

UNIVERSITÉ NATIONALE DU VIETNAM À HANOÏ



INSTITUT FRANCOPHONE INTERNATIONALE



OPTION : Master en Systèmes Intelligents et Multimédia
Promotion : 26

Rapport du Travail de Vision par Ordinateur

SIM.P26,
Prof. Nguyễn Thị Oanh

Rédigé par:

MUNDUKU MUNDUKU DEO

Introduction

Ce rapport évalue la méthode de Viola & Jones pour la détection de visages, implémentée dans OpenCV à l'aide de classificateurs en cascade pré-entraînés. Nous avons exploré l'influence de divers paramètres de détection, en particulier la taille de la fenêtre et le ratio d'overlap entre deux fenêtres consécutives, sur la performance du modèle de classification des visages.

Question I : Détection de Visages avec la Méthode de Viola & Jones

Méthodologie

Pour ces tests, nous avons utilisé le modèle pré-entraîné haarcascade_frontalface_default.xml, fourni par OpenCV, spécialement conçu pour détecter les visages frontaux. Les tests ont été réalisés en ajustant les paramètres minSize et minNeighbors afin d'évaluer leur impact sur la précision et la fiabilité de la détection dans différents scénarios. Dans le but d'observer attentivement les résultats des variations de différents paramètres dans divers scénarios et de déterminer s'il existe des explications justifiant les comportements de détection observés, nous avons choisi trois scénarios distincts.

A. Scénario avec un seul visage dans l'image

Pour obtenir une détection précise et fiable d'un seul visage dans une image, nous avons progressivement ajusté le paramètre MinNeighbors ainsi que la taille de la fenêtre, évaluant leur impact sur la qualité de la détection. Ce paramètre spécifie le nombre minimum de rectangles voisins qu'un candidat visage doit avoir pour être validé comme une détection. En augmentant le MinNeighbors, nous réduisons les faux positifs en exigeant une confirmation accrue par d'autres rectangles détectés autour de la même région avant de considérer une détection comme valide. En parallèle, nous avons évalué l'influence de la taille de la fenêtre sur la détection, en explorant différentes échelles de détection. Ce processus nous a permis de comprendre comment le modèle se comporte dans des situations où il n'y a qu'un seul visage dans l'image, et si les variations de paramètres ont un impact significatif sur la performance de détection.

A.1 Résultats Observés

Pour mener à bien notre premier scénario avec un seul visage, nous avons fixé la taille de la fenêtre et le paramètre MinNeighbors avec plusieurs variations afin de comprendre les différentes influences. Nous avons testé trois tailles de fenêtre différentes : (30, 30), (25, 25) et (40, 40), tandis que le paramètre MinNeighbors a été augmenté de 1 à 7.

Taille de la fenêtre: (30, 30) pour les deux figures (1 et 2) , MinNeighbors: 2 (pour la Figure 1) et (3 pour la Figure 2)



Fig.1: 1 seul Visage Détecté avec MinNeighbors =2

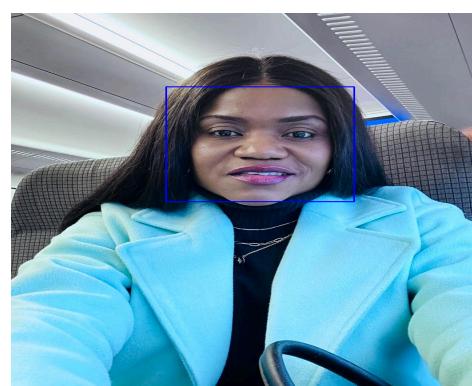


Fig.2: 1 seul Visage Détecté avec MinNeighbors =3

Taille de la fenêtre: (40, 40) pour les deux figures (3 et 4) , MinNeighbors: 4 (pour la Figure 3) et 5 (pour la Figure 4)

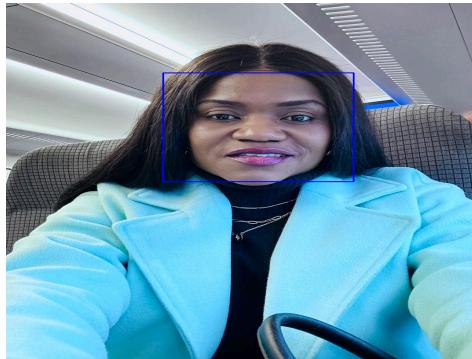


Fig.3: 1 seul Visage DéTECTé avec MinNeighbors =4

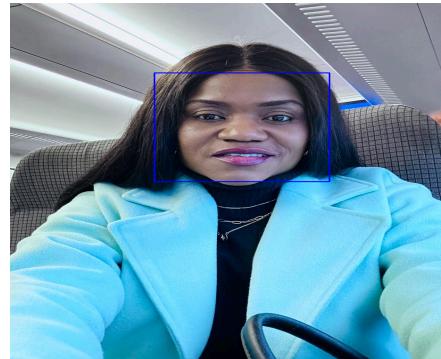


Fig.4: 1 seul Visage DéTECTé avec MinNeighbors =5

A.2 Mauvais résultat

Taille de la fenêtre: (30, 30) pour la Figure 5 et (40,40) pour la Figure 6, MinNeighbors: 1 pour les deux Figures

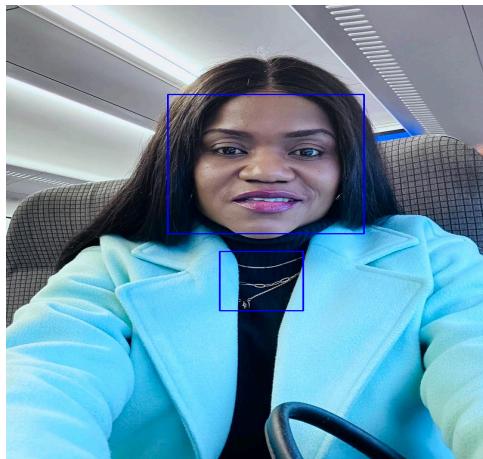


Fig.5: 1 mauvais résultat avec MinNeighbors =1

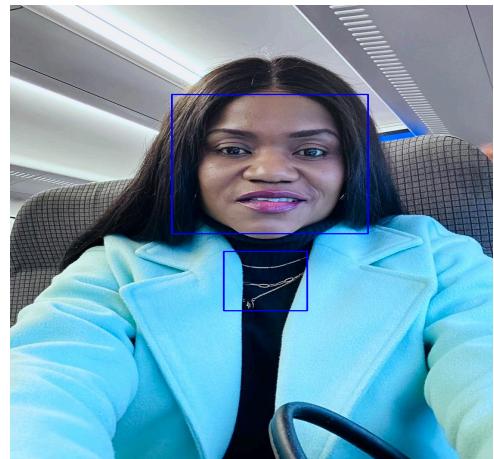


Fig.6: 1 mauvais résultat , avec MinNeighbors =1

B. Scénario avec un plusieurs visages dans l'image

Ce processus d'expérimentation nous a permis de comprendre comment le modèle se comporte lorsqu'il y a plusieurs visages dans une image. Nous avons analysé si les variations de paramètres ont un impact significatif sur la capacité du modèle à détecter avec précision chaque visage, ainsi que sur le nombre de faux positifs générés.

B.1. Résultats Observés

Pour mener à bien notre deuxième scénario avec plusieurs visages , nous avons fixé la taille de la fenêtre et le paramètre MinNeighbors avec plusieurs variations afin de comprendre les différentes influences. Nous avons testé trois tailles de fenêtre différentes : (30, 30), (25, 25) et (40, 40) , tandis que le paramètre MinNeighbors a été augmenté de 1 à 7.

Taille de la fenêtre: (30, 30) pour la figure 7 et (25,25) pour la Figure 8 ,
MinNeighbors: 5 (pour la Figure 7) ; et 6 (pour la Figure 8)



Fig.7: Détection à plusieurs visages avec MinNeighbors =5

Taille de la fenêtre: (40,40) pour les deux Figures (9 et 10) , MinNeighbors: 5 (pour la Figure 9) ; et 6 (pour la Figure 10)



Fig.8: Détection à plusieurs visages avec MinNeighbors =6



Fig.9: Détection à plusieurs visages avec MinNeighbors =5



Fig.10: Détection à plusieurs visages avec MinNeighbors =6

B.2 Mauvais résultat

Taille de la fenêtre: (50, 50) pour la Figure 11 et (25,25) pour la Figure 12; MinNeighbors: 1 (pour la Figure 11) et 2 (pour la Figure 12).



Fig.11: mauvais résultat avec MinNeighbors =1

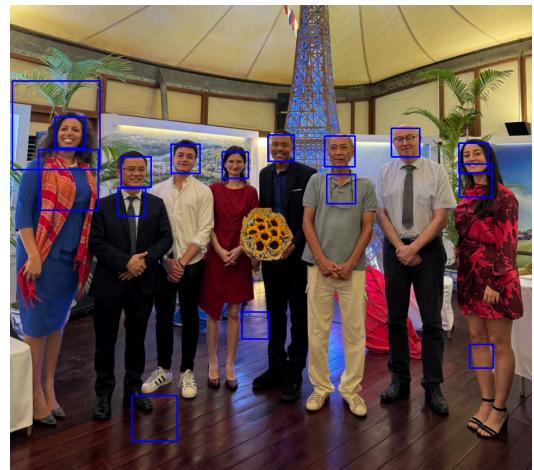


Fig.12: mauvais résultat avec MinNeighbors =1

OBSERVATION POUR LA PREMIÈRE QUESTION

Dans cette première question, nous avons exploré la détection de visages en utilisant la méthode de Viola & Jones dans deux scénarios distincts : un scénario avec un seul visage dans l'image et un autre avec plusieurs visages.

Dans le premier scénario, nous avons constaté que l'ajustement des paramètres MinNeighbors et la taille de la fenêtre a un impact significatif sur la performance de la détection. En augmentant le MinNeighbors, nous avons réussi à réduire les faux positifs, améliorant ainsi la précision de la détection. De même, en ajustant la taille de la fenêtre, nous avons pu évaluer différentes échelles de détection, ce qui a permis de mieux comprendre le comportement du modèle dans des situations où il n'y a qu'un seul visage dans l'image.

Dans le second scénario, où nous avons analysé des images contenant plusieurs visages, nous avons également observé une corrélation entre les paramètres MinNeighbors et la qualité de la détection. De plus, l'exploration de la taille de la fenêtre a mis en évidence son influence sur la capacité du modèle à détecter précisément chaque visage dans une image avec plusieurs sujets.

En se basant sur ces deux scénarios ,nous pouvons affirmer que Les variations de ces paramètres ont un impact significatif sur la performance du modèle même si il a été entraîné comme dans notre cas , ce qui souligne la nécessité d'une exploration minutieuse pour obtenir des résultats précis et fiables, que ce soit dans des scénarios avec un seul visage ou avec plusieurs visages dans l'image.

2.Reconnaissance d'objet à l'aide du modèle de classification et de la détection par sliding window

Dans cette section, nous abordons deux étapes essentielles de notre projet : l'apprentissage et la détection. Nous commencerons par la construction d'un modèle de(classification) reconnaissance d'objet via la classification, puis nous explorerons la détection de l'objet à l'aide de la technique de sliding window.

2.1. Construction du Modèle de Classification

La première étape de notre processus consiste à développer un modèle de classification robuste. Pour ce faire, nous avons utilisé un ensemble de données étiqueté de **MSCOCO** (<https://cocodataset.org/#home>) pour faire **le fine tuning de notre modèle**.le Tableau 1 ci-dessous montre la répartition de données pour l'entraînement (Fine Tuning)

Tableau 1.répartition des données

répartition de données pour l'entraînement	quantité
Training data	1200 images de voitures (70%)
Test data	180 (15%)
Validation data	180(15%)

L'objectif était d'atteindre une précision élevée lors de la classification et de permettre le modèle préentraîner de s'habituer avec notre jeu de données , ce qui est crucial pour la réussite des étapes suivantes de détection. Les résultats obtenus, illustrés par les Figures 1 et 2 ci-dessous (graphiques de perte et de précision) , montrent une amélioration significative de la précision du modèle au fil des époques lors de l'entraînement .

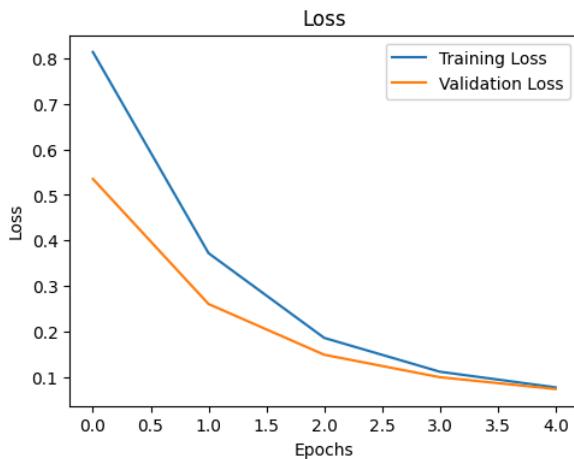


Fig.1: courbe de loss du Fine tuning

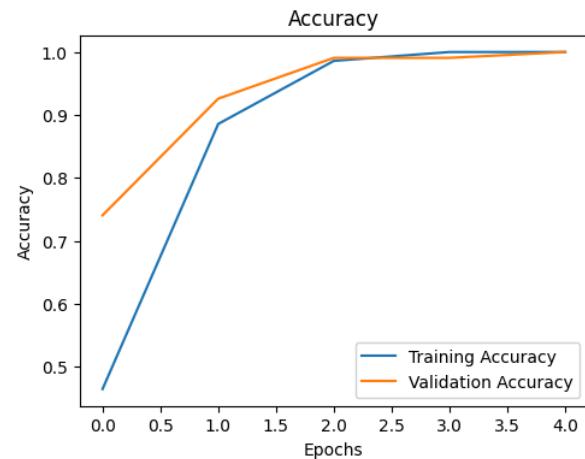


Fig.2: courbe de l'accuracy du Fine Tuning

Cette progression indique que notre modèle apprend efficacement à partir des données d'entraînement parce qu'il bénéficie du poids existant du modèle pré-entraîné. La perte d'entraînement diminue rapidement, tandis que la précision d'entraînement augmente, atteignant presque 98% à la dernière époque. Cependant, une légère divergence entre les courbes de validation et d'entraînement suggère le début d'une suradaptation, que nous prévoyons de contrôler à l'aide de techniques telles que l'arrêt précoce et la régularisation.

Tableau 2. Résultat du modèle après le Fine tuning

<u>Accuracy=0.98</u>	Précision=0.94	Recall=0.93	F1 Score=0.935
----------------------	----------------	-------------	----------------

2.2 Résultat visuelle de la classification

cette partie illustrer par la Figure 3 et 4 montre le résultat visuelle de la classification avant de faire la détection Nous utilisons une image pour illustrer tout le cas cad de la classification à la détection complète



Figure 3 : résultat de la classification



Figure 4 : résultat (correcte car la moto ne pas classifier de la classification de notre modèle)

2.2 présentation de la technique de sliding windows pour notre cas

Dans cette section, nous aborderons la détection d'objets à l'aide de la technique de sliding windows. Cette approche consiste à diviser l'image en fenêtres de taille fixe et à les parcourir de manière itérative pour détecter les objets présents; comme illustré dans notre Figure 4 qui montre le déplacement de la fenêtre glissante avant la détection .

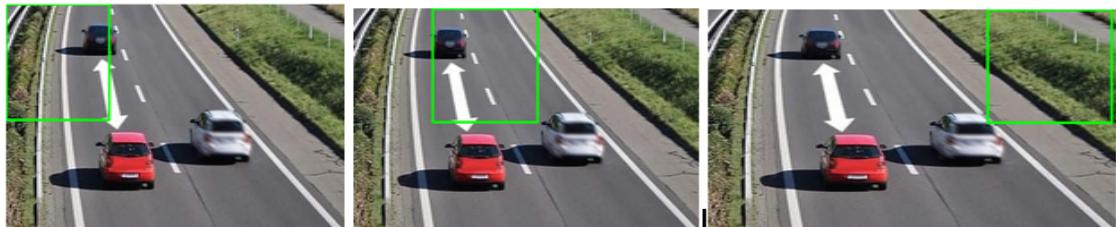


Figure 4 : illustration de déplacement de sliding windows dans notre champs d'application

2.2 Détection d'objet avec la technique de sliding windows

Dans cette section, nous aborderons la détection Finale à l'aide de la technique de sliding windows et la classification se fais a l'aide de notre modèle .

RESULTAT Final

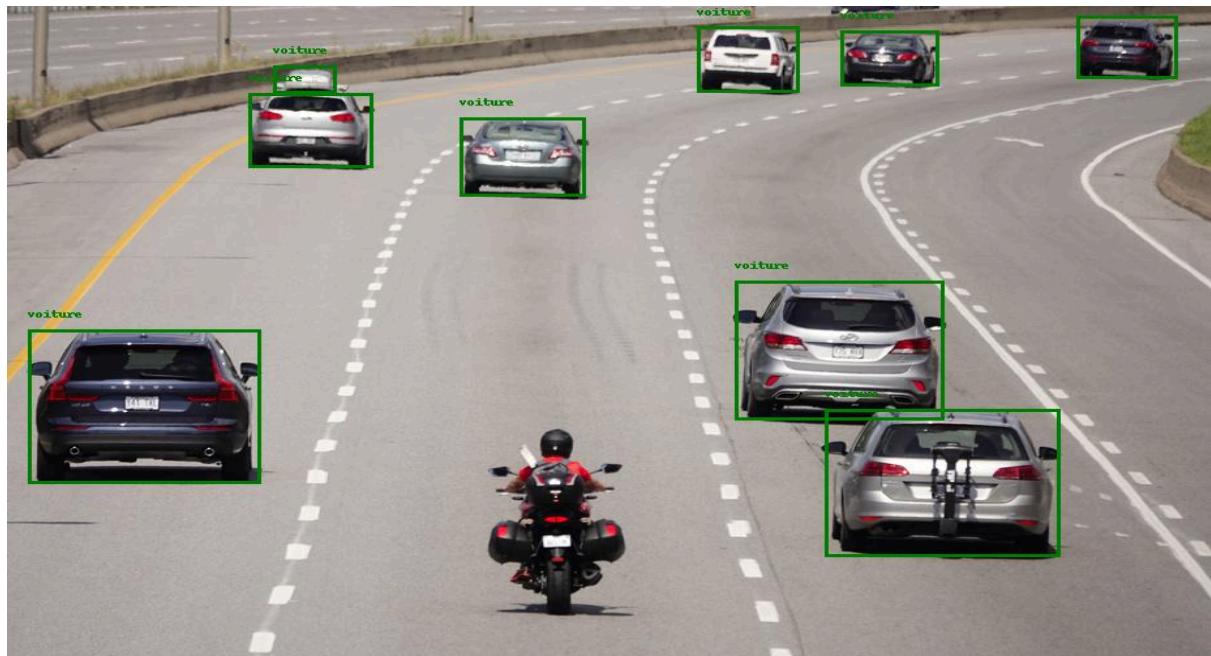


Figure 5 : résultat final de la détection de voiture à l'aide de la fusion de notre modèle (Fine tuning et la technique de sliding

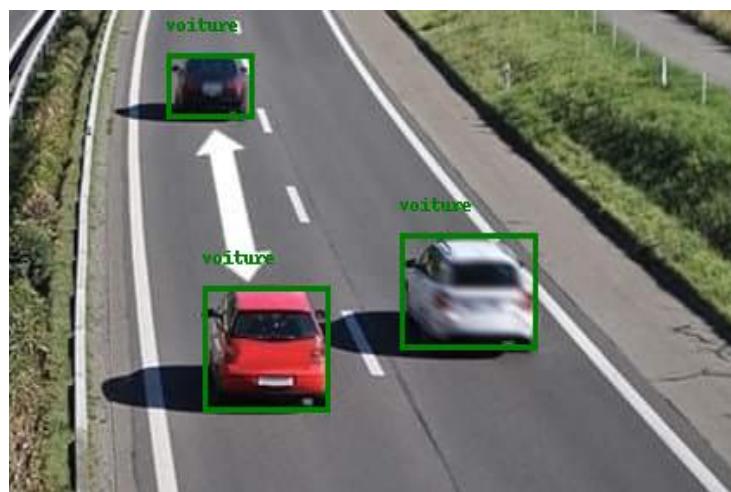


Figure 6 : resultat final de la detection de voiture a l'aide de la fusion de notre modele (Fine tuning et la technique de slinding windows)

