UNIVERSITÉ NATIONALE DU VIETNAM À HANOÏ
INSTITUT FRANCOPHONE INTERNATIONALE

OPTION : SYSTÈMES INTELLIGENTS ET MULTIMÉDIA (SIM)

PROMOTION : 25 ANNÉE ACADÉMIQUE 2022-2023

MODULE : APPRENTISSAGE AUTOMATIQUE

FACE MASK DETECTION

SOLUTION PROPOSÉE

TANKEU NGUEKEU BAUDELAIRE ISMAËL

NITCHEU MONKAM JUNIOR

IBANZI KILONGO CHRISTIAN

ENCADRÉS PAR :

VU VIET-VU, Lecturer-Researcher in Computer Sciences

Table des matières

Table des figures

1 Introduction

Le monde a souvent été bouleversé par de nombreux phénomènes naturel causant la mort à de nombreuses personnes. Parmi ces phénomènes, on peut citer les épidémies, les catastrophes naturelles, les guerres, etc. Une épidémie désigne l'augmentation rapide d'une maladie en un lieu donné sur un moment donné.
L'épidémie s'applique initialement aux maladies touchant les humains ; si la maladie s'étend rapidement à une part importante de la planète, on parle alors de pandémie. La maladie à Coronavirus en est un exemple et on estime le nombre de mort causé par cette maladie à $6,41 milion$ de morts à ce jour.
L'une des causes particulières de ce type de maladie, c'est qu'elle est très contagieuse d'ou un effet de contamination exponentiel pour un taux de décès souvent très élevé. De nombreuses mesures ont été prises afin de contrecarré cette maladie à covid19 comme les couvre-feux, le port du masque, le respect des mesures barrières, etc. Ces différentes mesures ont été utilisées pour limiter la propagation de la maladie.

2 Problématique et objectifs

Ne pouvant pas laissés les gens indéfiniment chez eux, il était question de pouvoir leur laisser continuer à continuer leur occupation quotidienne tout en évitant au maximum les risques de contamination. Une des mesures va alors attirer notre attention, celle du **port du masque**. Ainsi, le port du masque permet d'éviter de :

— contracter la COVID-19 ;

— transmettre la COVID-19 à d'autres personnes ;

Ces avantages de port de masque permettent ainsi de devoir s'assurer que les gens dans les milieux respectent cette mesure. Pour cela, il faudrait traquer les usagers véreux qui essaient de contourner cette règle dans les milieux publics. A partir des caméras existant dans ces milieux, on pourraient visualiser les personnes qui portent un masque ou qui portent à moitié ou qui ne portent même pas. Nous devons ainsi :

— détecter un le visage de chaque personne se trouvant dans un milieu public ;

— cadrer la partie se trouvant entre son menton et la partie inférieure de son nez en passant par le derrière de ses oreilles ;

— pouvoir dire si cette personne porte un masque de moitié ou pas ou bien n'en porte pas grâce à au modèle créé ;

3 Solution

L'apprentissage automatique peut nous été pour la création d'un tel modèle dans la détection de masque chez une personne.
Trois types d'algorithme meuble l'apprentissage automatique dans sa quête de solutions à un problème tels que l'apprentissage supervisé, l'apprentissage sémi-supervisé et l'apprentissage non supervisé.
Cette qui fera l'objet de notre rétention est l'apprentissage supervisé, qui est une approche proposant de nombreux algorithme comme le **K-Means**, **DBSCAN**, **l'arbre de décision**, **Random Forest**, **machines à vecteurs de support ou Support-vector machine (SVM)**, **deep**

learning, etc. De ces multiples algorithmes, le deep learning sera l'algorithme utilisé pour créer notre modèle.
L'application que nous allons réalisé, sera à mesure de prendre en entrée la photo d'une personne et à la sortie elle affiche le résultat en disant si la personne porte le masque ou pas.

4 Architecture du système

Le **Deep Learning** est un algorithme qui est venu revolutionné le **Machine Learning** plus particulièrement dans la Vision d'ordinateur. Il existe de nombreux framework permettant de créer de nouveaux modèles en se basant sur des modèles pré-entraînés à l'instar de **Pytorch, Tensorflow, etc.** Dans notre cas, nous intéresserons au framework Tensorflow.

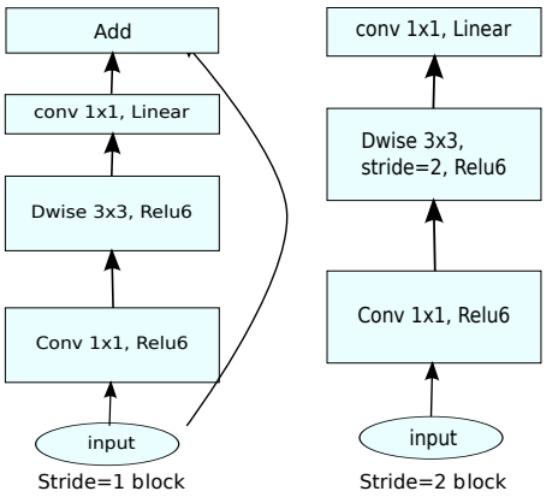
Les modèles officiels de TensorFlow sont une collection de modèles qui utilisent les API de haut niveau de TensorFlow. Ils sont destinés à être bien entretenus, testés et mis à jour avec la dernière API TensorFlow. Ils doivent également être raisonnablement optimisés pour des performances rapides tout en restant faciles à lire. Ces modèles sont utilisés comme tests de bout en bout, garantissant que les modèles s'exécutent avec une vitesse et des performances identiques ou améliorées à chaque nouvelle version de TensorFlow. Il existe ainsi de nombreux modèles dépendant de quel type d'opérations on voudrait réaliser. On peut avoir comme modèles de la vision par ordinateur selon l'approche :

— Segmentation : ResNet, Revisiting ResNets, EfficientNet et vision transformer.

— La détection et la segmentation : RetinaNet, Mask R-CNN, SpineNet et Cascade RCNN-RS and RetinaNet-RS

— La classification des vidéos : ALBERT, BERT et ELECTRA ELECTRA.

De ces nombreux modèles, nous avons utilisé MobileNetV2 qui est une architecture de réseau neuronal convolutif qui vise à obtenir de bonnes performances sur les appareils mobiles. Il est basé sur une structure résiduelle inversée où les connexions résiduelles se trouvent entre les couches d'étranglement. Son architecture est représenté de manière suivante :



(d) Mobilenet V2

FIGURE 1 – Architecture du modèle MobileNetV2

Ainsi par une telle architecture de ce modèle, nous pouvons ainsi construire l'architecture de notre système. Cette architecture se présente en deux parties qui sont : La partie entraînement et la partie détection de masque.

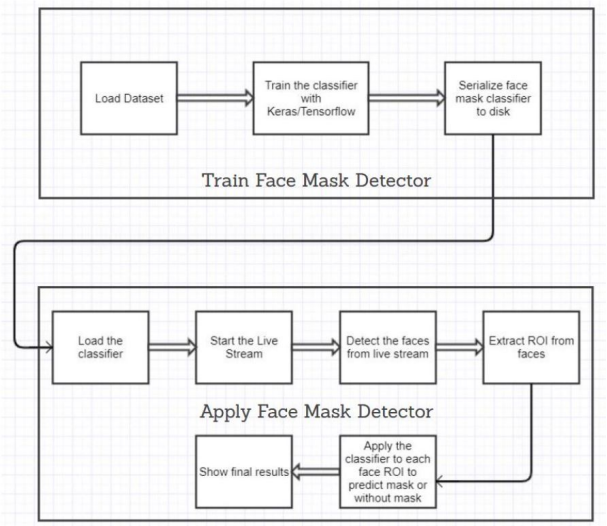


FIGURE 2 – Architecture générale du système

4.1 Les données d'apprentissages

Afin de mettre sur pieds notre système, Nous avons utilisé un jeu de données constitué de 3833 images contenant 1915 images des personnes portant un masque et 1918 images des personnes ne portant pas de masques d'où l'existence de deux classes (Personnes avec masques et personnes sans masques). Les figures ci-dessous présentent un exemplaire de de représentation de nos deux classes : Par ces deux classes d'images, il sera question pour nous

FIGURE 3 – Personne avec masque

FIGURE 4 – Personne avec masque

de créer un modèle permettant de dire si une personne possède un masque ou non. Pour y arriver, nous avons subdivisé notre dataset en deux (02) avec 20% d'images de personnes ayant un masque ou non pour le test et 80% pour l'entraînement.

4.2 L'Implementation

Notre programme a été implémenté dans l'environnement python3 avec l'environnement de Développent Intégré PyCharm.Nous avons pré-traité nos données d'entraînement afin d'obtenir un entraitement performant, nous avons aussi par la créer notre modèle et par la suite faire des test pour évaluer la qualité de notre modèle obtenu.

4.3 Le pré-traitement

Nous avons appliqué quelques techniques de pré-traitement de données afin de pouvoir convertir nos données état encore brutes en données propres susceptibles à être utilisé dans la phase d'entraînement. Le pré-traitement a consisté en deux principales étapes à savoir la labélisation et l'augmentation des données.En ce qui concerne la labélisation, nous avons identifié la classe d'appartenance de chacune des image à partir des dossiers with-mask et without-mask.Nous avons appliqué la binarisation sur ces labels permettant de les identifier par des valeurs numériques 0 et 1. En ce qui concerne l'augmentation des données d'entrainement nous avons utilisé des techniques telles que la rotation, le zoom, la translation verticale et horizontale. Nous avons entre autre redimenssionné nos images à la dimention (224,224) afin de les rendre compatible à l'architecture pré-entrainé qui constitue l'entrée de notre modèle.

4.4 L'entraînement du modèle

Notre modèle est constitué du modèle de base pré-entrainé MobileNetV2 auquel nous avons ajouté un modèle de sortir que nous avons concu. En effet, le modèle pré-entrainé MobileNetV2 quiconstitue l'entrée de notre modèle ne sera plus entrainé lors de l'entrainement. L'entrainement sera donc effectué uniquement sur le modèle que nous avons mis sur pied.En ce qui concerne les paramètres appliqués pour l'entrainement, nous avons Epochs = 20, Batch-Size = 32 et le learning-rate = 1e-4. Ceci, permet de visualiser un ensemble de résultats :

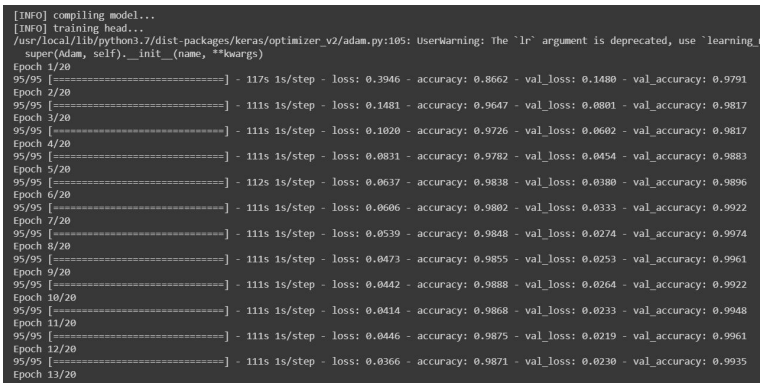


FIGURE 5 – Étapes d'entraînement du modèle en tenant compte des hyper-paramètres

5 Résultat

Après entraînement de notre modèle, nous obtenons le résultat ci-dessous afin d'évaluer la performance de notre modèle lors de l'entraînement et de la validation de celui-ci.

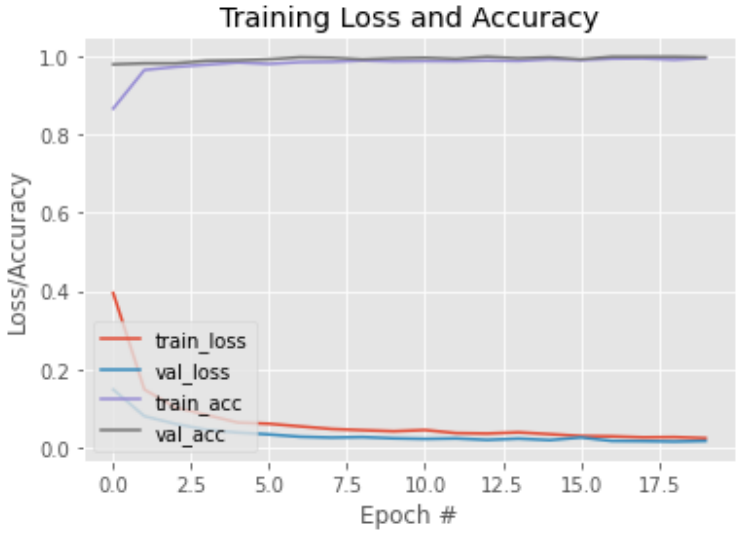


FIGURE 6 – précision et perte de l'entraînement et de la validation

Afin, d'approuver, la performance de ce modèle, il était question pour nous d'effectuer des tests afin de le valider. Ceci a été fait grâce à une prise vidéo pour confirmer le résultat obtenu lors de l'entraînement et de la validation. Nous l'avons testé avec nos propres données et nous obtenons le résultat suivant :

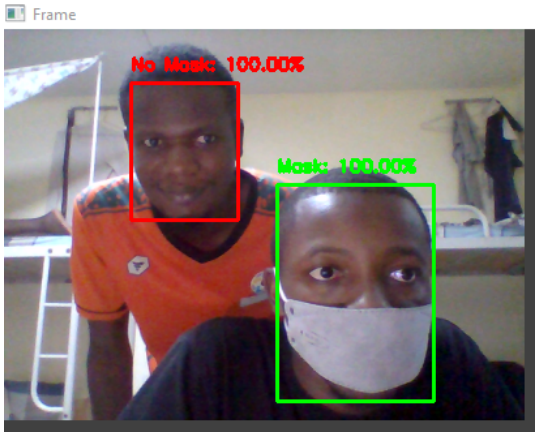


FIGURE 7 – Test de notre modèle

Par ce résultat de test, nous pouvons apprécier la qualité de notre modèle et dire qu'il est capable de détecter si une personne possède le masque ou pas. Mais, on peut aussi remarquer sans rapprochement de plus près à la caméra, il n'est pas capable de détecter les visage donc on ne peut avoir de prédiction.

6 Conclusion

Le problème traité dans ce projet a un enjeu capital car tous savons les lourdes pertes en vie humaines causées par le COVID19. Le développement de ce projet qui consiste à détecter si un visage porte un masque ou non fait appel à l'utilisation de plusieurs techniques de machine learning et de deep learning en passant par les étapes de pré-traitement, d'entraînement et d'évaluation.Nous avons particulièrement utilisé le modèle pré-entraîné MobileNetV2. Les résultats obtenus ont prouvé que notre modèle obtenu est un modèle performant.