

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

UNIVERSITE BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY



جامعة باجي مختار – عنابة

Faculté : de TECHNOLOGIE

Département : Electronique

Domaine : Sciences et Techniques

Filière : Électronique

Spécialité : Électronique des systèmes embarqués

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème :

Réalisation D'un Prototype de Détection du Port de Mask Covid Avec
Raspberry PI

Présenté par : *Saadi Abd El Kayoum*

Daoudi Kadour Raouf

Encadrant : *DAAS Sara*

M.A.B

Université de Badji Mokhtar Annaba

Jury de Soutenance :

BOUSBIA-SALAH Mounir	Prof	Université de Badji Mokhtar Annaba	Président
DAAS Sara	M.A.B	Université de Badji Mokhtar Annaba	Encadrant
BOUKARI Karima	M.C.A	Université de Badji Mokhtar Annaba	Examineur

Année Universitaire : 2021/2022

Remerciement

Tout d'abord, nous remercions Dieu notre Créateur de nous avoir donné la force, la volonté et le courage d'accomplir ce travail.

Nous tenons à remercier notre encadrante, Mme Daas

Pour ses conseils tout au long de ce travail.

Nous tenons également à remercier les membres du jury pour l'honneur qu'ils nous ont fait en acceptant de siéger à notre soutenance.

Enfin, nous tenons à exprimer notre profonde gratitude à nos familles et notre ami Riad qui nous ont toujours soutenus et à tous ceux qui ont contribué à la réalisation de cette mission.

Ainsi que tous les enseignants qui ont contribué à notre formation.

Sommaire

1.	Introduction :	4
1.1.	Transmission de la COVID-19 :	4
1.2.	Qu'est-ce que la COVID-19 ?	5
1.3.	La COVID-19, une catastrophe spécifique :	6
1.4.	L'impact de la COVID-19 :	7
1.5.	Modes de transmission de la COVID-19 :	10
1.6.	COVID-19 : Prévention et risques :	11
1.7.	Masque de protection de la COVID-19 :	12
1.8.	Type de masque :	13
1.8.1.	Masque jetable : chirurgical	13
1.8.2.	Masque lavable : en tissu "Grand Public" :	15
1.8.2.1.	Masque de catégorie 1 (UNS1) :	15
1.8.2.2.	Le Masque de catégorie 2 (UNS2) :	16
1.9.	Justification du port de masques :	17
	Conclusion :	18
2.	Introduction :	20
2.1.	La biométrie :	20
2.2.	Reconnaissance :	21
2.2.1.	Reconnaissance d'objets (Également appelée classification d'objets) :	21
2.2.2.	Identification :	21
2.2.3.	Detection :	21
2.3.	Réseaux neurones :	22
2.4.	Computer vision :	25
2.5.	Deep learning :	25
2.6.	Raspberry Pi et Camera :	26
2.7.	Python :	28
2.8.	Open CV :	29
2.9.	Visual Studio Code :	30
2.10.	Colab :	31
2.11.	Les avantages et les inconvénients des deux plateformes Colab et Visual Studio :	31

Conclusion :	32
3. Introduction :	34
3.1. Méthodologie proposée :	34
3.2. L'approche de « Face Mask Detection Algorithm » :	35
3.3. Système de détection de mask facial :	36
3.4. Mise en œuvre du système :	38
3.4.1. Phase de Train :	38
3.4.1.1. Train_Mask_Detector.py :	38
3.4.1.2. Réglage fin avec Keras et TensorFlow :	38
3.4.1.3. Base de données des masques :	40
3.4.1.4. Data augmentation :	42
3.4.1.5 MobileNet2 :	43
3.6 Phase Test :	50
3.6.1. Detect_Mask_Video.py :	50
Conclusion :	59
Conclusion Générale :	60

Liste des figures

Figure 1: Transmission de la COVID-19	5
Figure 2: photo approximative de la COVID-19.....	6
Figure 3: catastrophe de la COVID-19	7
Figure 4: le confinement durant la propagation du virus covid-19.....	8
Figure 5: L'enseignement à distance.....	9
Figure 6 : Les modes de transmissions de la covid-19	11
Figure 7 : Les masque de protection de la covid-19.....	13
Figure 8 : Masque jetable.....	14
Figure 9 : Masque de catégorie 1 (UNS1).....	16
Figure 10 : Masque de catégorie 2 (UNS2).....	17
Figure 11 : Niveau d'efficacité d'un masque	18
Figure 12 : Représentation des Réseaux neurones.....	24
Figure 13 : Schéma d'un neurone artificiel	24
Figure 14 : La caméra Raspberry Pi Rev1.3	28
Figure 15 : Block Diagramme.....	35
Figure 16 : Comparaison librairies apprentissage automatique.	39
Figure 17 : Base de données des masques.....	41
Figure 18: Exemple de data augmentation	43
Figure 19 : Détection de masque facial architecture de modèle basé sur CNN.....	44
Figure 20 : (a) la structure de base de MobileNet : (b) unité dans le MobileNet.	45
Figure 21 : Classificateur de masque facial construit à l'aide de l'architecture MobileNetV2.	46
Figure 22 : Extrait de l'architecture globale	46
Figure 23 : Graph de training loss et accuracy	50

Liste des Tableaux

Tableau 1 : Raspberry Pi 3 VS Raspberry Pi 4.....	27
Tableau 2 : Les avantages et les inconvenants des deux plateformes Colab et Visual Studio.....	31
Tableau 3 : Performances dans les tests en temps réel.....	59

Résumé :

Le personnel médical porte un masque pour prévenir la propagation de la maladie. Aujourd'hui, avec la pandémie de coronavirus, tout le monde doit porter un masque facial pour la même raison. Quand une personne proche de vous tousse, parle, éternue, il pourrait libérer des germes dans l'air qui peuvent vous infecter ou n'importe qui à proximité. Le port d'un masque fait partie d'une stratégie de contrôle des infections visant à éviter et à éliminer la contamination croisée. Même ainsi, les gens deviennent fatigués de porter des masques ou ils ne sont pas assez conscients de la gravité de la covid19 réelle.

Dans cet travaille (Project de Fin d'étude), nous proposons un détecteur de masque basé sur des dispositifs de système embarque et un algorithme d'apprentissage profond. Notre principal objectif est d'avertir les gens en temps réel s'ils ne portent pas de masque ou s'ils ne le portent pas correctement. La solution proposée génère des alertes après la détection d'un manque de respect de l'usure du masque en temps réel pour une réaction rapide. Pour avoir le détecteur le plus efficace en temps réel, nous avons testé le modèle de détection du masque en utilisant différentes versions du Raspberry Pi et du PC portable (Visual Studio Code) ainsi que des plateforme internet (google Colab). En conséquence, le détecteur de masque fonctionne parfaitement sur les appareils puissants, mais ses performances diminuent en temps réel en utilisant des appareils moins puissants tels qu'une ancienne version du Raspberry Pi.

Mots clés :

COVID-19 ; intelligence artificielle ; détection de masque facial ; apprentissage en profonde ; Raspberry Pi

Abstract:

Medical personnel wear masks to prevent the spread of the disease. Today, with the coronavirus pandemic, everyone has to wear a face mask for the same reason. When someone close to you coughs, talks, or sneezes, it could release germs in the air that can infect you or anyone nearby. Wearing a mask is part of an infection control strategy to avoid and eliminate cross-contamination. Even so, people become tired of wearing masks or they are not sufficiently aware of the severity of the actual covid19.

In this work, we propose a mask detector based on embedded system devices and a deep learning algorithm. Our main objective is to warn people in real-time if they are not wearing a mask or if they are not wearing it properly. The proposed solution generates alerts after the detection of a lack of respect for the wear of the mask in real time for a quick reaction. In order to have the most efficient detector in real-time, we tested the mask detection model using different versions of the Raspberry Pi and portable PC (Visual Studio Code) as well as internet platforms (google Colab). As a result, the mask detector works perfectly on powerful devices, but its performance decreases in real-time using less powerful devices such as an older version of the Raspberry Pi.

Keywords:

COVID-19; Artificial intelligence; face mask detection; deep learning; Raspberry-pi

ملخص

يرتدي الطاقم الطبي أقنعة لمنع انتشار المرض. اليوم، مع جائحة فيروس كورونا، يتعين على الجميع ارتداء قناع الوجه لنفس السبب. عندما يسعل شخص قريب منك أو يتحدث أو يعطس، يمكن أن يطلق جراثيم في الهواء يمكن أن تصيبك أو تصيب أي شخص قريب. يعد ارتداء القناع جزءًا من استراتيجية مكافحة العدوى لتجنب التلوث المتبادل والقضاء عليه. ومع ذلك، سئم الناس من ارتداء الأقنعة أو أنهم لا يدركون بما فيه الكفاية شدة الكوفيد19 الفعلية.

في هذا العمل، نقترح كاشفًا للأقنعة يعتمد على أجهزة النظام المضمنة وخوارزمية التعلم العميق. هدفنا الرئيسي هو تحذير الناس في الوقت الفعلي إذا كانوا لا يرتدون قناعًا أو إذا لم يرتدوه بشكل صحيح. يولد الحل المقترح تنبيهات بعد اكتشاف عدم احترام ارتداء القناع في الوقت الفعلي لرد فعل سريع. من أجل الحصول على أكثر أجهزة الكشف كفاءة في الوقت الفعلي، اختبرنا نموذج الكشف عن الأقنعة باستخدام إصدارات مختلفة من Raspberry Pi والكمبيوتر الشخصي المحمول (Visual Studio Code) بالإضافة إلى منصات الإنترنت (Google Colab) نتيجة لذلك، يعمل كاشف القناع بشكل مثالي على الأجهزة القوية، لكن أدائه ينخفض في الوقت الفعلي باستخدام أجهزة أقل قوة مثل نسخة قديمة من Raspberry Pi.

الكلمات المفتاحية:

كوفيد-19 ذكاء اصطناعي؛ كشف قناع الوجه؛ التعلم العميق؛ Raspberry Pi

Introduction générale

La pandémie causée par le COVID-19 a tenu la communauté mondiale éveillée pendant l'année précédente. Ces derniers temps, une nation avec un produit intérieur brut fort comme l'Inde a rapporté plus de 3,89 crores [1]. Compte tenu de ces faits, les experts consacrent des efforts considérables à la création de nouveaux remèdes à la catastrophe actuelle. En outre, les dossiers de l'OMS indiquent que des efforts importants ont été déployés par de nombreux pays développés et en développement avec la fourniture de masques, de respirateurs, de blocs opératoires, de masques faciaux et d'autres équipements de santé essentiels. Dans ce contexte, l'OMS a élaboré des réglementations strictes pour les patients COVID-19, les couvre-feux, l'isolement social et le dépistage. De plus, de nombreux pays ont intégré des outils numériques qui permettent aux citoyens de distinguer les patients COVID-19 à proximité via des technologies de suivi de position sans fil. Cependant, à la lumière de l'effet et de la prolifération du COVID-19, les chercheurs ont consacré de nombreuses ressources à la découverte et à l'amélioration des remèdes au COVID-19 ; de nombreuses nouvelles solutions ont été proposées pour la situation désastreuse existante, notamment des dispositifs de désinfection automatisés pour la désinfection des fournitures médicales et des humains, des dispositifs d'imagerie infrarouge et autres. De manière significative, plusieurs gouvernements judiciaires ont adopté des politiques de prévention sanitaire strictes et ont fait de sérieuses tentatives pour nettoyer des régions topographiques spécifiques avec une variété de désinfectants, en plus de mener divers efforts de sensibilisation à la santé pour promouvoir la santé de la société. Dans de rares cas, les gouvernements ont également pénalisé des individus pour avoir tenté d'enfreindre les réglementations en matière de santé et de sécurité. Il est encore discutable de savoir si oui ou non le strict respect des exigences de santé et de sécurité telles que le port du masque et le maintien de la distance sociale est observé. Les transports en commun mettent les gens de la communauté en contact direct avec tout le monde, souvent pour de plus longues durées, et en les soumettant à des zones régulièrement touchées, augmentent les chances d'une personne de contracter et de transmettre le COVID-19. Maintenir une distance de six pieds avec les gens est souvent difficile dans les transports en commun. Les personnes peuvent ne pas être en mesure de maintenir la distance requise de six pieds par rapport aux autres passagers assis à côté ou aux personnes se promenant ou transitant par les gares routières. Des affaires aux interactions sociales, tous les types d'activités hors et sur écran sont nécessaires dans les transports publics ainsi que dans les entrées publiques pour assurer le bien-être social. En conséquence, un

dispositif d'entrée intelligent est nécessaire qui reconnaît en permanence la présence d'un masque au niveau du mécanisme d'ouverture de la porte.

À notre connaissance, il n'existe pas de système complet d'identification et de surveillance de la détection des masques faciaux COVID-19 dans les transports publics utilisant les technologies intelligence artificiel.

Ce travail présente un système de détection de masque facial basé sur intelligence artificiel, grâce à la collecte de données en temps réel via des dispositifs de reconnaissance faciale. Les principales contributions de travail sont les suivantes :

- L'utilisation des techniques d'apprentissage en profondeur pour classer les images avec et sans masque facial et détecter la présence de masques faciaux dans le streaming vidéo en temps réel.
- Le développement d'un système de détection de masque facial efficace et peu coûteux pouvant être utilisé dans les transports en commun.
- Testé le modèle de détection du masque en utilisant différentes versions du Raspberry Pi

Ce qui suit définit l'organisation de cette mémoire. Port du masque dans le cadre de la COVID-19 est fourni au CHAPITRE 1. CHAPITRE 2 présente une explication détaillée de e traitement d'images pour détecter le port du masque contre le COVID-19. Une explication détaillée de la méthodologie du travail proposé dans le CHAPITRE 3. Enfin, une conclusion, ainsi que des recommandations pour les travaux futurs, sont incluses dans le dernier CHAPITRE.

Chapitre1

Port du masque dans le cadre de la COVID-19

1. Introduction :

Covid-19 la pandémie qui menace le monde et a forcé de nombreux peuples à renoncer à leur vie normale, ce virus a menacé la vie de nombreuses personnes et en a tué beaucoup d'autres en raison de sa nature et de son fonctionnement en attaquant le système respiratoire chez l'homme causant des problèmes respiratoires et donc la mort.

Les gens et les gouvernements réalisent un rôle important de la protection et décident donc de prendre des mesures en forçant les gens à faire certains actes qui peuvent réduire l'infection et éventuellement sauver des vies, comme le port de masques.

1.1. Transmission de la COVID-19 :

Tout d'abord, il est important de définir la transmission « aéroportée » et « par gouttelettes ». Les gouttelettes font référence à de grosses particules (supérieures à 5 microns) qui tombent au sol au niveau des particules en suspension dans l'air, également appelées noyaux de bronchoscopie et d'intubation - les AGMI mentionnés ci-dessus - qui peuvent générer ces particules en suspension dans l'air et exposer les travailleurs de la santé à un plus grand risque d'infection. Il est également important de noter que malgré la présentation binaire des termes « particules en suspension dans l'air » et « gouttelettes », le fait est qu'ils contiennent des particules respiratoires allant de 0,1 à 100 microns (c'est-à-dire à des temps d'aérosol malgré le consensus sur la taille des particules Il en résulte des recommandations contradictoires sur l'utilisation d'équipements de protection individuelle pour les travailleurs de la santé pour lutter contre ce virus, recommande le port de masques médicaux (chirurgicaux) pour les soins aux patients non **IMGA**. Il est également important de noter que la transmission par contact, c'est-à-dire la main ou la contamination de l'environnement se transmet de cette façon ; cependant, ce mode de transmission dans les hôpitaux consiste généralement à prendre des mesures pour démontrer la prédominance d'un mode de transmission sur un autre. Ils n'incluent que la simulation des sécrétions respiratoires du SRAS-CoV-2 provenant de la toux de patients malades. Source au destinataire.

Mais où il est difficile de contrôler les variables étrangères. Direct et permettant ainsi des inférences plus robustes, des voies de transmission dans les études [1].

CHAPITRE 1 Port du masque dans le cadre de la COVID-19

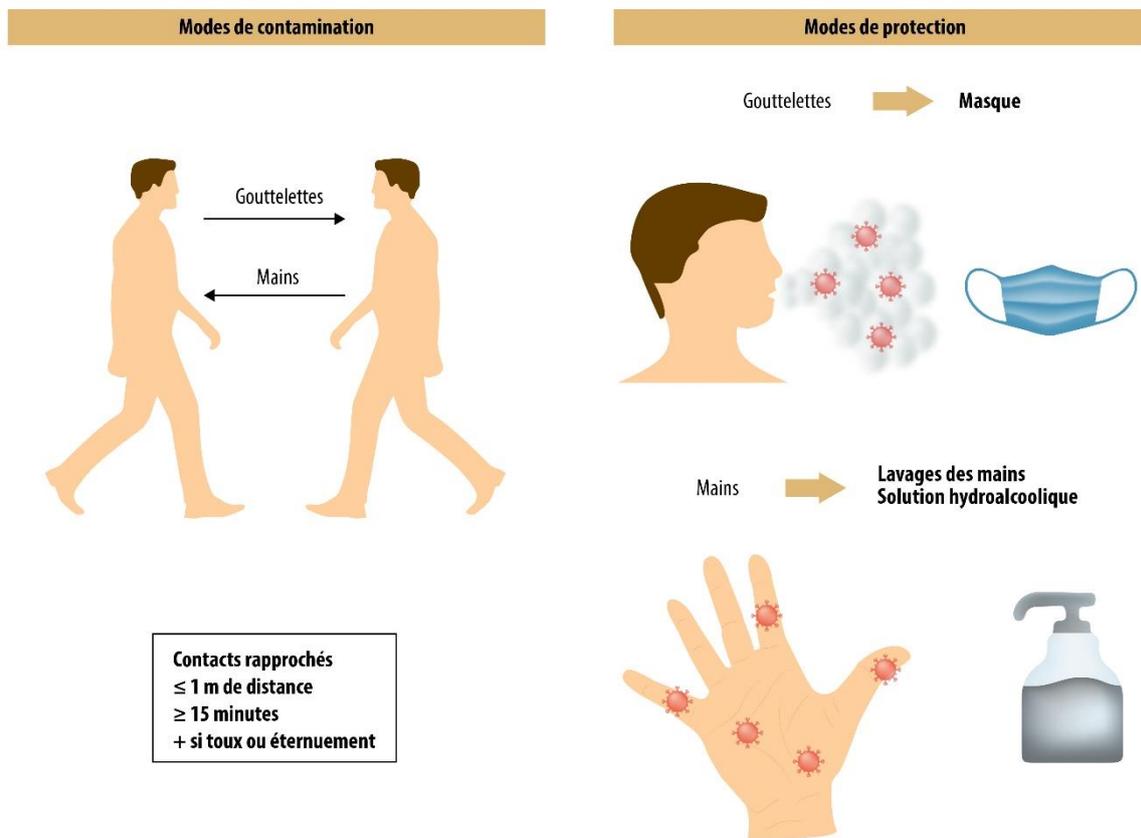


Figure 1: Transmission de la COVID-19

1.2. Qu'est-ce que la COVID-19 ?

Est une pandémie d'une maladie infectieuse émergente, appelée la maladie à coronavirus 2019 ou Covid-19, provoquée par le coronavirus SARS-CoV-2. Elle apparaît le 16 novembre 2019 à Wuhan⁴, dans la province de Hubei (en Chine centrale), avant de se propager dans le monde.

L'Organisation mondiale de la santé (OMS) alerte dans un premier temps la République populaire de Chine et ses autres États membres, puis prononce l'état d'urgence de santé publique de portée internationale le 30 janvier 2020.

Le 11 mars 2020, l'épidémie de Covid-19 est déclarée pandémie par l'OMS qui demande des mesures de protection essentielles pour prévenir la saturation des services de soins intensifs et pour renforcer l'hygiène préventive (suppression des contacts physiques, bises et poignées de mains, fin des attroupements ainsi que des déplacements et voyages non indispensables,

promotion du lavage des mains, mise en application de quarantaine, etc.). Pour freiner la formation de nouveaux foyers de contagion et préserver les capacités d'accueil de leurs hôpitaux, de nombreux pays décident des mesures de confinement, la fermeture de leurs frontières et l'annulation des manifestations sportives et culturelles. Ces décisions ont des conséquences économiques, sociales et environnementales et font peser des incertitudes et des craintes sur l'économie mondiale et sur l'éducation, la santé et les droits fondamentaux des populations [2].

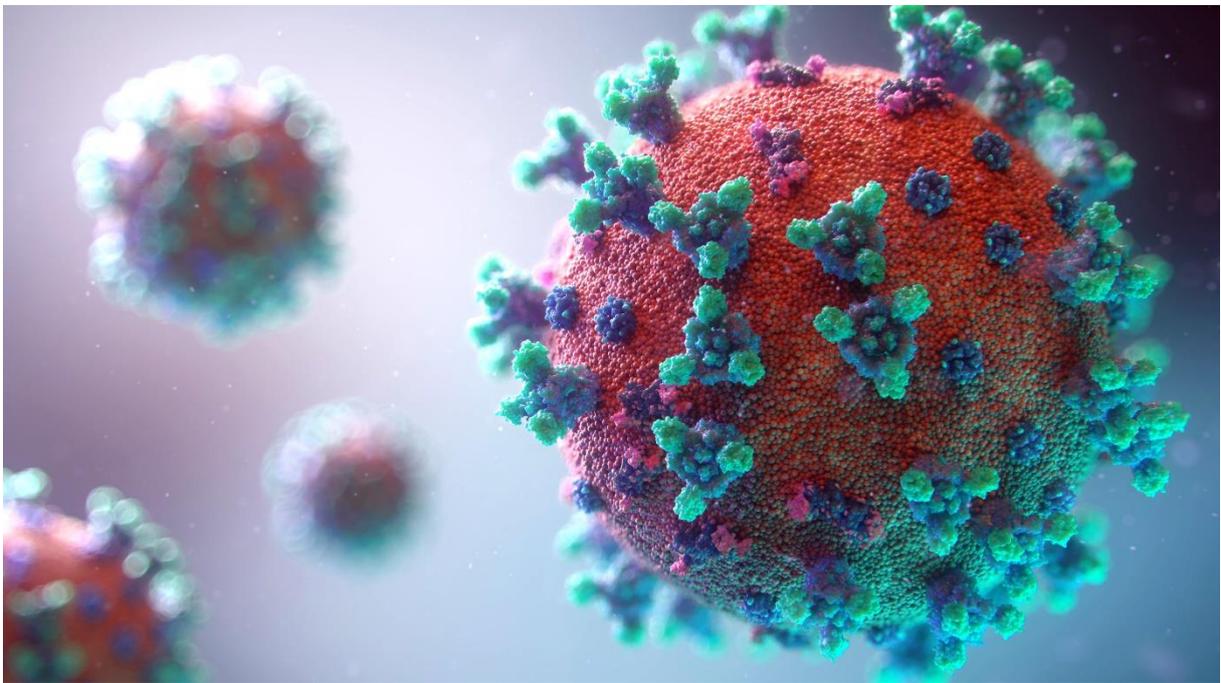


Figure 2: photo approximative de la COVID-19

1.3. La COVID-19, une catastrophe spécifique :

La maladie à coronavirus-19 est considéré comme catastrophe parce qu'il a tué plus de 6 millions de personnes dans le monde il a laissé une empreinte indélébile dans l'histoire de l'humanité. Étant très sensibles aux mutations, de nombreuses variantes du virus ont commencé à apparaître, et certaines étaient plus mortels et contagieux que le précédent. L'efficacité du

vaccin est également sévèrement affectée contre le nouveau une variante. L'infectiosité et de l'évasion immunitaire du virus. Comme c derniers temps la chine a déjà fait face à une nouvelle vague de la pandémie qui est le virus o microns qui est très puissants et se propage rapidement. Où l'organisation mondiale de la santé recommande fortement de respecter les gestes barrières pour limiter la propagation du virus [3].



Figure 3: catastrophe de la COVID-19

1.4. L'impact de la COVID-19 :

Face à la pandémie causée par le coronavirus, les autorités d'un nombre important de pays ont pris des mesures sans précédent pour limiter la propagation du virus.

Elles ont imposé, des changements majeurs dans les comportements habituels des individus comme des entreprises.

Si les conditions et la durée du confinement varient d'un pays à l'autre, plus de 2 milliards d'individus se sont retrouvés ou sont encore confinés à leur domicile, avec l'interdiction d'effectuer des déplacements considérés comme non essentiels. Ces mesures restrictives ont contribué à limiter le nombre de personnes admises en soins intensifs et en réanimation à la

suite d'une infection par le virus. Cependant, elles ont aussi eu, dans le même temps, un impact dramatique sur le plan sanitaire, Social et économique. Nous considérons tout d'abord la mise en place des Mesures d'hygiène et de distance, qui a appelé la population à des changements de comportements individuels de différents niveaux de difficulté. Tel que l'adoption du lavage des mains, la limitation de l'utilisation des transports publics, et des contacts sociaux. Ils montrent l'existence de barrières individuelles à adopter de nouveaux comportements.

Les études montrent aussi que depuis les premiers jours du confinement que le mal-être psychologique est d'autant plus important que les individus ont réduit leurs relations sociales du fait du confinement. Ils montrent aussi que la santé mentale des individus a été fortement affectée par des changements de revenus et des conditions de travail.

En effet, les entreprises n'ont pu poursuivre normalement leur activité durant le confinement.



Figure 4: le confinement durant la propagation du virus covid-19

Les conséquences à long terme des mesures de confinement sur les entreprises ont été très grave au niveaux économique les chercheurs montrent comment certains individus ou entreprises existantes ont adapté leur activité voire ont généré une nouvelle activité pour affronter la crise.

Enfin, un autre secteur de la société, qui a dû s'adapter à la situation, est celui de l'enseignement qui a modifié les modes d'enseignement pendant le confinement en faisant les cours et les examens a des distances [4].



Figure 5: L'enseignement à distance

1.5. Modes de transmission de la COVID-19 :

La COVID-19 se transmet d'une personne infectée à d'autres par le biais de gouttelettes respiratoires et d'aérosols dont la taille peut varier, tels que :

Grosses gouttelettes qui tombent rapidement au sol (en quelques secondes ou minutes) près de la personne infectée

Gouttelettes plus petites, parfois appelées aérosols, qui persistent dans l'air, en particulier dans les espaces intérieurs

Ces gouttelettes et aérosols respiratoires sont créés lorsqu'une personne infectée :

- Respire
- Pour parler
- Chante
- Des cris
- Toux
- Éternue

Le COVID-19 peut également se propager en touchant quelque chose qui contient le virus, puis en se touchant la bouche, le nez ou les yeux avec des mains non lavées.

Vous pouvez transmettre la COVID-19 avant de commencer à présenter des symptômes ou sans jamais développer de symptômes [5] .

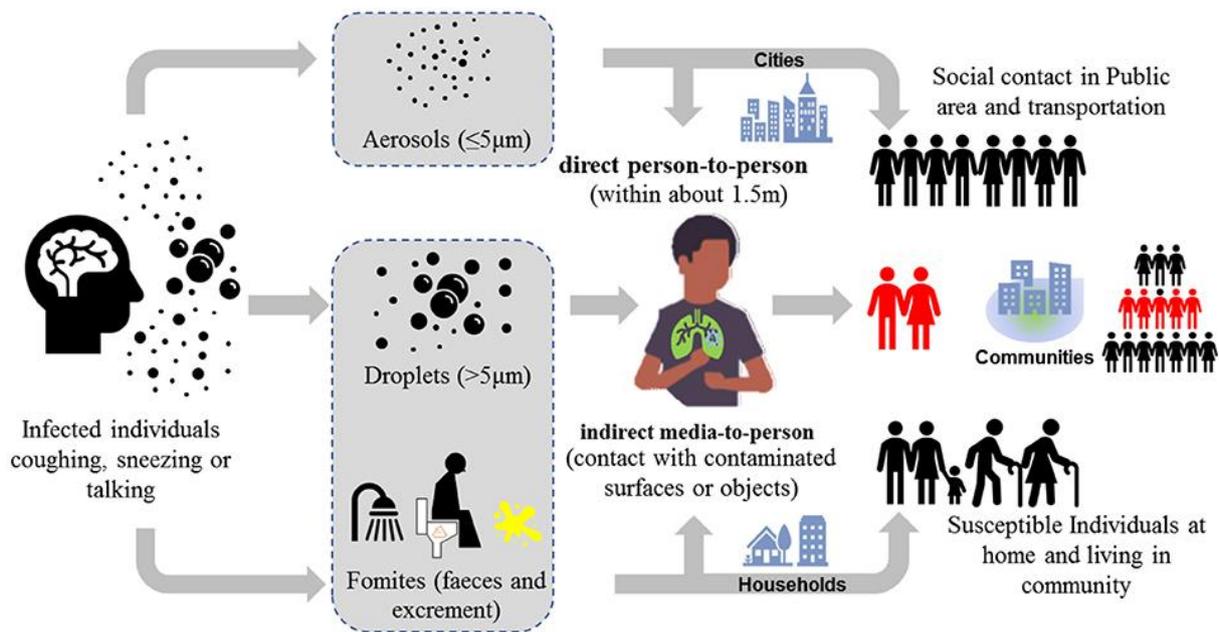


Figure 6 : Les modes de transmissions de la covid-19

1.6. COVID-19 : Prévention et risques :

La prévention vise essentiellement à éviter la contagion en suspendant tout contact proche avec une personne infectée confirmée et en limitant très fortement les autres contacts. Ceci passe notamment par l'instauration de mesures-barrières comme la distanciation physique, l'arrêt des embrassades et des échanges manuels, le lavage fréquent des mains, l'utilisation de mouchoirs jetables ou encore la pratique de tousser dans son coude pour éviter la production d'aérosols, la désinfection des surfaces potentiellement contaminées telles que des poignées de portes, boutons d'ascenseurs, etc. L'aération des locaux s'ajoute à ces comportements-barrière.

Il est vivement conseillé aux personnes présentant un état grippal de surveiller leur température.

« Les personnes âgées de 70 ans et plus (même si les patients entre 50 ans et 70 ans doivent être surveillés de façon plus rapprochée) ;

Les patients présentant une insuffisance rénale chronique dialysée, insuffisance cardiaque stade NYHA III ou IV ;

Les malades atteints de cirrhose au stade B au moins ;

Les patients aux antécédents (ATCD) cardiovasculaires : hypertension artérielle, ATCD d'accident vasculaire cérébral ou de coronaropathie, chirurgie cardiaque ;

Les diabétiques insulino-dépendants ou présentant des complications secondaires à leur pathologie (micro ou macro angiopathie).

Les insuffisants respiratoires chroniques sous oxygénothérapie ou asthme ou mucoviscidose ou toute pathologie chronique respiratoire susceptible de décompenser lors d'une infection virale.

Les personnes avec une immunodépression :

- Médicamenteuses : chimiothérapie anti cancéreuse, immunosuppresseur, biothérapie et/ou une corticothérapie à dose immunosuppressive ;
- Infection à VIH non contrôlé avec des CD4 < 200 par mm³ ;
- Consécutive à une greffe d'organe solide ou de cellules souche hématopoïétiques ;
- Atteint d'hémopathie maligne en cours de traitement ;
- Présentant un cancer métastasé ;
- Les femmes enceintes ;

Les personnes présentant une obésité morbide (indice de masse corporelle > 40 kg m⁻²) : par analogie avec la grippe A(H1N1). » [6].

1.7. Masque de protection de la COVID-19 :

Qu'il soit jetable (chirurgical) ou en tissu lavable et donc, réutilisable, le masque reste obligatoire dans tous les lieux publics clos pour se protéger du coronavirus (et même si on a été vacciné). En extérieur, le port du masque n'est plus obligatoire depuis le 17 juin 2021 sauf dans les lieux bondés, les files d'attentes, les tribunes de stade ou les marchés... Il en existe différents modèles dont des transparents (masques inclusifs) pour permettre de lire sur les lèvres de son interlocuteur et de mieux percevoir ses émotions. En raison de la propagation de nouveaux variants très contagieux, le ministre de la Santé, suivant l'avis du Haut conseil de santé publique (HCSP), recommande depuis janvier 2021 de ne plus porter les masques en tissu fabriqués à la maison ainsi que les masques industriels de catégorie 2 car ils ne sont pas assez filtrants. A l'école, les élèves doivent exclusivement porter un masque chirurgical ou en tissu de catégorie 1 à l'intérieur. Même chose dans les entreprises. [7].

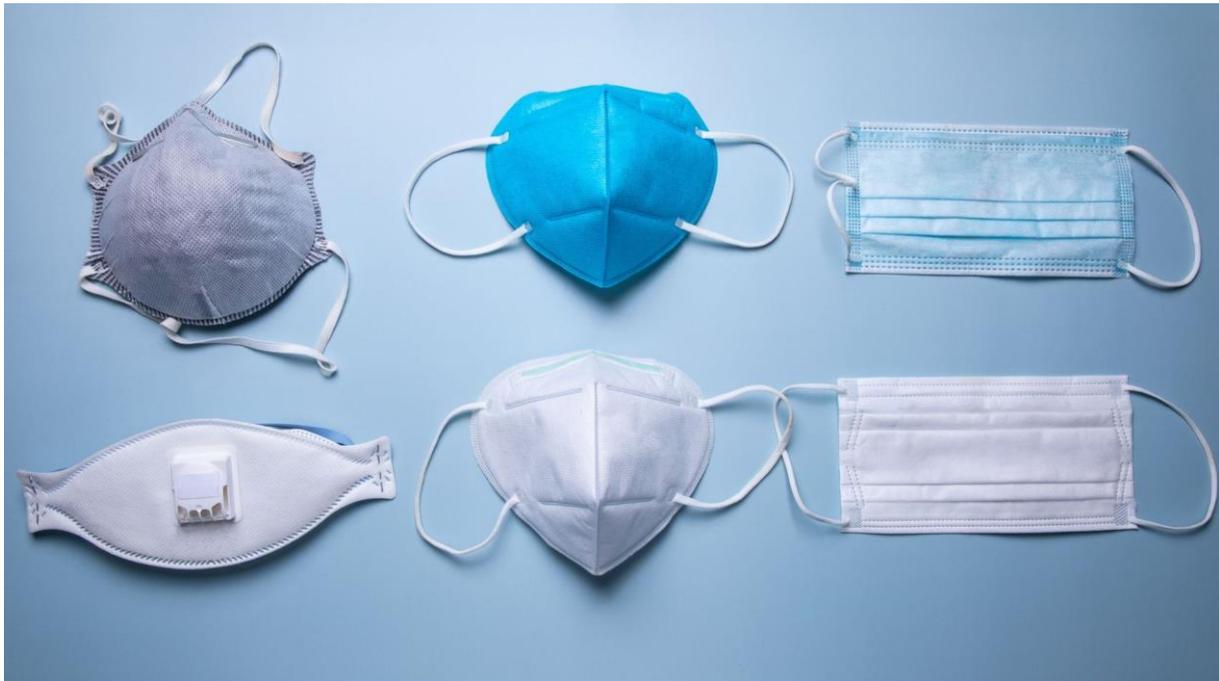


Figure 7 : Les masque de protection de la covid-19

1.8. Type de masque :

Le masque de protection est un dispositif médical destiné à filtrer les bactéries et à éviter de contracter un virus, comme celui de la grippe ou toute autre maladie virale. "Ces masques sont généralement portés au bloc opératoire pour éviter d'abord que les bactéries de la bouche du chirurgien ne soient projetées sur plaie chirurgicale du patient, mais peuvent aussi être portés par le grand public pour se protéger des micro-organismes dans un contexte d'épidémie, comme la grippe ou le Covid-19 par exemple", indique le Dr Pierre Parneix. Il existe plusieurs types de masque qui ont des niveaux de filtration variables. "Pour définir le niveau de filtration, on réalise un test avec un aérosol contenant des souches de staphylocoque doré", explique le spécialiste.

1.8.1. Masque jetable : chirurgical

Le masque chirurgical, parfois en forme de bec de canard, évite la projection de gouttelettes émises par le porteur du masque. Il peut être de type I qui filtre 95% des bactéries ou de type II qui filtre plus de 98% des bactéries et de type IIR encore plus étanches et résistants aux projections. Ils sont à usage unique et donc jetables au bout de 3 à 4 heures d'utilisation. "Au-delà, il faut le jeter à la poubelle. Le masque est à usage unique et en aucun cas réutilisable.

Après chaque sortie, dès le retour à la maison, il ne faut pas le manipuler et le jeter car il est potentiellement contaminé", prévient le Dr Parneix.

Seuls les masques chirurgicaux à usage unique, respectant la norme EN 14683 (avec une face bleue et une face blanche) sont recommandés par le Haut Conseil de la Santé Publique [7].



Figure 8 : Masque jetable

Les masques FFP1, FFP2, FFP3 :

Un masque FFP est un appareil de protection respiratoire destiné à protéger celui qui le porte contre l'inhalation à la fois de gouttelettes et de particules en suspension dans l'air. Le port de ce type de masque est plus contraignant (inconfort thermique, résistance respiratoire) que celui d'un masque chirurgical. Il est réservé aux professionnels de la santé de toute la chaîne (transport sanitaire, pompiers, médecins libéraux, infirmiers...).

La différence des masques de protection coronavirus :

Le masque FFP1 filtre au moins 80 % des aérosols de taille moyenne 0,6 μm .

Le masque FFP2 filtre au moins 94 % des aérosols de taille moyenne 0,6 μm .

Le masque FFP3 filtre au moins 99 % des aérosols de taille moyenne 0,6 μm .

Le masque de type FFP a une durée de protection de 8 heures. Au-delà, il doit être jeté.

1.8.2. Masque lavable : en tissu "Grand Public" :

Les masques dits "grand public" développés dans le cadre de l'épidémie de Covid-19 sont des masques en tissu, la plupart du temps lavables et réutilisables. Ils ont des propriétés de filtration allant d'au moins 70% à plus de 90% de filtration des particules émises d'une taille égale ou supérieure à 3 microns, en fonction de la catégorie à laquelle il appartient (UNS1 ou UNS2). La mention de la catégorie figure sur l'emballage ou la notice. Un masque grand public aux normes doit présenter ce logo officiel bleu, blanc et rouge (à droite) avec en sous-texte son nombre de lavages maximum. Il y a deux types de masques en tissu industriels testés par un laboratoire et respectant les normes de l'Afnor :

1.8.2.1. Masque de catégorie 1 (UNS1) :

Conseillé par le Haut Conseil de la Santé publique car il est assez filtrant. Il protège des projections directes de gouttelettes avec une efficacité de filtration supérieure à 90 % pour des particules émises d'une taille de 3 μm . Suivant un avis du Haut conseil de santé publique publié le 20 janvier 2021, un décret du 27 janvier 2021 recommande de ne commercialiser que des masques "grand public" relevant de cette catégorie, en les affichant comme tels, sous réserve qu'ils aient été testés au préalable [7].



Figure 9 : Masque de catégorie 1 (UNS1)

1.8.2.2. Le Masque de catégorie 2 (UNS2) :

: déconseillé par le Haut Conseil de la Santé Publique car il n'est pas assez filtrant. Ce masque protège des projections directes de gouttelettes avec une efficacité de filtration supérieure à 70 % pour des particules émises d'une taille de 3 μm . Il ne serait pas suffisant pour protéger de façon optimale du virus. Le décret du 27 janvier 2021 n'en interdit pas la fabrication, mais demande au fabricant d'informer le consommateur que ces masques ne répondent pas aux prescriptions des autorités sanitaires. AFNOR recommande d'éviter les masques UNS 2 [7].



Figure 10 : Masque de catégorie 2 (UNS2)

1.9. Justification du port de masques :

Le masque sert à éviter de contracter un virus, comme celui de la grippe, du Covid-19 ou de toute autre maladie virale. Mais aussi à protéger les autres si on est soi-même malade. Ainsi, une personne qui présente des symptômes d'infection respiratoire ou suspectée d'être malade peut porter un masque chirurgical pour protéger les autres de ses symptômes (à la maison, dans les lieux publics...). Le masque peut également servir dans un contexte de prévention de l'exposition au virus. Il a un "effet barrière" qui va empêcher le passage des particules bactériennes et virales. "Le mode de transmission du coronavirus est sensiblement le même que celui de la grippe, c'est-à-dire qu'il se transmet d'homme à homme lors de contacts rapprochés (se toucher ou se serrer la main par exemple) et par voie aérienne en toussant ou en éternuant (gouttelettes de salive, postillons).



Figure 11 : Niveau d'efficacité d'un masque

Conclusion :

Dans ce chapitre nous avons vu c'est quoi cette pandémie qui est la covid ces risques ces modes de transmissions et les gestes barrières pour lutter contre se virus qui principalement le port du masque et dans le prochain chapitre nous allons voir comment faire respecter le port du masque a l'aide d'une application électronique.

Chapitre 2

Le traitement d'images pour détecter le port du masque contre le COVID-19

2. Introduction :

Dans ce chapitre nous allons voir les principes de base biométrie et les outils utiliser pour créé une application de détection de masque à l'aide de Raspberry pi.

2.1. La biométrie :

Un système biométrique est un processus automatisé qui utilise un dispositif électronique pour saisir des identifiants biométriques, extrait des données biométriques données de l'identifiant soumis, compare l'identifiant avec les données capturées précédemment, fait correspondre l'identifiant capturé avec un modèle, détermine si l'identité fournie est authentique. Autrement dit, un système biométrique acquiert des données biométriques d'un individu et le compare à un modèle pour déterminer l'identité d'une personne. L'ajout de l'IA dans les systèmes d'identité biométriques grâce à la détection des visages peut être avantageux mais difficile en raison de multiples défis liés à l'apprentissage continu, à la prise de décision et aux problèmes de sécurité. Pour qu'un système biométrique fonctionne, des documents externes contenant des données biographiques, tels qu'un permis de conduire, doivent être présentés pour « l'inscription » au système biométrique. L'interaction initiale d'un individu.

Permettre la capture d'un identifiant biométrique tel qu'une empreinte digitale ou un œil scan ou prendre une photo avec un masque, qui est enregistré et lié au document externe présenté. Une fois ceux-ci connectés, la rencontre suivante d'un individu avec le système lorsque les données biométriques sont reprises seront par rapport à ce modèle au dossier, pris lors de la phase d'inscription. Soutiennent qu'un système biométrique idéal devrait considérer certains facteurs.

- Performance : consiste à atteindre une précision et une vitesse acceptables dans capturer la biométrie, les besoins en ressources et les facteurs opérationnels et environnementaux qui peuvent affecter cela.
- Acceptabilité : la mesure dans laquelle les individus sont disposés à autoriser l'utilisation d'une caractéristique spécifique.
- Contournement : la facilité avec laquelle le système biométrique peut être trompé par des méthodes frauduleuses.

De même que on suggère des facteurs presque identiques et des facteurs supplémentaires tels que :

- Évolutivité : capacité à fonctionner efficacement lorsque le nombre de personnes est augmenté
- Interopérabilité : relation à l'utilisation des mêmes normes communes, biométrique dispositif et données/résultats.
- Convivialité : la facilité avec laquelle le système peut être utilisé et apprendre à l'utiliser
- Confidentialité : techniques pour éviter le vol ou l'utilisation abusive des données biométriques personnelles Les données.[8]

2.2. Reconnaissance :

Le problème classique de la vision par ordinateur, du traitement d'image et de la vision artificielle consiste à déterminer si les données d'image contiennent ou non un objet, une caractéristique ou une activité spécifique.

2.2.1. Reconnaissance d'objets (Également appelée classification d'objets) :

Un ou plusieurs objets ou classes d'objets prédéfinis ou appris peuvent être reconnus, généralement avec leurs positions 2D dans l'image ou leurs poses 3D dans la scène. Blippar, Google Goggles et LikeThat fournissent des programmes autonomes qui illustrent cette fonctionnalité.

2.2.2. Identification :

Une instance individuelle d'un objet est reconnue. Les exemples incluent l'identification du visage ou des empreintes digitales d'une personne spécifique, l'identification de chiffres manuscrits ou l'identification d'un véhicule spécifique.

2.2.3. Détection :

Les données d'image sont numérisées pour une condition spécifique, Les exemples incluent la détection d'éventuelles cellules ou tissus anormaux dans les images médicales ou la détection d'un véhicule dans un système de péage routier automatique. La détection basée sur des calculs relativement simples et rapides est parfois utilisée pour trouver des régions plus petites de

données d'image intéressantes qui peuvent être analysées plus avant par des techniques de calcul plus exigeantes pour produire des techniques d'interprétation correctes, Actuellement, les meilleurs algorithmes pour de telles tâches sont basés sur des réseaux de neurones convolutifs. Une illustration de leurs capacités est donnée par le défi de reconnaissance visuelle à grande échelle ImageNet ; Il s'agit d'une référence en matière de classification et de détection d'objets, avec des millions d'images et 1 000 classes d'objets utilisées dans le cadre du concours, Les performances des réseaux de neurones convolutifs sur les tests ImageNet sont désormais proches de celles des humains, Les meilleurs algorithmes ont encore du mal avec des objets petits ou fins, comme une petite fourmi sur une tige de fleur ou une personne tenant une plume à la main. Ils ont également des problèmes avec les images qui ont été déformées par des filtres (un phénomène courant avec les appareils photo numériques modernes). En revanche, ce genre d'images trouble rarement les humains. Les humains, cependant, ont tendance à avoir des problèmes avec d'autres problèmes. Par exemple, ils ne sont pas bons pour classer les objets en classes à grain fin, comme la race particulière d'animal ou l'espèce d'oiseau, alors que les réseaux de neurones convolutifs gèrent cela avec facilité.[9]

2.3. Réseaux neurones :

Les réseaux de neurones artificiels (ANN) ont été développés en tant que généralisations de modèles mathématiques de systèmes neuronaux biologiques.

La première vague d'intérêt pour les réseaux de neurones a émergé après l'introduction des neurones simplifiés par McCulloch et Pitts (1943) également connus sous le nom de modèles connectés.

Un réseau de neurones artificiels est un réseau de groupes de processeurs très simples ("neurones") chacun avec une (petite quantité de) mémoire locale.

Les unités fonctionnent uniquement sur leurs propres données locales et sur les entrées qu'elles reçoivent via des connexions ou des liens unidirectionnels.

L'unité de réseau contient une règle de collecte des signaux entrants et une règle de calcul du signal de sortie qui est ensuite envoyé à d'autres unités de réseau.

Selon Callen, les règles de calcul de la sortie sont connues sous le nom de fonction d'activation.

Un réseau de neurones se compose de trois couches dans sa structure.

La première couche est la couche d'entrée qui interagit directement avec les mondes externes ; La deuxième couche est une unité cachée où le calcul est effectué selon la fonction fournie, et la dernière couche est la couche de sortie à partir de laquelle nous obtenons la sortie.

Les connaissances sont stockées dans des réseaux de neurones sous forme de poids imbriqués entre les neurones.

Le réseau propage les données d'entrée d'une couche à l'autre jusqu'à ce que les données de sortie soient générées.

Si le réseau est multicouche avec l'algorithme de rétropropagation et que la sortie est différente de la sortie requise, l'erreur est calculée et propagée à travers le réseau.

Les poids de serrage sont ajustés avec la propagation des erreurs.

Aujourd'hui, les chercheurs tentent de construire un réseau électronique à base de silicium inspiré de la fonction et de la forme du cerveau humain.

La généralisation est la seule capacité qui fait de l'ANN un outil puissant. Comment fonctionne le cerveau humain, il apprend à créer des schémas et à s'en souvenir. De même, les réseaux de neurones sophistiqués ont la capacité d'apprendre et de mémoriser des modèles.

Les méthodes d'apprentissage dans les réseaux de neurones peuvent être classées en trois catégories.

Il s'agit de l'apprentissage supervisé, de l'apprentissage non supervisé et de l'apprentissage par renforcement.[10]

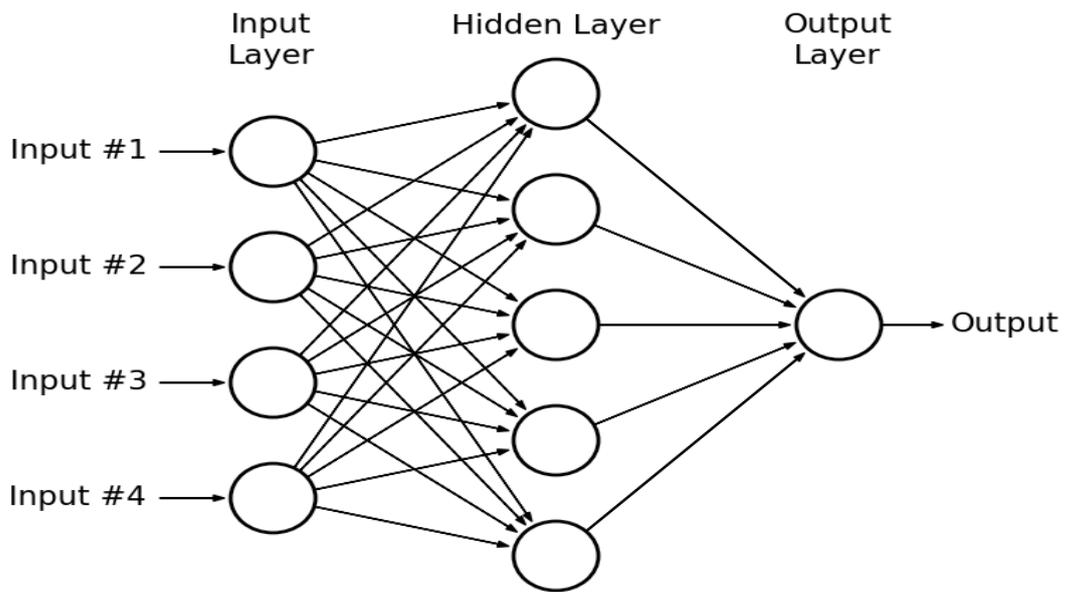


Figure 12 : Représentation des Réseaux neurones

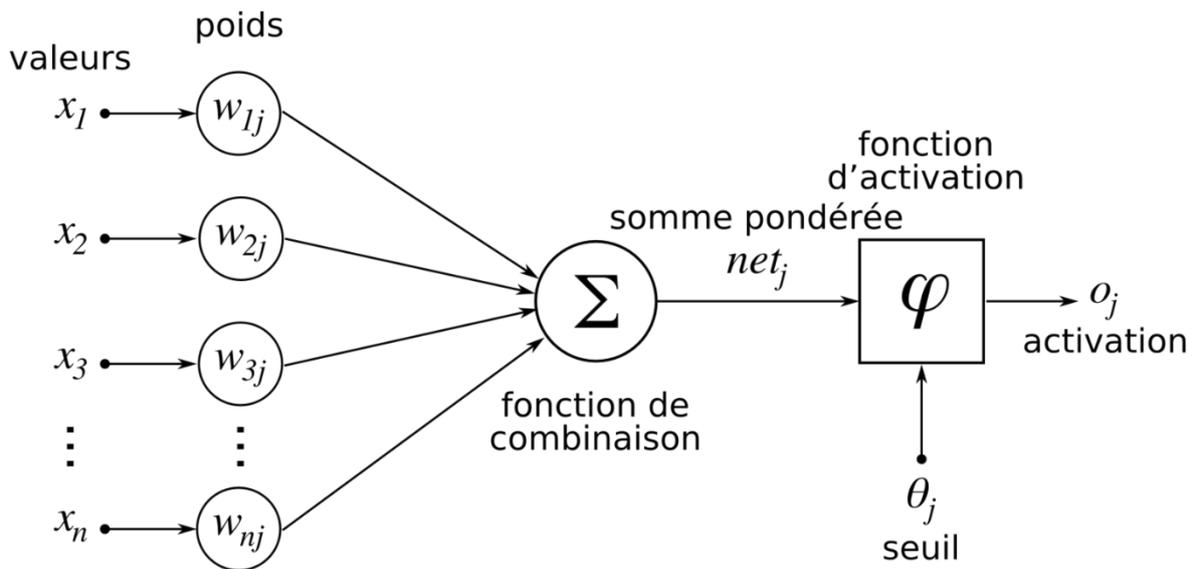


Figure 13 : Schéma d'un neurone artificiel

2.4. Computer vision :

La vision par ordinateur est un domaine interdisciplinaire qui traite de la façon dont les ordinateurs peuvent être conçus pour acquérir une compréhension de haut niveau à partir d'images ou de vidéos numériques. Du point de vue de l'ingénierie, il cherche à automatiser les tâches que le système visuel humain peut effectuer.

La vision par ordinateur concerne l'extraction, l'analyse et la compréhension automatiques d'informations utiles à partir d'une seule image ou d'une séquence d'images. Elle implique le développement d'une base théorique et algorithmique pour parvenir à une compréhension visuelle automatique.

En tant que discipline scientifique, la vision par ordinateur concerne la théorie derrière les systèmes artificiels qui extraient des informations à partir d'images. Les données d'image peuvent prendre de nombreuses formes, telles que des séquences vidéo, des vues de plusieurs caméras ou des données multidimensionnelles d'un scanner médical.

En tant que discipline technologique, la vision par ordinateur cherche à appliquer ses théories et modèles pour la construction de systèmes de vision par ordinateur.[11]

2.5. Deep learning :

Les approches d'apprentissage profond basées sur les réseaux neuronaux ont montré un succès significatif dans un large éventail de tâches d'IA. L'un des facteurs critiques de leur succès est l'utilisation d'ensembles de données à grande échelle pour la formation. Cependant, lorsque les ensembles de données d'entraînement sont collectés auprès d'individus et contiennent des informations sensibles, les paramètres du modèle pourraient coder des informations privées et augmenter le risque d'une violation de données. La tendance croissante au partage et à la publication de modèles préétablis a encore accru les risques pour la vie privée des individus. Pour résoudre ce problème, nous proposons une nouvelle approche privée de la formation des réseaux neuronaux.

Plusieurs nouvelles stratégies sont incluses dans notre approche pour optimiser à la fois la perte de confidentialité et la précision du modèle. Nous avons utilisé une généralisation de la confidentialité différentielle appelée confidentialité différentielle concentrée (CDP), ainsi qu'une analyse formelle et raffinée de la perte de confidentialité sur deux méthodes différentes de traitement des données. Pendant la formation, nous avons utilisé un système de budgétisation dynamique de la confidentialité pour améliorer la précision du modèle. Une expérience approfondie montre que notre approche améliore efficacement la comptabilisation des pertes de confidentialité, l'efficacité de la formation et la qualité des modèles dans les limites d'un certain budget de confidentialité.[12]

2.6. Raspberry Pi et Camera :

Les Objets Intelligents jouent un rôle majeur dans la vision de l'Internet des Objets. Ces objets, équipés des technologies de l'information et de la communication, peuvent stocker leur contexte, sont mis en réseau, sont capables d'accéder à des services Internet et d'interagir entre eux et avec les humains.

La Raspberry Pi Fondation est une organisation basée au Royaume-Uni qui aide les jeunes, les chercheurs et les étudiants à apprendre l'informatique et la fabrication numérique. Le Raspberry Pi est un petit ordinateur de bureau qui pourrait être intégré dans plusieurs applications telles que : la robotique, les maisons intelligentes, le cœur de l'IA en réseau, le contrôleur d'usine et bien plus encore.

Dans ce projet, nous utilisons les versions 3 et 4 du Raspberry pi pour comparer les performances de notre application en temps réel.

Les chercheurs, dans [14], ont fait une comparaison des performances de référence entre différentes versions du Raspberry Pi.[13]

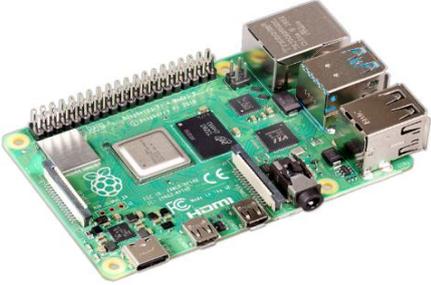
Naturellement, la dernière génération de Raspberry pi 4 affiche le matériel le plus performant. Dans notre cas, puisque nous avons les deux matériaux, nous avons décidé de les tester tous les deux dans le cas de la détection du masque facial.

Les principales différences entre le Raspberry Pi 3 modèle B et le Raspberry Pi 4 modèle B sont données dans le tableau 1.

Nous mettons en évidence les spécifications techniques liées au traitement

Puissance des matériaux : processeur et mémoire.

Tableau 1 : Raspberry Pi 3 VS Raspberry Pi 4.

Version	Détails techniques
<p>Raspberry Pi 3 Model B (2016)</p> 	<p>Processor :</p> <p>Broadcom BCM2837</p> <p>Quad Core ARM Cortex-A53</p> <p>1.2GHz 64bit CPU</p> <p>Memory :1GB</p>
<p>Raspberry Pi 4 Model B (2018)</p> 	<p>Processor :</p> <p>Broadcom BCM2711</p> <p>Quad-Core ARM Cortex-A72(ARM v8)</p> <p>SoC @ 1.5GHz 64bit CPU</p> <p>Memory :1GB and 8GB</p>

De la même fondation, il s'agit cette fois d'une caméra Pi dédiée à être utilisée avec la carte précédente. La caméra Raspberry Pi se branche directement sur le connecteur CSI du Raspberry Pi. Il est capable de fournir une image de résolution cristalline de 5 MP. Facile à installer et à utiliser. Il est entièrement compatible avec toutes les versions de Raspberry. Nous utilisons Raspberry Pi Camera Rev 1.3 (figure 14).

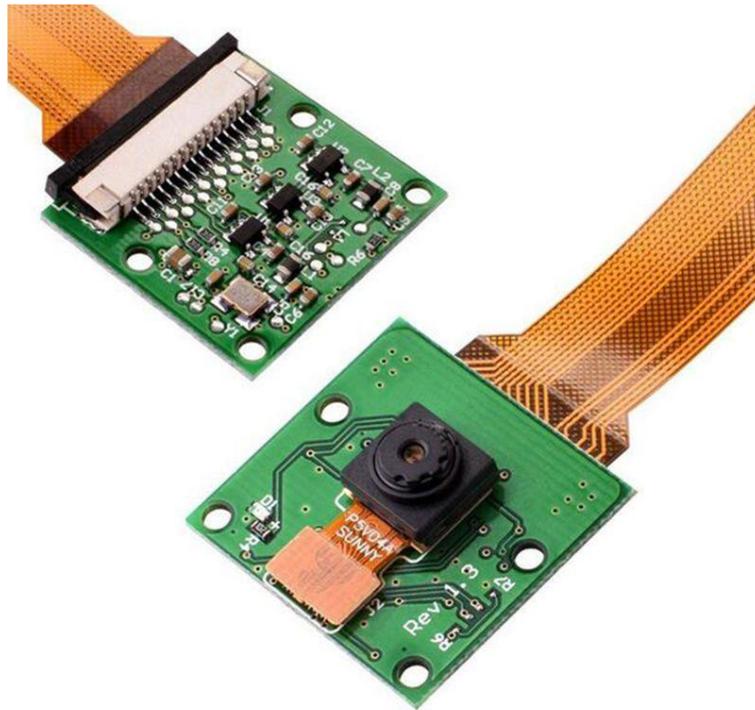


Figure 14 : La caméra Raspberry Pi Rev1.3

2.7. Python :

Python est un langage objet interprété de haut niveau, créé au début des années quatre-vingt-dix par Guido Van Rossum au Centrum voor Wiskunde à Informatica, Amsterdam. En 1995, Rossum poursuivit Le développement de Python à la Corporation for National Research Initiatives de Reston (Virginie). Et en 2000, Rossum créa l'équipe BeOpen PythonLabs qui, en octobre de la même année est incorporée à Zope Corporation puis à la société Digital Creations. En 2001, la PSF (Python Software Foundation) est créée.

Il s'agit d'une organisation à but non lucratif détenant les droits de propriété intellectuelle de Python.

Elle est sponsorisée en particulier par Zope Corporation. Python est distribué sous forme de logiciel libre.

Python est couvert par sa propre licence (voir le site [[www-PyLicence](http://www.PyLicence)]). Toutes les versions Depuis la 2.0.1 sont compatibles avec la licence GPL (GNU Public Licence1).

Le langage Python est aussi dit interprété car il est directement exécuté sans passer par une phase de compilation qui traduit le programme en langage machine, comme c'est le cas pour

le langage C. En quelque sorte, il fonctionne autant comme une calculatrice que comme un langage de programmation. Afin d'accélérer l'exécution d'un programme Python, il est néanmoins possible de traduire un programme dans un langage (bytecode) qui est ensuite interprété par une machine virtuelle Python. Ce mécanisme est semblable à celui propre au langage Java.

On considère que le langage Python est de haut niveau car il propose des fonctionnalités avancées et automatiques telle le garbage collecting.

Cette tâche correspond à la destruction automatique des objets créés lorsqu'ils ne sont plus utilisés.

Il propose également des structures de données complexes telles que des dictionnaires, éloignées des types numériques standards.

Le langage Python est modulaire.

La définition du langage est très succincte et autour de ce noyau concis, de nombreuses bibliothèques ou modules ont été développées.

Python est assez intuitif, être à l'aise avec ce langage revient à connaître tout autant sa syntaxe que les nombreux modules disponibles, eux-mêmes écrits en Python.

Le langage Python est à syntaxe positionnelle en ce sens que l'indentation fait partie du langage. Le point-virgule permet de séparer les instructions en langage C, l'accolade permet de commencer un bloc d'instruction.

En Python, seule l'indentation permet de marquer le début et la fin d'un tel bloc, ce procédé consiste à décaler les lignes vers la droite pour montrer qu'elles appartiennent au même bloc d'instructions.

2.8. Open CV :

Bibliothèque de vision par ordinateur open source. OpenCV a été développé pour une utilisation rapide dans les infrastructures de vision par ordinateur populaires et les produits de perception des machines grand public. En tant que logiciel sous licence BSD, OpenCV encourage les entreprises à utiliser et à modifier le code. La bibliothèque compte plus de 2 500 algorithmes intégrés, y compris des algorithmes traditionnels et avancés de vision par ordinateur et

d'apprentissage automatique. Ces algorithmes peuvent être utilisés pour détecter et reconnaître des images, identifier des objets, classer l'activité humaine dans des images, suivre le mouvement de la caméra, suivre des objets en mouvement, extraire des modèles 3D d'objets, créer des nuages de points 3D à partir de caméras stéréo, générer des images haute résolution de scènes entières. Image. OpenCV compte plus de 47 000 membres de groupes d'utilisateurs et on estime qu'il est téléchargé plus de 18 millions de fois. Les bibliothèques sont largement utilisées dans les entreprises, les sociétés de conseil et les agences gouvernementales. Outre les entreprises en place employées par des bibliothèques telles que Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota, etc., il existe également des startups utilisant OpenCV, telles que Applied Minds, Video Surf et Zeitera. Les utilisations d'OpenCV incluent les domaines suivants : collecter des vues de rue, détecter des intrusions dans la surveillance par caméra israélienne, surveiller des machines minières chinoises, permettre à des robots de manipuler et de collecter des objets dans les garages de Willow, détecter des noyades dans des piscines en Europe, diffuser de l'art interactif en Espagne et en Nouvelle-Zélande. York, surveillance des pistes de débris en Turquie, vérifiez les étiquettes professionnelles dans le domaine du trafic. Il prend en charge Linux, Windows, Mac OS et Android et possède des interfaces Java, Python, C++ et MATLAB. OpenCV utilise les instructions MMX et SSE lorsqu'elles sont disponibles, principalement pour les applications de vision en temps réel, et construit actuellement activement des interfaces CUDA et Open CV matures, avec plus de 500 algorithmes et environ 10 fois le nombre de fonctions algorithmiques. Open CV est natif de C++ et inclut un modèle d'interface avec des conteneurs STL qui fonctionnent de manière transparente.[12]

2.9. Visual Studio Code :

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS.

Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégrer. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.

Le code source de Visual Studio Code provient du projet logiciel libre et open source VSCode de Microsoft publié sous la licence MIT permissive, mais les binaires compilés constituent un freeware, c'est-à-dire un logiciel gratuit pour toute utilisation mais privé.

Dans le sondage auprès des développeurs réalisé par Stack Overflow en 2021, Visual Studio Code a été classé comme l'outil d'environnement de développement le plus populaire, avec 71,06 % des 82 277 répondants déclarant l'utiliser.[14]

2.10. Colab :

Colaboratory, souvent raccourci en "Colab", est un produit de Google Research. Colab permet à n'importe qui d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. C'est un environnement particulièrement adapté au machine Learning, à l'analyse de données et à l'éducation. En termes plus techniques, Colab est un service hébergé de notebooks Jupiter qui ne nécessite aucune configuration et permet d'accéder sans frais à des ressources informatiques, dont des GPU.[18]

2.11. Les avantages et les inconvénients des deux plateformes Colab et Visual Studio :

Tableau 2 : Les avantages et les inconvénients des deux plateformes Colab et Visual Studio

Colab		Visual studios	
Avantage	Inconvénient	Avantage	Inconvénient
Exécute des programmes complexes sans recours à une installation	La connexion s'interrompt avec le serveur après un moment d'inactivité	Gratuit sur l'utilisation	Requière l'installation de beaucoup d'extensions
Donne l'accès à un espace de donnée RAM et ROM CPU et GPU virtuellement	La confidentialité est très faible	Rapidité d'exécution	Bug au niveau de la programmation

Conclusion :

Dans ce chapitre nous avons vu les définitions essentielles de base et les pré requis pour comprendre notre projet et son fonctionnement et dans le prochain chapitre nous allons voir notre propre travail et le résultat finale.

Chapitre 3 :

Conception et validation

3. Introduction :

Dans ce chapitre nous allons voir la méthode que nous avons fait et la procédure du travail appliqué avec les algorithmes du face détections et utilisé dans Visual studio code , Googel Colab , le Raspberry pi 3 et le Raspberry pi 4 pour réaliser ce projet.

3.1. Méthodologie proposée :

La méthodologie du système proposé comprend deux étapes principales : la première étape est la création d'un modèle d'appariement de visages à l'aide de techniques d'apprentissage en profondeur et d'apprentissage automatique traditionnelles. Le principal défi consistait à créer un ensemble de données composé de visages avec et sans masque facial. Un détecteur de visage basé sur la vision par ordinateur a été créé à l'aide de l'ensemble de données créé, OpenCV et Python avec TensorFlow utilisant Visual studio code, ainsi que dans notre cadre d'apprentissage automatique personnalisé. Les techniques de vision par ordinateur et d'apprentissage en profondeur ont été utilisées pour déterminer si la personne porte un masque facial ou non. Cela contribue à accélérer la prolifération de la vision par ordinateur dans les domaines actuellement naissants tels que la signalisation numérique, la conduite autonome, la reconnaissance vidéo, le service client, la traduction linguistique et les applications mobiles.

L'élément principal de l'apprentissage en profondeur est les DNN, qui permet la segmentation de la reconnaissance d'objets. La méthodologie proposée utilise un classificateur CNN profond hybride pour segmenter les caractéristiques pertinentes du visage. Les DNN sont généralement utilisés dans les tâches liées à la vision par ordinateur car ils agissent comme un outil efficace pour augmenter la résolution d'un classifieur. Les modèles de reconnaissance faciale et de classification peuvent être formés à l'aide de CNN, d'extraction de caractéristiques avancées et de méthodes de classification pour identifier et classer les images faciales avec des caractéristiques minimales et stocker des détails fins. CNN est utilisé pour collecter des photos de personnes portant des masques faciaux, plutôt que des photos d'une base de données, puis

faire la distinction entre les images de personnes portant des masques faciaux et les photos d'autres personnes en fonction des expressions faciales, du contenu et des informations spatiales.

Le but principal de l'utilisation du Raspberry Pi est d'effectuer des tâches critiques, telles que le CPU, le GPU, l'entrée et la sortie. Caractéristiques de la carte Raspberry Pi Les broches GPIO sont essentielles à l'utilisation de la programmation matérielle pour permettre au Raspberry Pi de contrôler les circuits électroniques et les dispositifs de traitement de données sur les données d'entrée et de sortie. Il est possible d'installer et d'exécuter le Raspberry Pi sous le système d'exploitation Raspbian et de le programmer à l'aide du langage de programmation Python. Par conséquent, il peut être simple d'identifier une personne à la porte du bus ou au point d'entrée d'une gare grâce à un flux image/vidéo à l'aide de techniques de vision par ordinateur et d'apprentissage en profondeur.

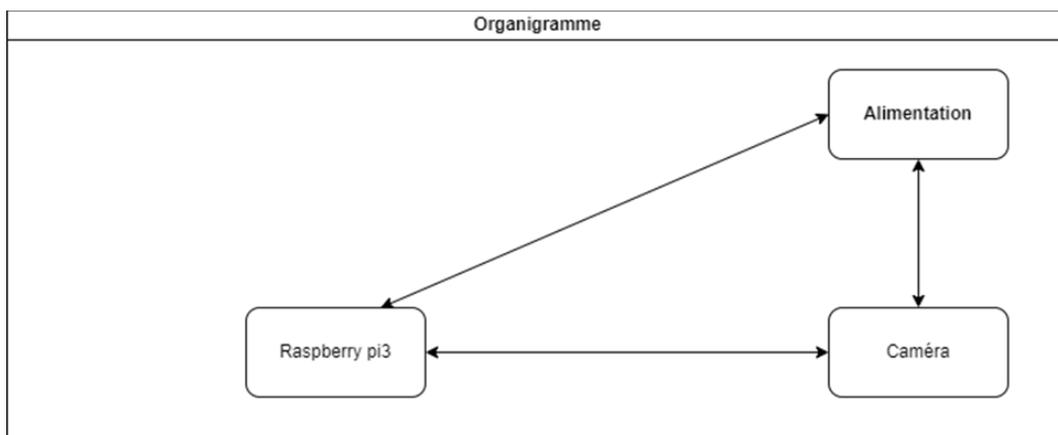


Figure 15 : Block Diagramme

Les sous-sections suivantes décrivent en détail le modèle de détection de masque facial et la technologie opérationnelle.

3.2. L'approche de « Face Mask Detection Algorithm » :

Lors du développement de cet algorithme de détection de masque, certaines technologies ont été utilisées pour identifier les visages

Des personnes et analyser leur visage, puis décider si elles portent un masque ou non, l'une de ces technologies est DNN elle est fondamentalement la détection d'objets en temps réel. Cet algorithme, qui est l'un des algorithmes de détection d'objets les plus efficaces qui englobe également plusieurs des idées les plus innovantes issues de la communauté de recherche en vision par ordinateur. La détection d'objets est une capacité essentielle de la technologie l'industrie. C'est un domaine de la vision par ordinateur qui explose et fonctionne tellement mieux qu'il y a à peine quelques années.

Cet algorithme rend la détection de masque possible en implémentant un DNN spécial capable de traiter les données extrêmement rapidement en fonction de fonctions de coût qui

Tentent de minimiser l'erreur autant que possible et d'obtenir la meilleure prédiction possible.

3.3. Système de détection de mask facial :

À l'aide d'une combinaison de techniques d'apprentissage en profondeur et de CNN, un système de détection de masque facial en temps réel a été développé. La méthode de segmentation d'image produit des résultats efficaces et précis pour la détection de visage. La lecture du visage a montré que plus de la moitié des participants pouvaient déterminer avec précision si l'individu portait un masque ou non.

L'algorithme 1 explique la méthodologie derrière le processus de détection du masque facial.

Algorithme 1 Algorithme de détection de masque facial

1 : **Entrée** : ensemble de données d'image avec et sans masque facial

2 : **Sortie** : Images classifiées avec étiquettes avec et sans masque

3 : **Pour** chaque image du jeu de données, **faites** :

4 : Créez deux catégories pour l'image.

5 : Étiquetez chaque catégorie en fonction du masque avec et sans.

- 6 : Convertissez l'image RVB en une image en niveaux de gris d'une taille de 100×100 pixels.
- 7 : **si** le visage est détecté **alors**
- 8 : Transformez contextuellement l'image et intégrez-la dans un tableau à quatre dimensions.
- 9 : Incorporez une couche de convolution avec 200 filtres au mélange.
- 10 : Incorporez la 2e couche de convolution de 100 filtres à l'image.
- 11 : Ajoutez une surface aplatie au classificateur CNN profond pour le rendre plus précis.
- 12 : Incorporer une couche dense de 64 neurones au modèle.
- 13 : Incorporer la dernière couche Dense
- 14 : **si** le masque est détecté **alors**
- 15 : Ajouter l'image à la catégorie Face avec masque
- 16 : **sinon**
- 17 : Ajouter l'image à la catégorie Visage sans masque
- 18 : **fin**
- 19 : **sinon** si le visage n'est pas détecté **alors**
- 20 : Revenir à l'image suivante dans l'ensemble de données
- 21 : **fin**
- 22 : **fin**

3.4. Mise en œuvre du système

3.4.1. Phase de Train :

3.4.1.1. `Train_Mask_Detector.py` :

1/d'abord nous importons les packages et les bibliothèques nécessaires à l'entraînement de notre algorithme de détection de masque.

3.4.1.2. Réglage fin avec Keras et TensorFlow :

Nous effectuons un réglage fin en prenant un neurone convolutif Modèle de réseau pré-entraîné et affiné pour qu'il fonctionne Classification d'images et reconnaître les classes sur lesquelles il n'a jamais été formé.

Le réglage fin a été utilisé dans plusieurs applications telles que la classification des fausses nouvelles [18], l'identification des maladies des plantes à l'aide d'images de feuilles [19], la classification des tumeurs cérébrales [20] et autres. Dans notre cas, nous devons recycler un réseau de neurones pour reconnaître les personnes portant un masque facial, ce qu'il n'a pas été initialement formé pour reconnaître.

Le réglage fin est l'une des méthodes d'apprentissage par transfert [21].

Tout d'abord, nous devons trouver l'ensemble final de couches entièrement connectées

À partir d'un réseau de neurones convolutionnelles pré-entraînés. On l'appelle généralement la "tête" du réseau et c'est là que les prédictions des étiquettes de classe sont renvoyées. Deuxièmement, nous remplaçons la tête par un nouvel ensemble de couches entièrement connectées en fonction de nos besoins. En particulier, toutes les couches sous la tête sont figées afin que leur poids ne puisse pas être modifié, même pendant la rétropropagation.

Troisièmement, nous formons le réseau en utilisant un petit taux d'apprentissage afin que le nouvel ensemble de couches entièrement connectées puisse apprendre des modèles à partir des couches convolutionnelles précédemment apprises plus tôt dans le réseau.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.layers import AveragePooling2D
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Input
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications.mobilenet_v2
import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from imutils import paths
import matplotlib.pyplot as plt
import numpy as np
import os
```

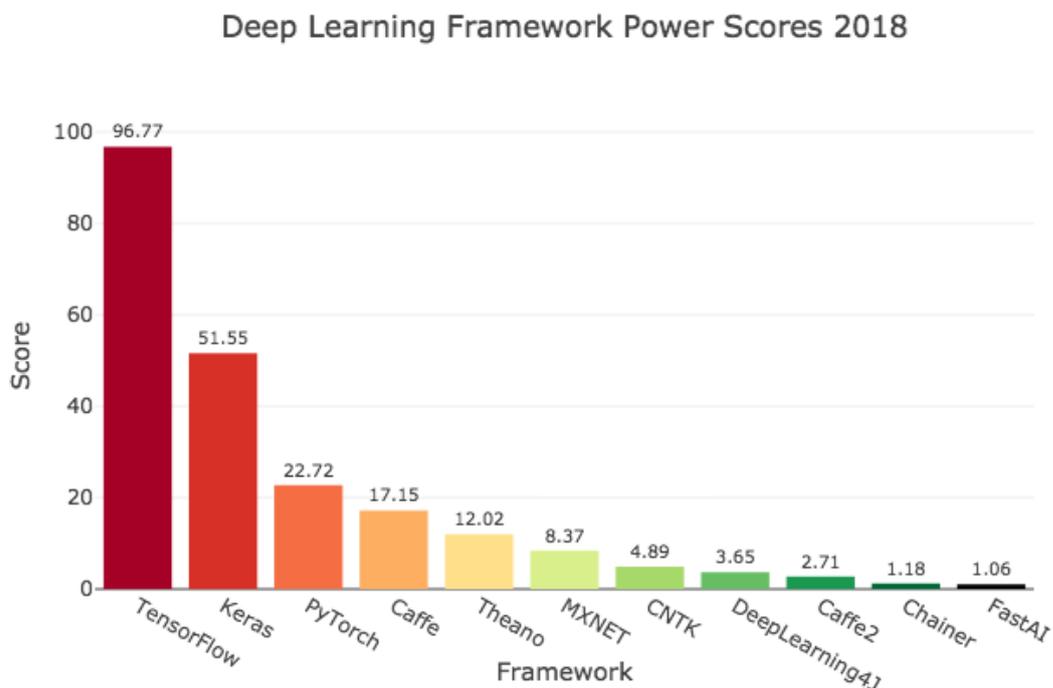


Figure 16 : Comparaison librairies apprentissage automatique.

2/ Visualisation des données : ensuite on introduit le chemin des modèles préformés dans le directory qu'il faut classer les entrées en deux classes pour faciliter l'entraînement a la détection des modèles de masque.

3.4.1.3. Base de données des masques :

Plusieurs ensembles de données d'images fixes à usage unique sont disponibles pour la détection de visage uniquement. Presque tous les ensembles de données peuvent être considérés comme faux en l'absence d'informations du monde réel, et la plupart de ceux qui existent souffrent de l'inclusion d'informations incorrectes et de bruit. Ensembles de données sur les masques médicaux de Kaggle et Witkowski ont été utilisés pour étendre les ensembles de données d'apprentissage du modèle. De plus, la collecte de données a été effectuée à l'aide de l'ensemble de données masqué, qui impliquait une application en aveugle. L'ensemble de données Kaggle contient de nombreuses personnes dont les visages sont floutés pour protéger leur vie privée, ainsi que des fichiers XML pertinents qui décrivent leurs dispositifs de protection de l'anonymat. L'ensemble de données contient un total de 678 photographies. Puis Recherchez l'extension de l'ensemble de données PyImage dans les paramètres de masquage naturel et renvoyez-le en tant que "Prajury B".

La base de données utiliser se compose de 3846 photos qui sont divisées en deux groupes :

- Celles qui ont des masques (with_mask) 1916 images.
- Celles qui n'ont pas de masques (\without_mask) 1930 images.





Figure 17 : Base de données des masques

```
datasetP=r'C:\Users\Abd El Kayoum\Desktop\Face-Mask-Detection-  
Raspberry-pi-2021-main\dataset'  
plotP=r'C:\Users\Abd El Kayoum\Desktop\Face-Mask-Detection-  
Raspberry-pi-2021-main'  
modelP=r'C:\Users\Abd El Kayoum\Desktop\Face-Mask-Detection-  
Raspberry-pi-2021-main'  
cwd=os.getcwd()
```

3/Fractionner les données : L'ensemble de données est divisé en deux ensembles : 80 % représentent l'ensemble de données d'apprentissage pour la couche de convolution et 20 % représentent l'ensemble de données de validation pour la méthode proposée.

```
print("[INFO] loading images...")  
imagePaths = list(paths.list_images(datasetP))  
data = []  
labels = []  
for imagePath in imagePaths:  
  
    label = imagePath.split(os.path.sep)[-2]  
  
    image = load_img(imagePath, target_size=(224, 224))  
    image = img_to_array(image)  
    image = preprocess_input(image)  
  
    data.append(image)  
    labels.append(label).  
data = np.array(data, dtype="float32")  
labels = np.array(labels)
```

```
lb = LabelBinarizer()
labels = lb.fit_transform(labels)
labels = to_categorical(labels)

(trainX, testX, trainY, testY) = train_test_split(data, labels,
                                                  test_size=0.20, stratify=labels, random_state=42)
```

Augmentation des données : Pour l'augmentation des données, d'autres images ont été ajoutées à l'ensemble de données et chacune des images a été tournée au fur et à mesure que nous poursuivons cette étape. En utilisant la méthode d'augmentation des données,

3.4.1.4. Data augmentation :

L'augmentation des données est une technique utilisée pour créer plus d'exemples, artificiellement, à partir d'un ensemble de données existant. Ceci est utile si votre ensemble de données est petit et que vous souhaitez augmenter le nombre d'exemples. L'augmentation des données peut souvent résoudre le surajustement afin que votre modèle se généralise bien après l'entraînement. Pour les images, une variété d'augmentations peut être appliquée pour augmenter le nombre d'exemples.[15].





Figure 18: Exemple de data augmentation

```
aug = ImageDataGenerator(  
    rotation_range=20,  
    zoom_range=0.15,  
    width_shift_range=0.2,  
    height_shift_range=0.2,  
    shear_range=0.15,  
    horizontal_flip=True,  
    fill_mode="nearest")
```

5/Modélisation

Après la configuration ci-dessus, nous pouvons maintenant commencer à créer le CNN. La première chose est la couche d'entrée du CNN qui est chargée de lire les données de l'utilisateur et de les transmettre aux couches cachées pour traitement on charge le réseau MobileNetV2.

3.4.1.5 MobileNet2 :

Est une architecture de réseau neuronal convolutif qui cherche à bien fonctionner sur les appareils mobiles. Il est basé sur une structure résiduelle inversée où les connexions résiduelles se situent entre les couches de goulot d'étranglement. La couche d'expansion intermédiaire utilise des convolutions légères en profondeur pour filtrer les caractéristiques comme source de non-linéarité. Dans son ensemble, l'architecture de MobileNetV2 contient la couche initiale entièrement convolutive avec 32 filtres, suivie de 19 couches résiduelles de goulot d'étranglement.

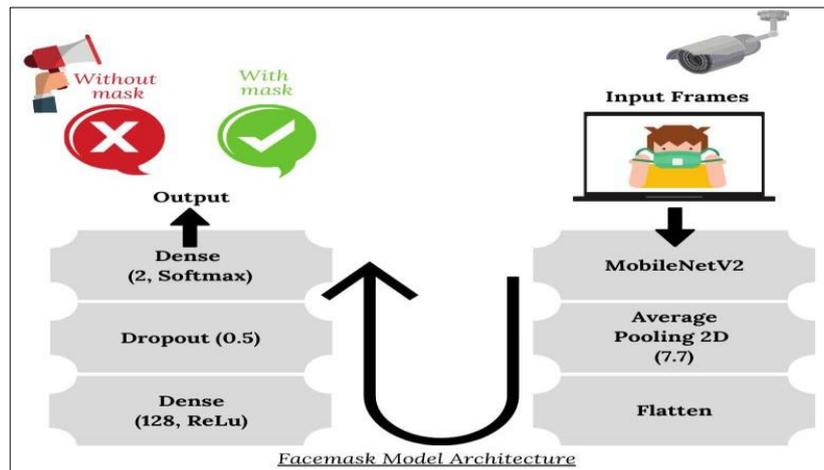


Figure 19 : Détection de masque facial architecture de modèle basé sur CNN

Dans notre cas, nous chargeons l'architecture MobileNetV2 [22], laissant de côté les couches entièrement connectées. Ensuite, nous ajoutons cinq autres couches : Average pooling, flatten, Dense with ReLu, Dropout et Dense using Softmax for final result. Puisqu'il s'agit d'une classification binaire, la sortie finale donne soit le visage est « avec masque » ou « sans masque ». L'architecture du modèle de base CNN est détaillée dans les figures 20 et 21.

```
baseModel = MobileNetV2(weights="imagenet", include_top=False,
    input_tensor=Input(shape=(224, 224, 3)))

headModel = baseModel.output
headModel = AveragePooling2D(pool_size=(7, 7))(headModel)
headModel = Flatten(name="flatten")(headModel)
headModel = Dense(128, activation="relu")(headModel)
headModel = Dropout(0.5)(headModel)
headModel = Dense(2, activation="softmax")(headModel)

model = Model(inputs=baseModel.input, outputs=headModel)

layer.trainable = False
```

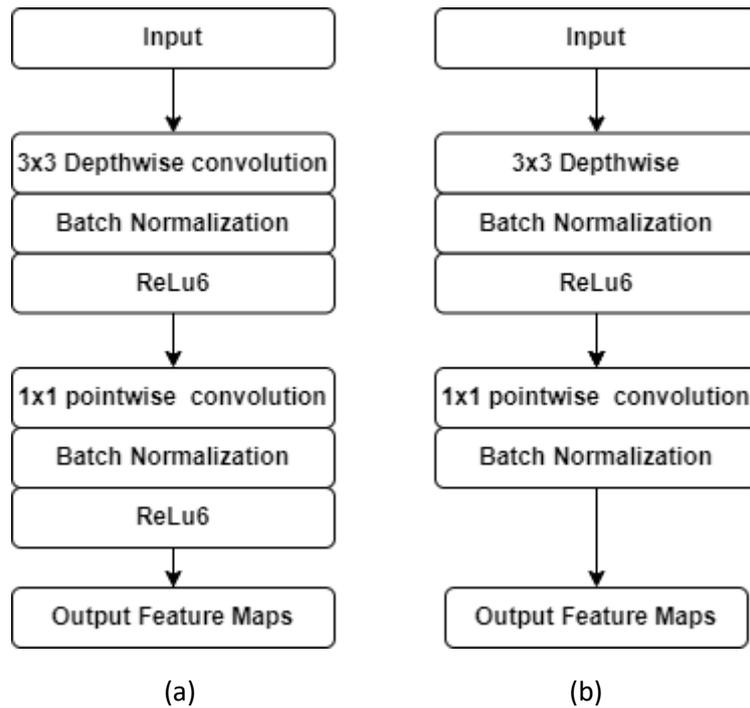


Figure 20 : (a) la structure de base de MobileNet : (b) unité dans le MobileNet.

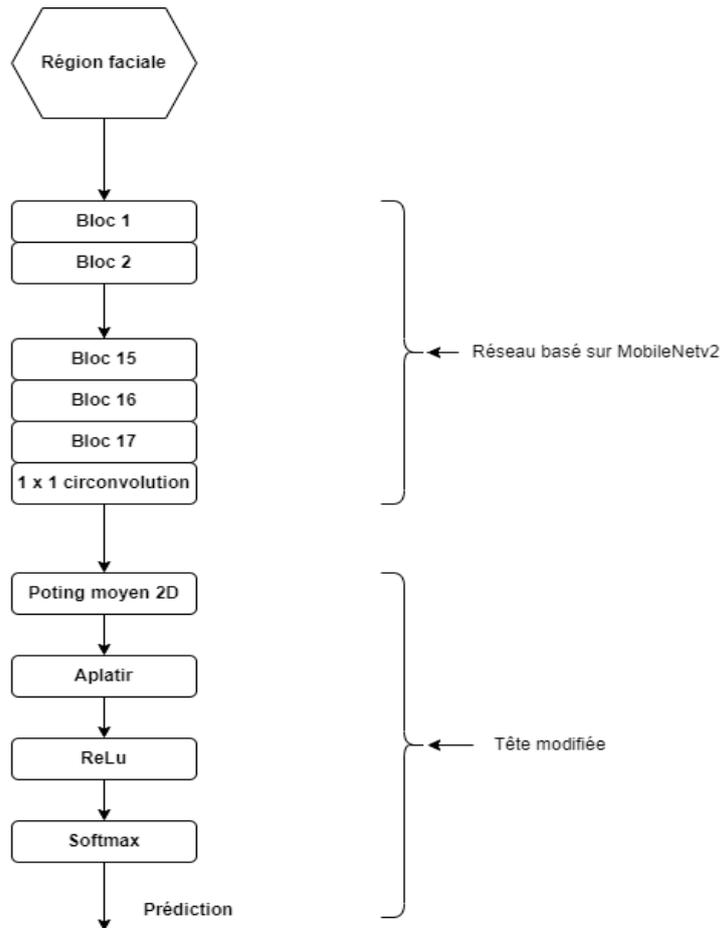


Figure 21 : Classificateur de masque facial construit à l'aide de l'architecture MobileNetV2.

average_pooling2d (AveragePooli (None, 1, 1, 1280)	0	out_relu[0][0]	
flatten (Flatten)	(None, 1280)	0	average_pooling2d[0][0]
dense (Dense)	(None, 128)	163968	flatten[0][0]
dropout (Dropout)	(None, 128)	0	dense[0][0]
dense_1 (Dense)	(None, 2)	258	dropout[0][0]
=====			
Total params: 2,422,210			
Trainable params: 2,388,098			
Non-trainable params: 34,112			

Figure 22 : Extrait de l'architecture globale

6/ Validation du modèle

À ce stade, la couche d'entrée du CNN est terminée, nous pouvons maintenant commencer le training, d'abord la construction la tête du réseau pour faire la des prédictions sur l'ensemble des test pour chaque image de l'ensemble de test pour que nous devons trouver l'indice de l'étiquette avec la plus grande probabilité prédite correspondante et afficher un rapport de classification bien formaté sérialiser le modèle sur disque au format h5 du fichier, puis la constructions de la base de données qui est devisées en deux classes avec masque et sans masque et à la fin le model travaille en fusionnent ces deux étapes.

```
INIT_LR = 1e-4
EPOCHS = 20
BS = 32

print("[INFO] compiling model...")
opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
model.compile(loss="binary_crossentropy", optimizer=opt,
              metrics=["accuracy"])

print("[INFO] training head...")
H = model.fit(
    aug.flow(trainX, trainY, batch_size=BS),
    steps_per_epoch=len(trainX) // BS,
    validation_data=(testX, testY),
    validation_steps=len(testX) // BS,
    epochs=EPOCHS)
```

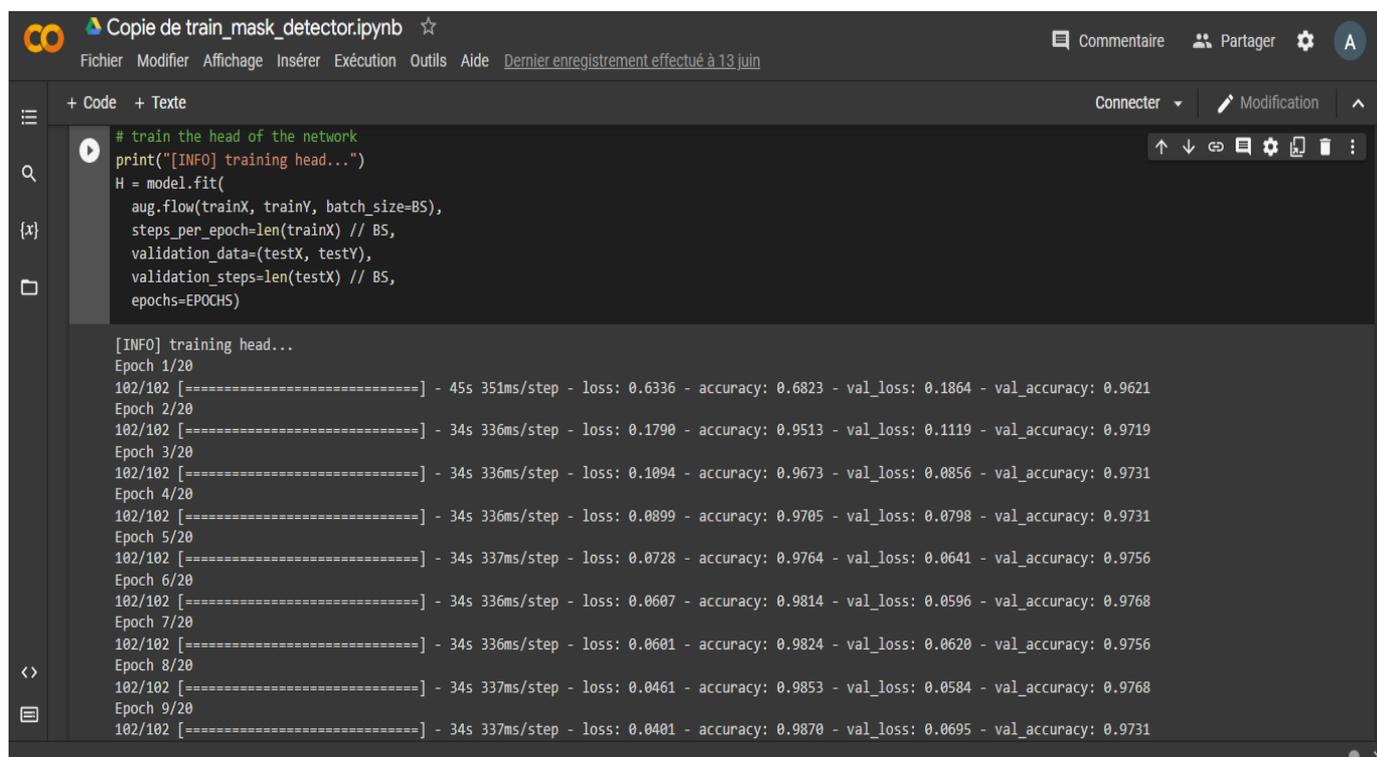
```
print("[INFO] evaluating network...")
predIdxs = model.predict(testX, batch_size=BS)

predIdxs = np.argmax(predIdxs, axis=1)

print(classification_report(testY.argmax(axis=1), predIdxs,
                            target_names=lb.classes_))

print("[INFO] saving mask detector model...")
model.save('model', save_format="h5")
N = EPOCHS
plt.style.use("ggplot")
plt.figure()
plt.plot(np.arange(0, N), H.history["loss"], label="train_loss")
plt.plot(np.arange(0, N), H.history["val_loss"], label="val_loss")
```

```
plt.plot(np.arange(0, N), H.history["acc"], label="train_acc")
plt.plot(np.arange(0, N), H.history["val_acc"], label="val_acc")
plt.title("Training Loss and Accuracy")
plt.xlabel("Epoch #")
plt.ylabel("Loss/Accuracy")
plt.legend(loc="lower left")
plt.savefig(plotP)
plt.show()
```

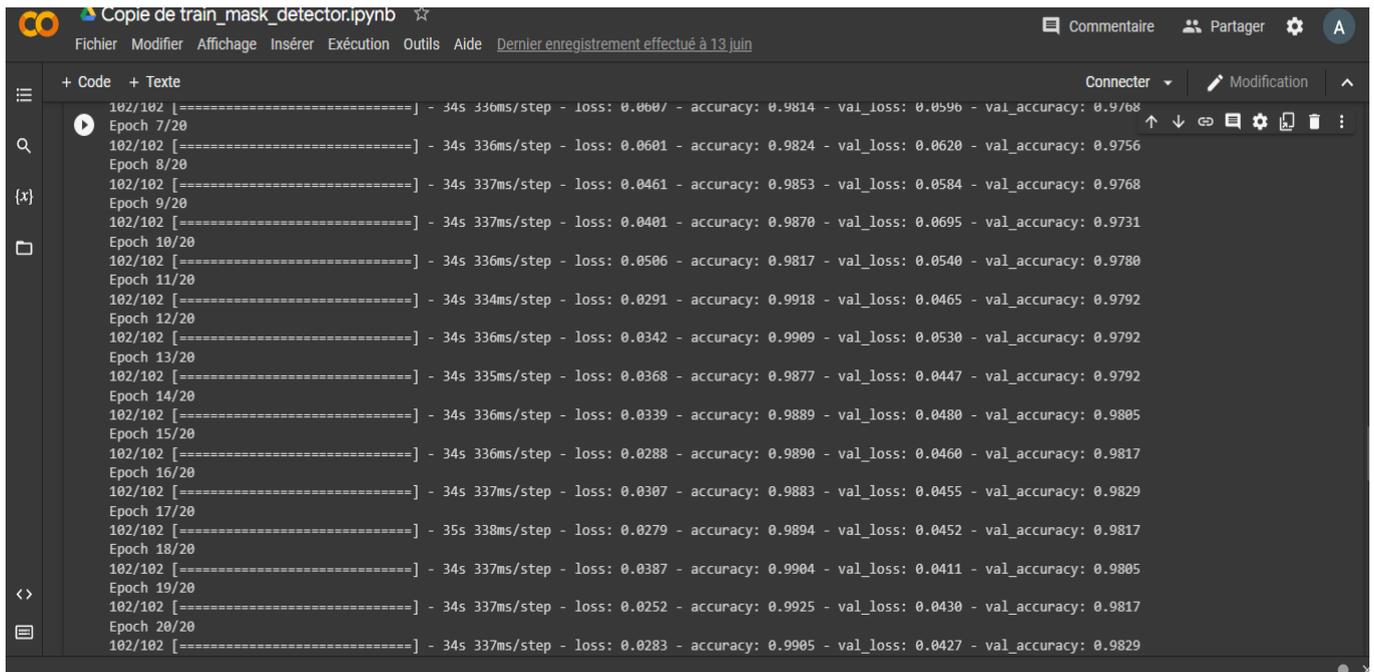


The screenshot shows a Jupyter Notebook titled "Copie de train_mask_detector.ipynb". The code cell contains the following Python code:

```
# train the head of the network
print("[INFO] training head...")
H = model.fit(
    aug.flow(trainX, trainY, batch_size=BS,
             steps_per_epoch=len(trainX) // BS,
             validation_data=(testX, testY),
             validation_steps=len(testX) // BS,
             epochs=EPOCHS)
```

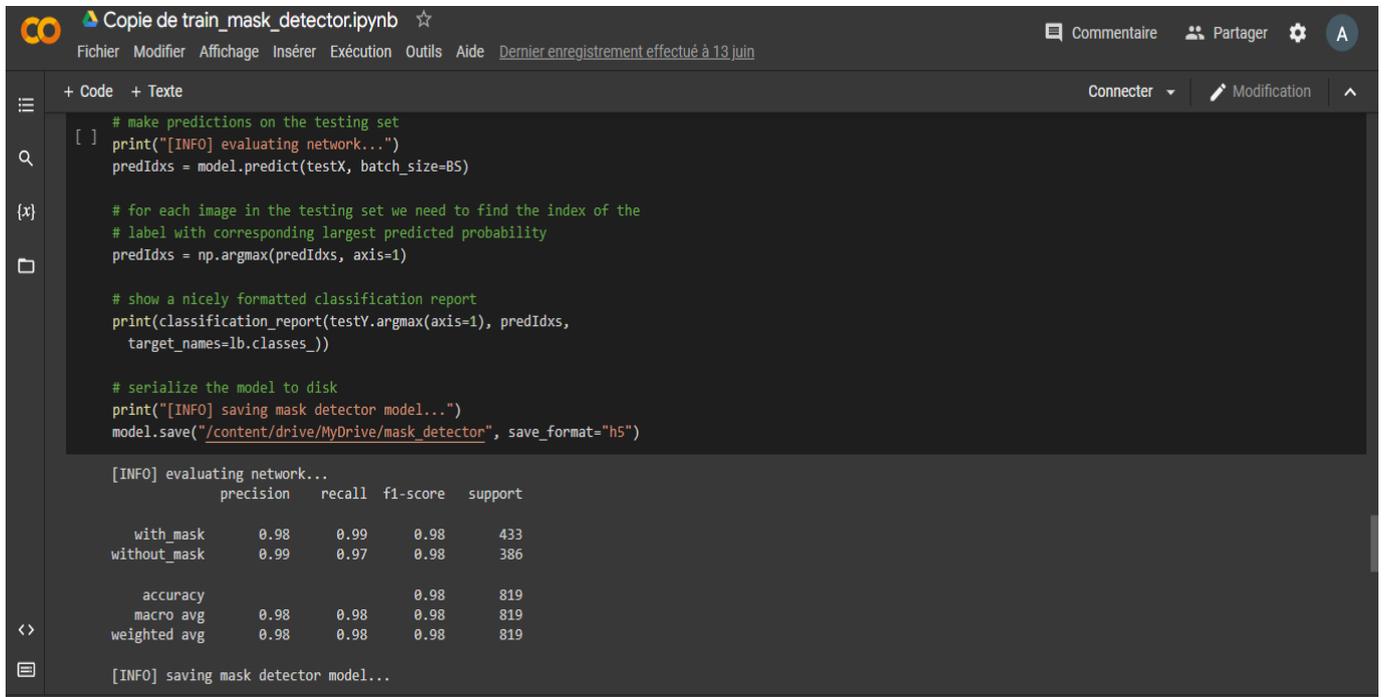
The output of the code cell shows the training progress over 9 epochs:

```
[INFO] training head...
Epoch 1/20
102/102 [=====] - 45s 351ms/step - loss: 0.6336 - accuracy: 0.6823 - val_loss: 0.1864 - val_accuracy: 0.9621
Epoch 2/20
102/102 [=====] - 34s 336ms/step - loss: 0.1790 - accuracy: 0.9513 - val_loss: 0.1119 - val_accuracy: 0.9719
Epoch 3/20
102/102 [=====] - 34s 336ms/step - loss: 0.1094 - accuracy: 0.9673 - val_loss: 0.0856 - val_accuracy: 0.9731
Epoch 4/20
102/102 [=====] - 34s 336ms/step - loss: 0.0899 - accuracy: 0.9705 - val_loss: 0.0798 - val_accuracy: 0.9731
Epoch 5/20
102/102 [=====] - 34s 337ms/step - loss: 0.0728 - accuracy: 0.9764 - val_loss: 0.0641 - val_accuracy: 0.9756
Epoch 6/20
102/102 [=====] - 34s 336ms/step - loss: 0.0607 - accuracy: 0.9814 - val_loss: 0.0596 - val_accuracy: 0.9768
Epoch 7/20
102/102 [=====] - 34s 336ms/step - loss: 0.0601 - accuracy: 0.9824 - val_loss: 0.0620 - val_accuracy: 0.9756
Epoch 8/20
102/102 [=====] - 34s 337ms/step - loss: 0.0461 - accuracy: 0.9853 - val_loss: 0.0584 - val_accuracy: 0.9768
Epoch 9/20
102/102 [=====] - 34s 337ms/step - loss: 0.0401 - accuracy: 0.9870 - val_loss: 0.0695 - val_accuracy: 0.9731
```



```
102/102 [=====] - 34s 336ms/step - loss: 0.0607 - accuracy: 0.9814 - val_loss: 0.0596 - val_accuracy: 0.9768
Epoch 7/20
102/102 [=====] - 34s 336ms/step - loss: 0.0601 - accuracy: 0.9824 - val_loss: 0.0620 - val_accuracy: 0.9756
Epoch 8/20
102/102 [=====] - 34s 337ms/step - loss: 0.0461 - accuracy: 0.9853 - val_loss: 0.0584 - val_accuracy: 0.9768
Epoch 9/20
102/102 [=====] - 34s 337ms/step - loss: 0.0401 - accuracy: 0.9870 - val_loss: 0.0695 - val_accuracy: 0.9731
Epoch 10/20
102/102 [=====] - 34s 336ms/step - loss: 0.0506 - accuracy: 0.9817 - val_loss: 0.0540 - val_accuracy: 0.9780
Epoch 11/20
102/102 [=====] - 34s 334ms/step - loss: 0.0291 - accuracy: 0.9918 - val_loss: 0.0465 - val_accuracy: 0.9792
Epoch 12/20
102/102 [=====] - 34s 336ms/step - loss: 0.0342 - accuracy: 0.9909 - val_loss: 0.0530 - val_accuracy: 0.9792
Epoch 13/20
102/102 [=====] - 34s 335ms/step - loss: 0.0368 - accuracy: 0.9877 - val_loss: 0.0447 - val_accuracy: 0.9792
Epoch 14/20
102/102 [=====] - 34s 336ms/step - loss: 0.0339 - accuracy: 0.9889 - val_loss: 0.0480 - val_accuracy: 0.9805
Epoch 15/20
102/102 [=====] - 34s 336ms/step - loss: 0.0288 - accuracy: 0.9890 - val_loss: 0.0460 - val_accuracy: 0.9817
Epoch 16/20
102/102 [=====] - 34s 337ms/step - loss: 0.0307 - accuracy: 0.9883 - val_loss: 0.0455 - val_accuracy: 0.9829
Epoch 17/20
102/102 [=====] - 35s 338ms/step - loss: 0.0279 - accuracy: 0.9894 - val_loss: 0.0452 - val_accuracy: 0.9817
Epoch 18/20
102/102 [=====] - 34s 337ms/step - loss: 0.0387 - accuracy: 0.9904 - val_loss: 0.0411 - val_accuracy: 0.9805
Epoch 19/20
102/102 [=====] - 34s 337ms/step - loss: 0.0252 - accuracy: 0.9925 - val_loss: 0.0430 - val_accuracy: 0.9817
Epoch 20/20
102/102 [=====] - 34s 337ms/step - loss: 0.0283 - accuracy: 0.9905 - val_loss: 0.0427 - val_accuracy: 0.9829
```

L'efficacité des données du modèle proposé est déterminée en comparant la perte et la précision des cycles d'entraînement et de validation. Il est essentiel de surveiller l'exactitude et la perte de précision du modèle pendant les phases d'entraînement et de test, qui durent au total 20 époques. En dehors de cela, comme le montre graphiquement la figure 23, la précision du modèle s'améliore avec le temps et devient stable après époque = 2.



```
[ ] # make predictions on the testing set
print("[INFO] evaluating network...")
predIdxs = model.predict(testX, batch_size=BS)

# for each image in the testing set we need to find the index of the
# label with corresponding largest predicted probability
predIdxs = np.argmax(predIdxs, axis=1)

# show a nicely formatted classification report
print(classification_report(testY.argmax(axis=1), predIdxs,
    target_names=lb.classes_))

# serialize the model to disk
print("[INFO] saving mask detector model...")
model.save("/content/drive/MyDrive/mask_detector", save_format="h5")

[INFO] evaluating network...
          precision    recall  f1-score   support

 with_mask      0.98      0.99      0.98         433
without_mask    0.99      0.97      0.98         386

 accuracy
macro avg      0.98      0.98      0.98         819
weighted avg   0.98      0.98      0.98         819

[INFO] saving mask detector model...
```

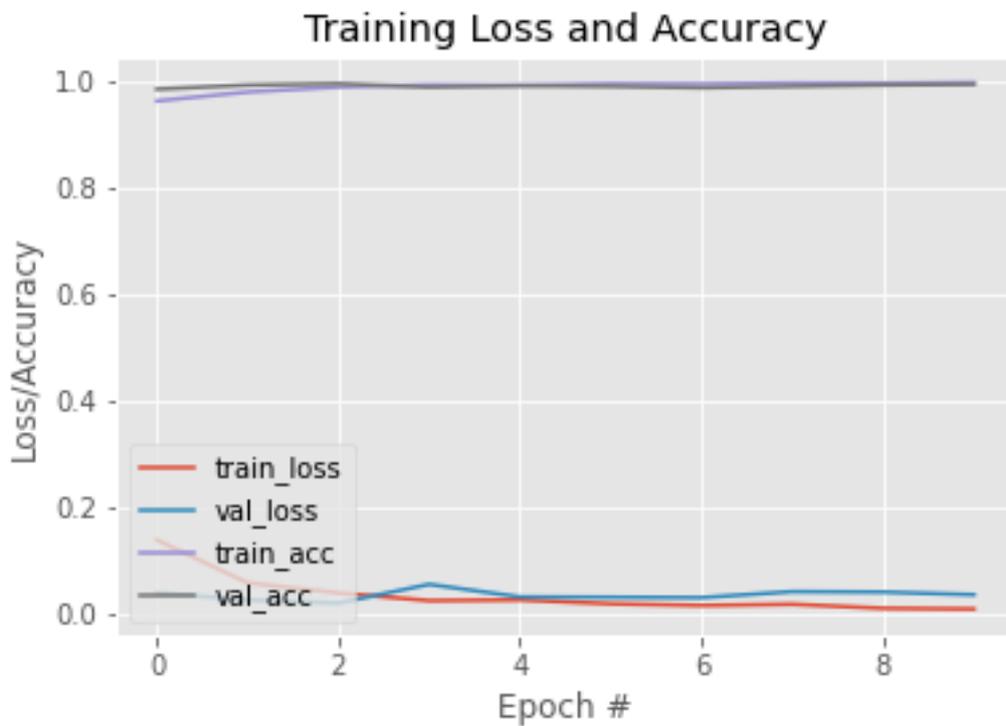


Figure 23 : Graph de training loss et accuracy

3.6 Phase Test :

3.6.1. Detect_Mask_Video.py :

Importation du programme de détection de visage. Ici, le camera est utilisée pour voir si un masque facial est porté ou non. Initialement, un programme de détection de visage a été mis en œuvre et les classificateurs de séquence basés sur les caractéristiques de Haar sont utilisés pour détecter les caractéristiques faciales

```
# USAGE
# python detect_mask_video.py

# import the necessary packages
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
from imutils.video import VideoStream
import numpy as np
import argparse
```

```
import imutils
from imutils.video import FPS
import time
import cv2
import os

def detect_and_predict_mask(frame, faceNet, maskNet):
    # grab the dimensions of the frame and then construct a blob
    # from it
    (h, w) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(frame, 1.0, (300, 300),
        (104.0, 177.0, 123.0))

    # pass the blob through the network and obtain the face detections
    faceNet.setInput(blob)
    detections = faceNet.forward()

    # initialize our list of faces, their corresponding locations,
    # and the list of predictions from our face mask network
    faces = []
    locs = []
    preds = []

    # loop over the detections
    for i in range(0, detections.shape[2]):
        # extract the confidence (i.e., probability) associated with
        # the detection
        confidence = detections[0, 0, i, 2]

        # filter out weak detections by ensuring the confidence is
        # greater than the minimum confidence
        if confidence > 0.6 :
            # compute the (x, y)-coordinates of the bounding box for
            # the object
            box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
            (startX, startY, endX, endY) = box.astype("int")

            # ensure the bounding boxes fall within the dimensions of
            # the frame
            (startX, startY) = (max(0, startX), max(0, startY))
            (endX, endY) = (min(w - 1, endX), min(h - 1, endY))

            # extract the face ROI, convert it from BGR to RGB channel
            # ordering, resize it to 224x224, and preprocess it
            face = frame[startY:endY, startX:endX]
            face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)
            face = cv2.resize(face, (224, 224))
```

```
face = img_to_array(face)
face = preprocess_input(face)

# add the face and bounding boxes to their respective
# lists
faces.append(face)
locs.append((startX, startY, endX, endY))

# only make a predictions if at least one face was detected
if len(faces) > 0:
    # for faster inference we'll make batch predictions on *all*
    # faces at the same time rather than one-by-one predictions
    # in the above `for` loop
    faces = np.array(faces, dtype="float32")
    preds = maskNet.predict(faces, batch_size=32)

# return a 2-tuple of the face locations and their corresponding
# locations
return (locs, preds)

# load our serialized face detector model from disk
print("[INFO] loading face detector model...")
prototxtPath = r"C:\Users\Abd El Kayoum\Desktop\Face-Mask-Detection-Raspberry-pi-2021-main\face_detector\deploy.prototxt"
weightsPath = r"C:\Users\Abd El Kayoum\Desktop\Face-Mask-Detection-Raspberry-pi-2021-main\face_detector\res10_300x300_ssd_iter_140000.caffemodel"
faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)
maskDP=r'C:\Users\Abd El Kayoum\Desktop\Face-Mask-Detection-Raspberry-pi-2021-main\model'
# load the face mask detector model from disk
print("[INFO] loading face mask detector model...")
maskNet = load_model(maskDP)

# initialize the video stream and allow the camera sensor to warm up
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()
fps=FPS().start()

time.sleep(2.0)

# loop over the frames from the video stream
while True:
    # grab the frame from the threaded video stream and resize it
    # to have a maximum width of 400 pixels
    frame = cv2.flip(vs.read(),1)
```

```
fps.update()

# detect faces in the frame and determine if they are wearing a
# face mask or not
(locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet)

# loop over the detected face locations and their corresponding
# locations
for (box, pred) in zip(locs, preds):
    # unpack the bounding box and predictions
    (startX, startY, endX, endY) = box
    (mask, withoutMask) = pred

    # determine the class label and color we'll use to draw
    # the bounding box and text

    if mask > withoutMask:
        label = "Mask"
    else:
        label="No Mask"
        warning="warning hazard threat detected"
        cv2.putText(frame, warning, (startX, endY+10),
            cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0,0,255), 2)

    color = (0, 255, 0) if label == "Mask" else (0, 0, 255)
    # include the probability in the label
    label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)

    # display the label and bounding box rectangle on the output
    # frame
    cv2.putText(frame, label, (startX, startY - 10),
        cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
    cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)

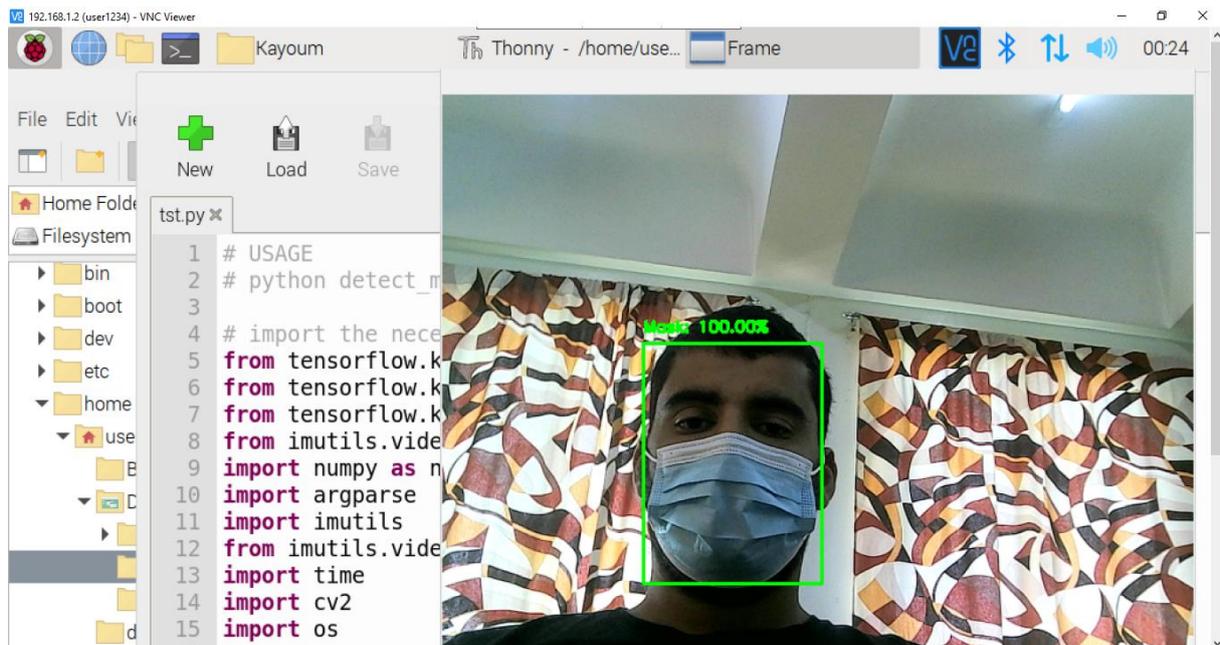
# show the output frame

cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF

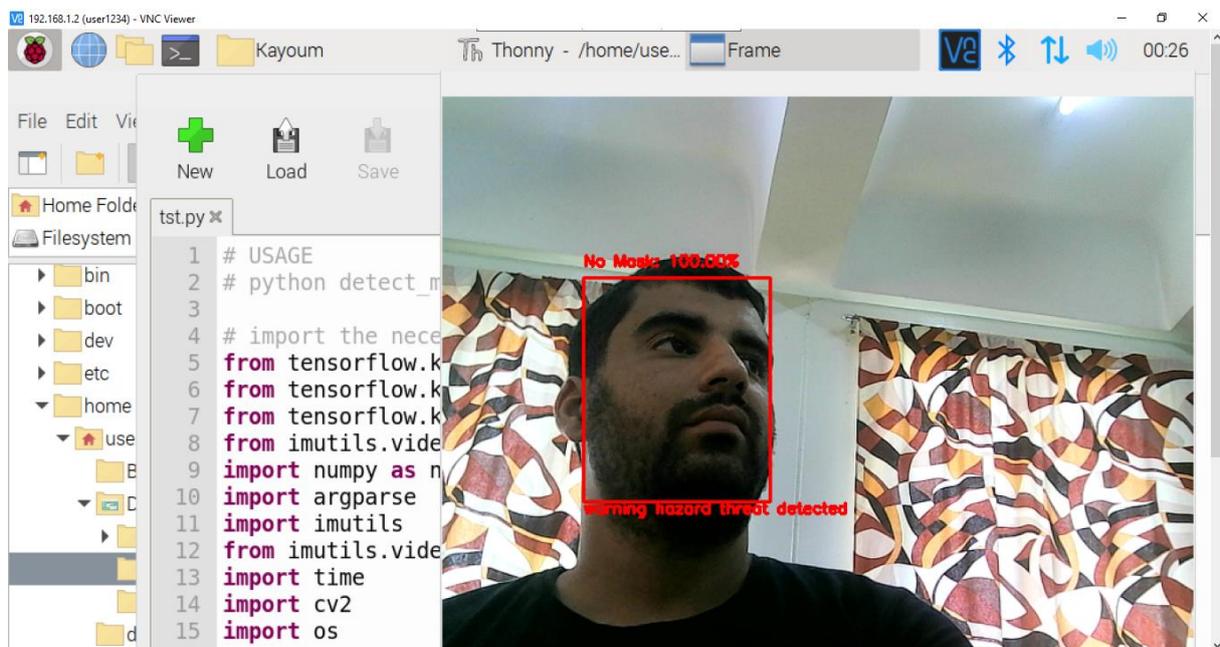
# if the `q` key was pressed, break from the loop
if key == ord("q"):
    fps.stop()
    break
```

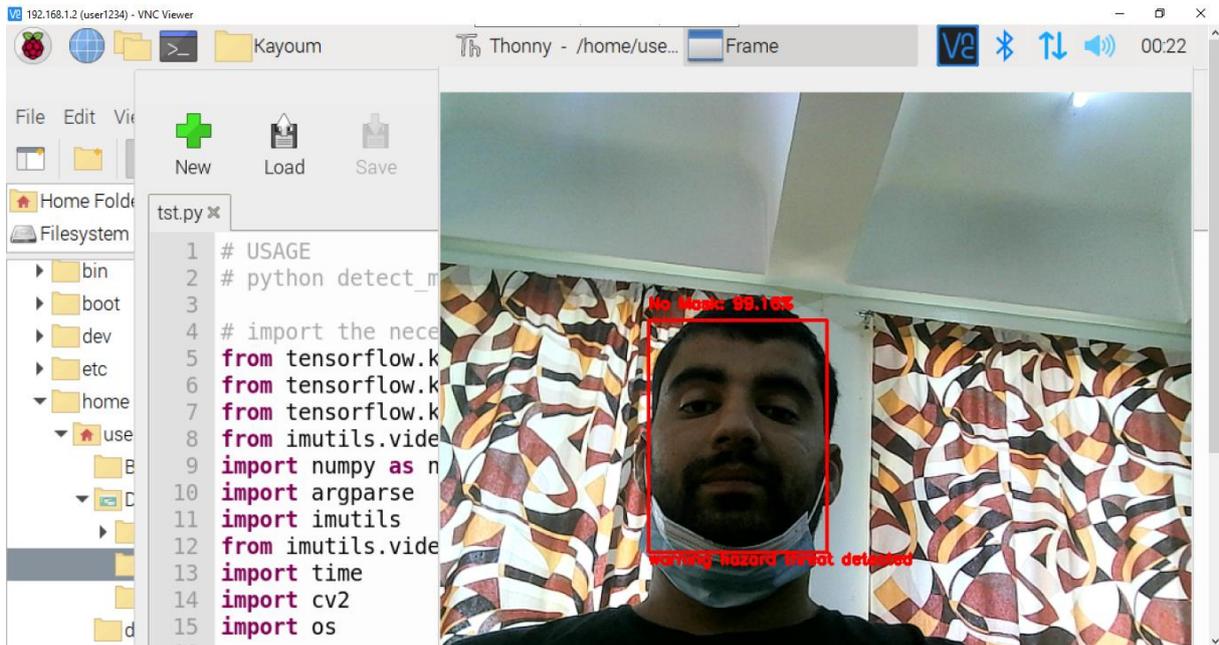
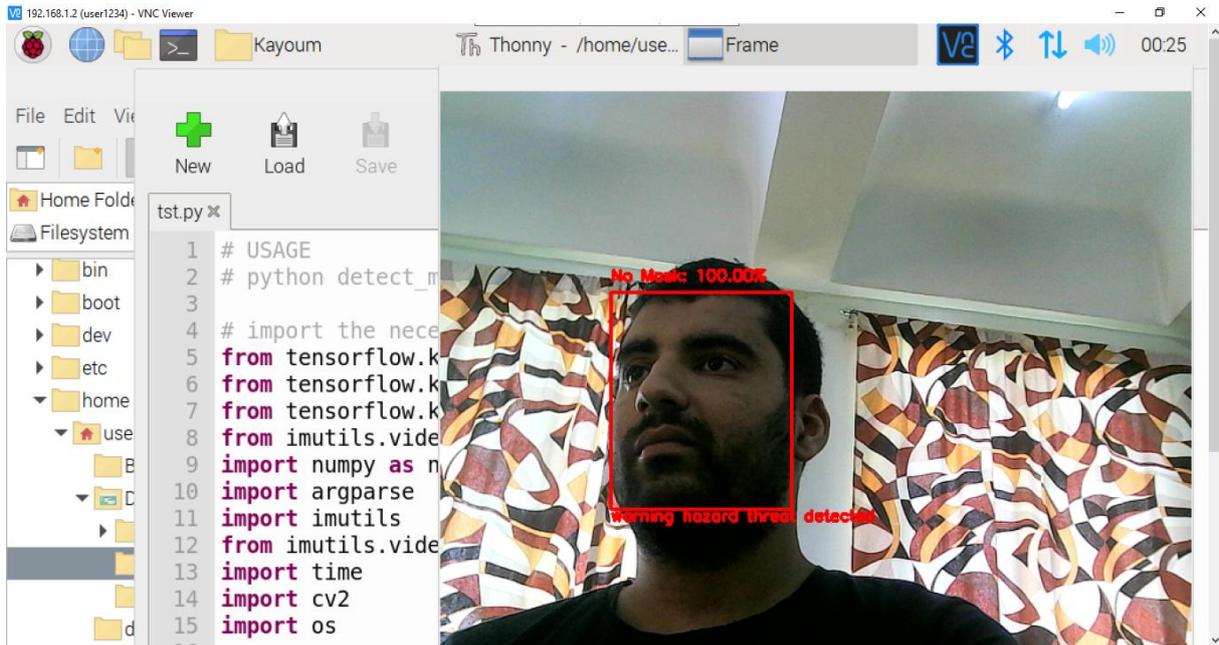
```
# do a bit of cleanup
print("fps is {:2f}".format(fps.fps()))
cv2.destroyAllWindows()
vs.stop()
```

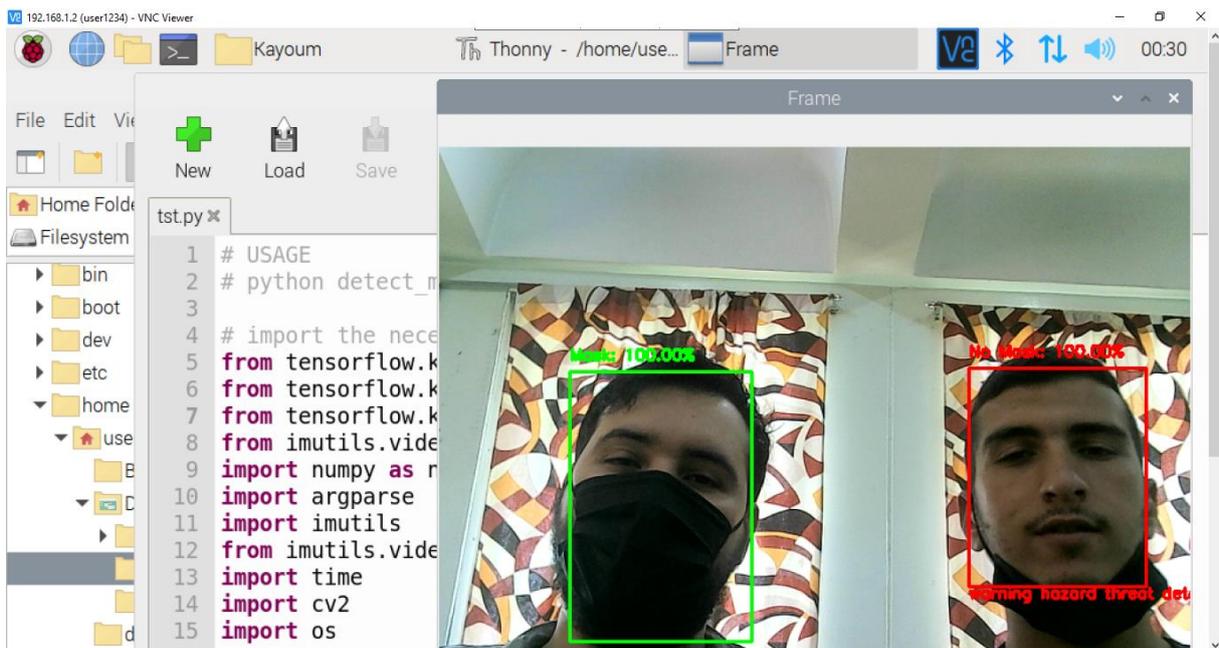
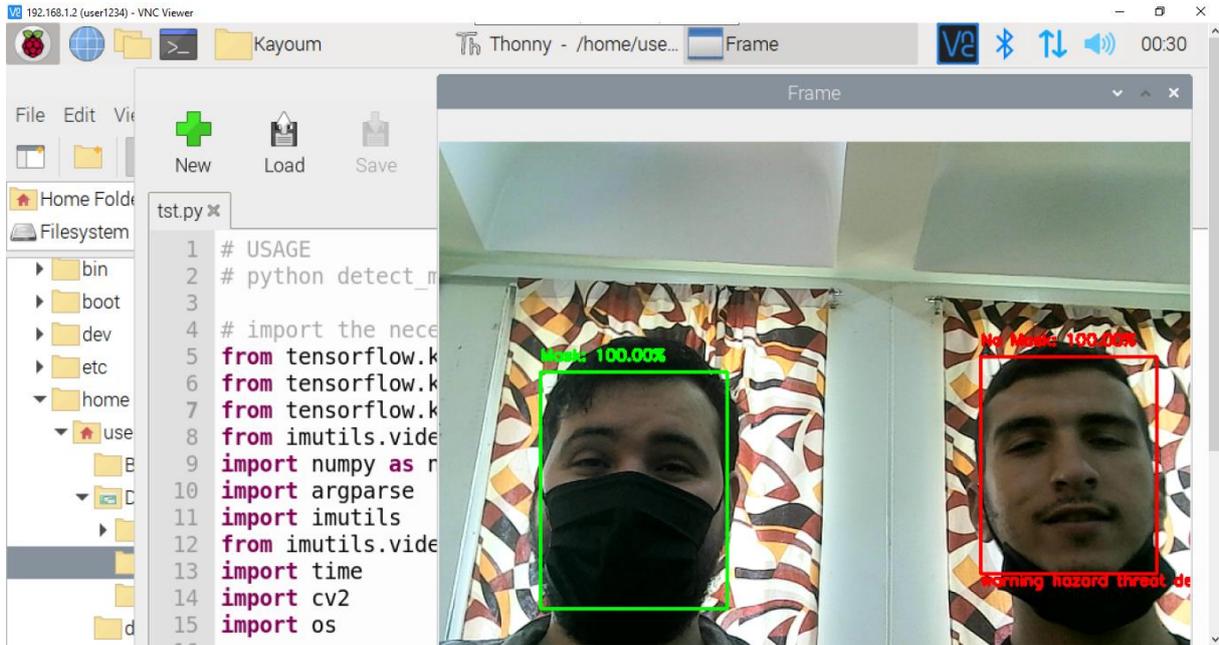
Après avoir exécuté toutes les étapes nécessaires, le résultat ressemblera à ceci : Si le visage est reconnu comme portant un masque facial, le résultat ressemblera à ceci :



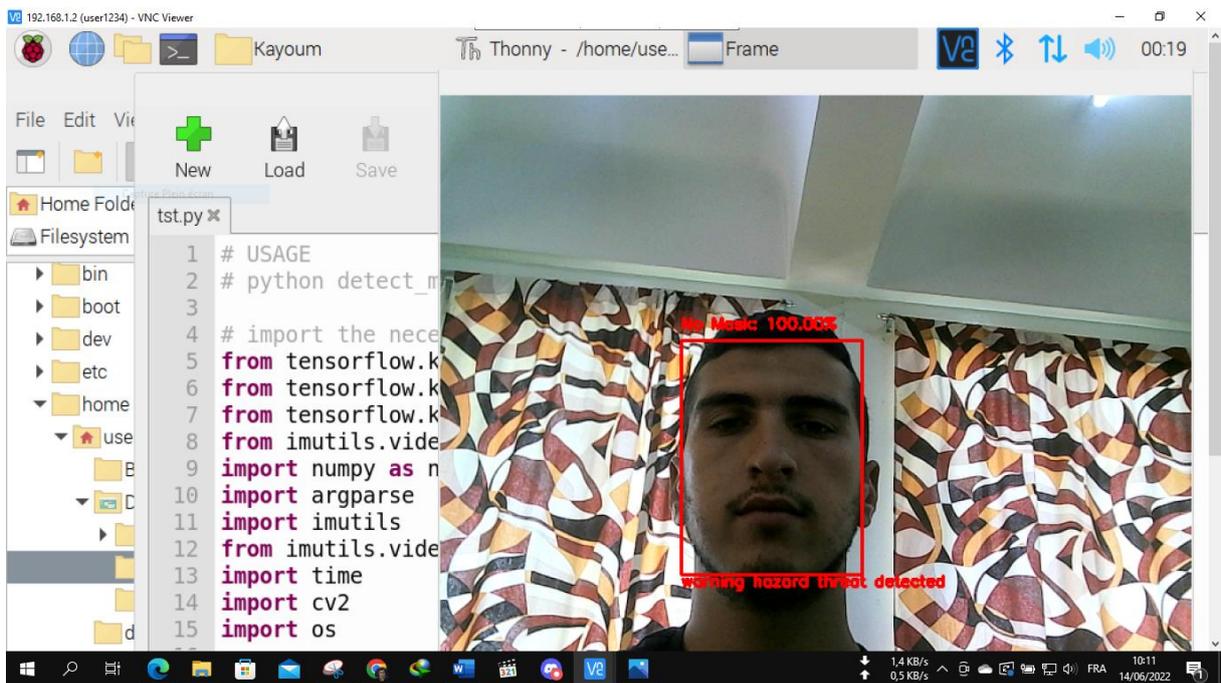
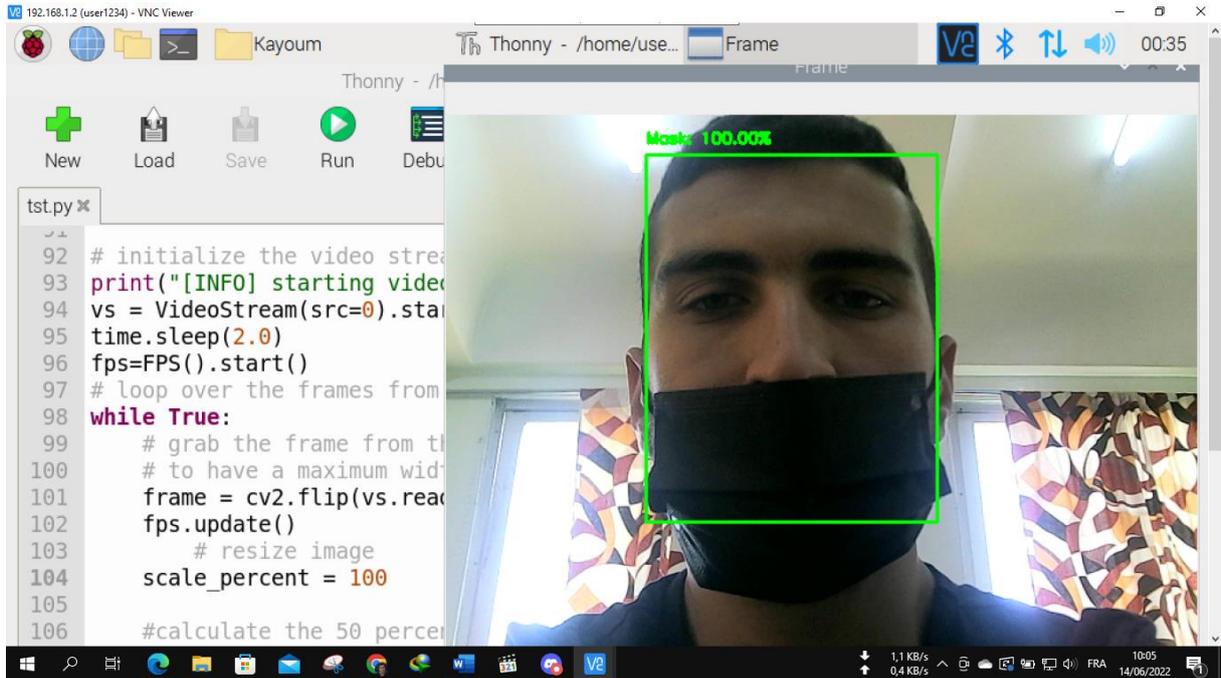
En revanche, si la personne ne porte pas de masque, le résultat sera le suivant:



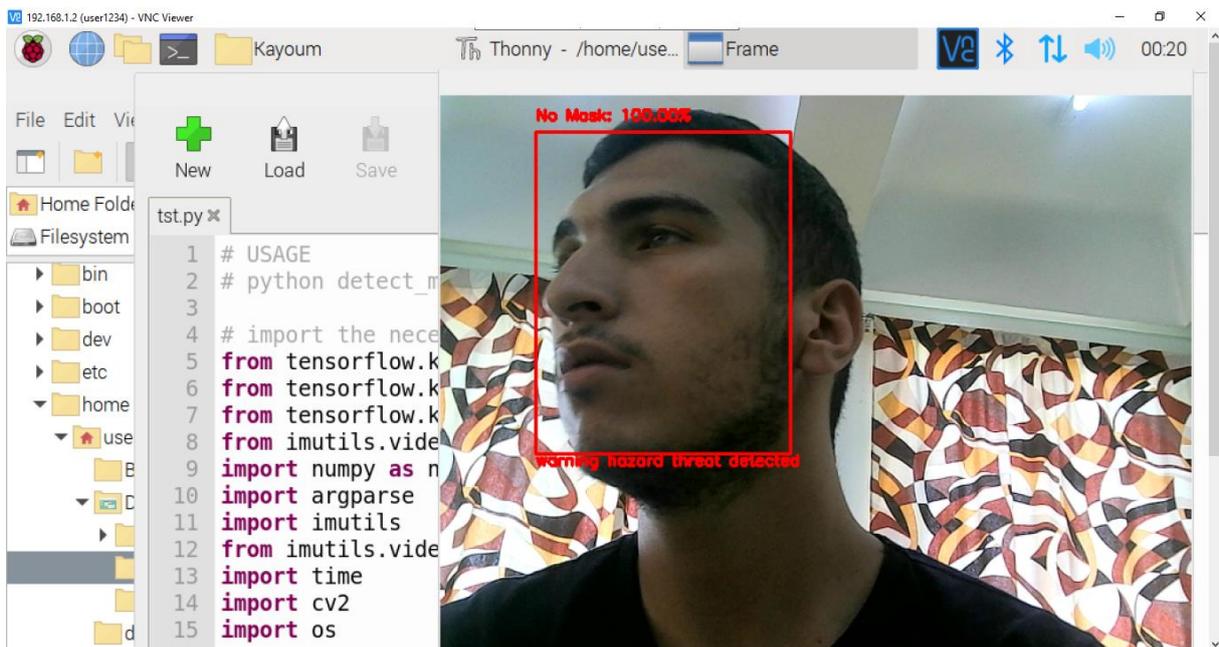
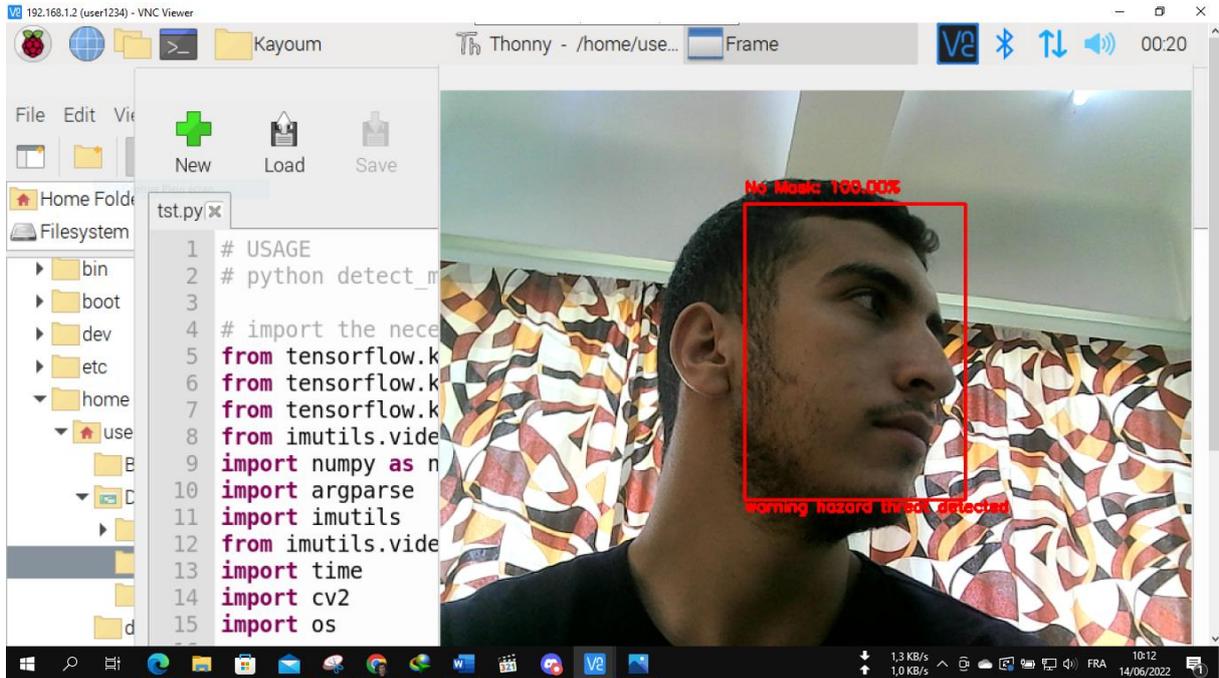




Chapitre 3 : Conception et validation



Chapitre 3 : Conception et validation



Ensuite, nous avons comparé les temps D'exécution des différents matériels utilise :

Tableau 3 : Performances dans les tests en temps réel

Matériel	Version/Spécifications	FPS moyen	Temps D'exécution (ms)
PC Portable	Intel Core i3-1115G4 @ 3.00GHz 3.00 GHz RAM 8gb	FPS \approx 7.4	0.41
	Intel Core i5-4300U @ 1.9.00GHz 2.50 GHz RAM 4gb	FPS \approx 5.2	0.68
Raspberry pi	Raspberry Pi 3 Modèle B ARM Cortex-A53 RAM 1GB	FPS \approx 0.43	0.97
	Raspberry Pi 4 Modèle B ARM Cortex-A72 RAM 4GB	FPS \approx 1.07	0.61

Conclusion :

Dans ce chapitre, nous avons présenté une approche de contrôle du port de masque très importante en se basant sur la programmation orientée objet qui se repose sur l'intelligence artificielle d'un côté. L'algorithme proposé détecte d'abord le visage de cette personne, puis vérifie si un masque est mis ou non. Nous avons vu la méthode que nous avons fait et la procédure du travail appliqué avec les algorithmes du face détections et utilisé dans Visual studio code, Google Colab, le Raspberry pi 3 et le Raspberry pi 4. le Raspberry pi 4 est plus performant comparais aux autre appareil.

Conclusion Générale

COVID-19, l'épidémie de virus qui prévaut, nous a fait reconnaître les avantages des masques faciaux. L'utilisation de masques faciaux. Dans ce travail, un système de détection de masque facial basé sur des classificateurs CNN profonds.

En résumé, nous avons développé un prototype de détecteur de masque facial basé sur des dispositifs embarqués et des modèles avancés d'IA. Nous avons affiné, entraîné et testé le détecteur de masque facial basé sur des modèles de détection de visage pré-entraînés CNN. Ensuite, nous avons utilisé une carte Raspberry pi avec sa caméra pour détecter les visages des personnes et voir si elles portent ou non leur masque. Lorsqu'un visage détecté ne porte pas le masque facial, des alertes sont générées en temps réel. Compte tenu de l'importance de porter correctement un masque facial pendant la pandémie de coronavirus.

À l'avenir, notre système de détection de masque facial pourra être utilisé dans les aéroports, les centres commerciaux et autres lieux de circulation pour détecter et renforcer l'importance du port du masque facial.

- [1] Kumar, T. A., Rajmohan, R., Pavithra, M., Ajagbe, S. A., Hodhod, R., & Gaber, T. (2022). Automatic face mask detection system in public transportation in smart cities using IoT and deep learning. *Electronics*, 11(6), 904
- [2] Murray, S. (2020). COMMENT SE PROPAGE LA COVID-19 ? LA CONTROVERSE ET LES PREUVES.16/05/2022
- [3] https://fr.wikipedia.org/wiki/Pand%C3%A9mie_de_Covid-19 16/05/2022
- [4] Thakur, V., Bhola, S., Thakur, P., Patel, S. K. S., Kulshrestha, S., Ratho, R. K., & Kumar, P. (2021). Waves and variants of SARS-CoV-2: understanding the causes and effect of the COVID-19 catastrophe. *Infection*, 1-16.16/05/2022
- [5] Banse, E., Bigot, A., De Valkeneer, C., Lorant, V., Luminet, O., Nicaise, P., ... & Guillaume, A. (2020). Quelques enseignements sur les impacts sociaux et économiques de la stratégie de réponse à la pandémie du coronavirus en Belgique. *Louvain médical*, 139(5), 375. 16/05/2022
- [6] <https://www.canada.ca/en/public-health/services/diseases/2019-novel-coronavirusinfection/prevention-risks.html>
- [7] https://fr.wikipedia.org/wiki/Maladie_%C3%A0_coronavirus_2019#cite_note-312
<https://solidarites-sante.gouv.fr/archives/archives-presse/archives-communiques-depresse/article/covid-19-procedure-d-arret-de-travail-simplifiee-pour-les-personnesvulnerables> 21/05/2022
- [8] <https://sante.journaldesfemmes.fr/fiches-maladies/2609189-masque-covid-19-coronavirus-transparent-enfant-chirurgical-lavable-tissu-ffp2-achat-afnor-prix/>
- [9] <https://www.sciencedirect.com/science/article/pii/S2667096821000422>
- [10] https://en.wikipedia.org/wiki/Computer_vision#Definition
- [11] Dagli, C. H. (Ed.). (2012). *Artificial neural networks for intelligent manufacturing*. Springer Science & Business Media. [PDF] *Advanced applications of neural networks and artificial intelligence: A review*
- [12] https://en.wikipedia.org/wiki/Computer_vision#Related_fields 62 Références
- [13] <https://www.annalsofrscb.ro/index.php/journal/article/view/3749>

- [14] Maksimović, M., Vujović, V., Davidović, N., Milošević, V., & Perišić, B. (2014). Raspberry Pi as Internet of things hardware: performances and constraints. *design issues*, 3(8), 1-6.
- [15] 'Raspberry Pi 4 vs Raspberry Pi 3B+', *The MagPi magazine*. <https://magpi.raspberrypi.org/articles/raspberry-pi-4-vs-raspberry-pi3b-plus> 09/06/2022
- [16] https://fr.m.wikipedia.org/wiki/Visual_Studio_Code?fbclid=IwAR0zgPxprpdBy7iaEapEMpyac_dTx3InuIsi7kH3PaKW9b6G-QE6HMQtVnc
- [17] <https://research.google.com/colaboratory/faq.html?hl=fr#:~:text=Colaboratory%2C%20souvent%20raccourci%20en%20%22Colab, donn%C3%A9es%20et%20%C3%A0%20l'%C3%A9ducation.>
- [18], 'Classification of Fake News by Fine-tuning Deep Bidirectional Transformers based Language Model', p. 163973, 05/06/2022
- [19], 'A comparative study of fine-tuning deep learning models for plant disease identification', *Computers and Electronics in Agriculture*, 05/06/2022
- [20] 'Brain tumor classification for MR images using transfer learning and fine-tuning', *Computerized Medical Imaging and Graphics*, 05/06/2022
- [21] *Hands-On Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras*, 05/06/2022
- [22] <https://fr.coursera.org/projects/data-augmentation-keras>
- [23] 'MobileNetV2: Inverted Residuals and Linear Bottlenecks', arXiv:1801.04381 [cs], Mar. 2019, Accessed: Oct. 16, 2020. [Online]. Available: <http://arxiv.org/abs/1801.04381>
- [24] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).
- [25] <https://github.com/inmicro/Face-Mask-Detection-Raspberry-pi-2020>