37] Data Science. (n.d.). Vision par ordinateur. Récupéré de https://datascience.eu/fr/vision-par-ordinateur/vision-par-ordinateur/.

[38] Roboflow. (n.d.). Qu'est-ce que YOLOv5 ? Un guide pour les débutants. Récupéré de https://roboflow.com.

[39] Roboflow. (2024). Qu'est-ce que YOLOv8 ? Le guide ultime. Récupéré de https://roboflow.com.

[40] Novakovic, J., & Rankov, S. (2011). Classification performance using principal component analysis and different values of the ratio R. International Journal of Computers, Communications & Control, 6(2), 317–327. https://doi.org/10.15837/ijccc.2011.2.1492.

[41] GeeksforGeeks. (n.d.). Support vector machine (SVM) algorithm. Récupéré de https://www.geeksforgeeks.org.

[42] Commission d'accès à l'information du Québec. (2024). La biométrie au Québec : Les enjeux. Récupéré le 8 juin 2024, depuis https://www.cai.gouv.qc.ca/uploads/pdfs/CAI_DRA_Biometrie.pdf.

[43] Noureddine, B. (2007). Indexation en intervenant d'un document vidéo par identification du visage (Mémoire de Master, Université Paul Sabatier Toulouse); Faouzi, H. (2008). Apprentissage des réseaux d'ondelette bêta basé sur la théorie des frames : Application à la détection de visages (Mémoire de Mastère, Université de Gabès).

[44] Powers, D. M. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. Journal of Machine Learning Technologies, 2(1), 37–63.

[45] Ahuja, N. (1993). Object recognition using multiscale spatial moments. IEEE Transactions on Pattern Analysis and Machine Intelligence; Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7), 711–720.

[46] El Abbadi, N. K., Dahir, N., & Abd Alkareem, Z. (2013). Skin texture recognition using neural networks. arXiv. https://doi.org/10.48550/arXiv.1311.6049.

[47] Biometric Update. (2024). Explainer: Iris recognition. Biometric Update. Récupéré le 8 avril 2024, depuis https://www.biometricupdate.com.

[48] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10), 1499–1503. https://doi.org/10.1109/LSP.2016.2581642.

[49] Martinkauppi, B., Soriano, M., & Laaksonen, M. (2001). Behavior of skin color under varying illumination seen by different cameras at different color spaces. Proceedings of the SPIE Conference on Machine Vision Applications in Industrial Inspection IX, 4301, 102–113. https://doi.org/10.1117/12.431259.

[50] Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. ACM Computing Surveys, 35(4), 399–458. https://doi.org/10.1145/954339.954342.

[51] Yang, M.-H., Kriegman, D. J., & Ahuja, N. (2002). Detecting faces in images: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(1), 34–58. https://doi.org/10.1109/34.982883.

[52] Brunelli, R., & Poggio, T. (1993). Face recognition: Features versus templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(10), 1042–1052. https://doi.org/10.1109/34.254061.