

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ DE 8 MAI 1945 - GUELMA -
FACULTÉ DES MATHÉMATIQUES, D'INFORMATIQUE ET DES SCIENCES DE LA MATIÈRE

Département d'Informatique



Mémoire de Fin d'études Master

Filière : Informatique

Option : Science et Technologie de l'Information et de la Communication

Thème

Systeme de Détection du Port de Masque dans les Salles de Cours par Deep learning (Application Implémentée sur Raspberry pi)

Encadré Par :

DR. BENCHERIET

CHEMESSE ENNEHAR

Présenté par :

LAHIOUEL RIDHA

15^e Juin 2022

Remerciement

Nous remercions Allah le tout puissant et miséricordieux, qui ma a donné la force et la patience d'accomplir ce modeste travail.

j'exprime ma profonde gratitude au Dr. BENCHERIET Chemesse Ennehar, qui m'a fait l'honneur d'avoir veillé et dirigé ce travail. j'ai eu le privilège de bénéficier de son enseignement, de son savoir et de ses conseils pertinents.

Veillez bien madame recevoir ma grande estime pour le grand honneur que vous m'avez fait d'accepter l'encadrement de ce travail.

Je tiens aussi à remercier vivement la responsable du laboratoire LAIG Dr Kechida Sihem, et bien à l'ingénieur du laboratoire LAIG monsieur Fisli Sofiane

Je tiens à exprimer mes considérations à nos membres de jury.

Enfin je remercie toute ma famille et tout particulièrement mes parents pour leur patience et leur soutiens au long de mon parcours.

Dédicace

Je Dédie ce modeste travail à :

Mes chers parents, pour leur endurance, leurs encouragements, leurs prières et surtout leurs immenses sacrifices ont été derrière la réussite de mes études

Mon frère Alouli et sœurs Asma, Yasmine et Sonia, en reconnaissance de leur affection toujours constante

Tous mes proches et amis qui mon toujours soutenu et encouragé au cours de la réalisation de ce mémoire

A tous ceux qui portent le nom de Lahiouel et Boukhari

Aux étudiants de ma promotion et plus spécialement à mon chère ami Anis

A Tous ceux qui mos ont aidés dans la réalisation de ce mémoire

Résumé

Depuis 2019 le monde est confronté à une énorme crise sanitaire en raison de la transmission rapide du coronavirus (COVID-19). Plusieurs directives ont été émises par l'Organisation Mondiale de la Santé (OMS) pour la protection contre la propagation du coronavirus.

Selon l'OMS la mesure préventive la plus efficace contre le virus consiste à porter un masque dans les lieux publics et les zones surpeuplées. Il est cependant très difficile de surveiller les personnes dans ces zones. Une solution intelligente a été proposée par les chercheurs c'est de mettre en œuvre des caméras intelligentes dans les lieux (publics et fermés) capables de détecter et de signaler les personnes qui ne portent pas de masques. Notre projet consiste en la conception d'un système intelligent de détection du port de masque dans les salles de cours basé sur les réseaux à convolution profond (DCNN). Le système mis en œuvre est embarqué sur RaspBerry Pi 3. Il a été construit en affinant le modèle d'apprentissage en profondeur « Inception V3 » entrainé sur la base Images 12K contenant plus de 12 000 images de trois classes de personnes : (1) avec des masques portés correctement, (2) avec des masques mal portés et (3) sans masques.

Les tests ont été effectués sur des images prises dans les salles de cours (Amphithéâtres), les salles de travaux dirigés et les salles de travaux pratiques. Les résultats obtenus sont très prometteurs mais un affinement de l'apprentissage, surtout sur la classe des masques mal portés, permettra de réduire les erreurs de détection et de confusion de cette classe et par conséquent de booster la fiabilité de notre système.

Mot clé : : Masque de protection, Détection du port de masque, Réseaux à convolution profond (DCNN), Apprentissage profond, Inception V3, Rasperry Pi 3.

Abstract

Since 2019 the world has been facing a huge health crisis due to the rapid transmission of the coronavirus (COVID-19). Several guidelines have been issued by the World Health Organization (WHO) for protection against the spread of the coronavirus.

According to the WHO, the most effective preventive measure against the virus is to wear a mask in public places and crowded areas. However, it is very difficult to monitor people in these areas. The researchers have proposed an intelligent solution to implement smart cameras in places (public and closed) capable of detecting and reporting people who are not wearing masks.

Our project involves designing an intelligent system for detecting mask-wearing in classrooms based on deep convolutional networks (DCNN). The implemented system is embedded in Raspberry Pi 3. It was built by refining the “Inception V3” deep learning model trained on 12K Images dataset containing more than 12,000 images of three classes of people : (1) with masks worn correctly, (2) with masks worn incorrectly, and (3) without masks.

The tests were carried out on images taken in classrooms (Amphitheatres), tutorial rooms, and practical work rooms. The results obtained are encouraging, but a refinement of the learning, especially in the class on badly worn masks, will make it possible to reduce the errors of detection and confusion in this class and consequently boost the reliability of our system.

Keywords :Protective Mask, Mask Wearing Detection, Deep Convolutional Networks (DCNN), Deep Learning, Inception V3, Raspberry Pi 3.

ملخص

منذ عام 2019، يواجه العالم أزمة صحية ضخمة بسبب الانتقال السريع لفيروس كورونا (COVID-19). تم إصدار العديد من الإرشادات من قبل منظمة الصحة العالمية (OMS) للحماية من انتشار فيروس كورونا.

وفقًا لمنظمة الصحة العالمية، فإن الإجراء الوقائي الأكثر فعالية ضد الفيروس هو ارتداء قناع في الأماكن العامة والمناطق المزدحمة. ومع ذلك، من الصعب للغاية مراقبة الناس في هذه المناطق. اقترح الباحثون حلاً ذكيًا يتمثل في استخدام كاميرات ذكية في الأماكن (العامة والمغلقة) القادرة على اكتشاف الأشخاص الذين لا يرتدون الأقنعة والإبلاغ عنها.

يتكون مشروعنا من تصميم نظام ذكي للكشف عن ارتداء القناع في الفصول الدراسية بناءً على الشبكات التلافيفية العميقة (DCNN). تم تضمين النظام المنفذ في 3 Pi RaspBerry. وقد تم إنشاؤه من خلال تحسين تدريب نموذج التعلم العميق "V3 Inception" استنادًا إلى 12 كيلو صور الذي يحتوي على أكثر من 12000 صورة لثلاث فئات من الأشخاص (1) مع ارتداء الأقنعة بشكل صحيح، (2) مع الأقنعة التي يتم ارتداؤها بشكل غير صحيح، و (3) بدون أقنعة.

تم إجراء الاختبارات على صور تم التقاطها في الفصول الدراسية (المدرجات) وغرف الاعمال الموجهة وغرف الاعمال التطبيقية. النتائج التي تم الحصول عليها واعدة للغاية، ولكن تحسين التعلم، خاصة على فئة الأقنعة التي تم ارتداؤها بشكل سيئ، سيجعل من الممكن تقليل أخطاء الكشف والارتباك في هذه الفئة وبالتالي تعزيز موثوقية نظامنا.

الكلمات المفاتيحية: القناع الواقي، كشف ارتداء القناع، الشبكات التلافيفية العميقة (DCNN)، التعلم العميق، V3 Inception، 3 Pi RaspBerry.

Table des matières

Liste des Figures	11
Liste des Tableaux	14
Introduction générale	15
Chapitre 1 Etat de L'art	17
1 Introduction	17
2 Le masque comme objet de protection	17
3 Le masque poursuit son développement	19
4 Masques et demi-masques (jetables ou réutilisables)	19
5 Types de masques	19
5.1 Masques chirurgicaux	19
5.2 Masques de protection pièce faciale filtrante (FFP)	20
5.3 Les masques barrières (en Tissu)	20
5.4 Masque à gaz	21
5.5 Masque à oxygène	22
6 Coronavirus(COVID-19)	22
7 La transmission du COVID-19	23
8 Base des données	24
8.1 Base 'RMFD'	24
8.2 Base 'SMFD'	24
8.3 Base 'LFW'	25
8.4 Base 'MD-2'	25
8.5 Base 'MD-3'	26
8.6 Base 'CelebFaces (Synthetic CelebA)'	27
8.7 Base 'MMD'	28

9	Quelques recherches dans le domaine	29
9.1	Article 01 [2]	29
9.2	Article 02 [3]	30
9.3	Article 03 [4]	31
9.4	Article 04 [5]	32
9.5	Article 05 [6]	33
9.6	Article 06 [7]	34
9.7	Article 07 [8]	35
9.8	Article 8 [9]	37
9.9	Article 09 [10]	37
9.10	Article 10 [11]	38
10	Conclusion	39
Chapitre 2 Apprentissage Profond et Réseaux de Neurones Convolutifs		40
1	Introduction	40
2	Relation entre IA/ML/DL	40
3	Apprentissage automatique (Machine Learning)	41
3.1	Apprentissage supervisé	42
3.2	Apprentissage non-supervisé	42
3.3	Apprentissage par-renforcement	42
4	Avantages de l'apprentissage automatique	43
5	Inconvénients de l'apprentissage automatique	43
5.1	Neurone biologique	44
5.2	Neurone formel	45
6	Apprentissage en profondeur (Deep Learning)	46
6.1	Historique	46
6.2	Définition	46
7	Machine Learning VS Deep Learning	47
8	Avantages du Deep Learning	48
9	Inconvénient du Deep Learning	48
10	Domaines d'application de Deep Learning	49
11	Différentes Architectures du Deep Learning	49
11.1	Réseaux neuronaux convolutifs (CNN)	49

11.2	Réseau de neurones récurrents (RNN)	50
11.3	Modèle génératif	51
12	Structure des réseaux de neurones convolutifs	52
12.1	Architecture des réseaux de neurones convolutifs	52
12.2	La couche de convolution	52
12.3	Fonctions d'activations	54
12.4	Couche de Pooling	56
12.5	Couche entièrement connectée (FC :Fully Connected)	57
12.6	La couche de perte (LOSS)	58
13	Quelques réseaux convolutifs célèbres	58
13.1	LeNet-5 [17]	58
13.2	AlexNet [18]	58
13.3	VGG-16 [19]	59
13.4	Inception-v4 [20]	59
13.5	Inception-ResNet-V2 [20]	60
13.6	ResNet-50 [21]	60
14	Construction d'un modèle d'apprentissage en profondeur	61
14.1	Étape 1 : Rassembler l'ensemble de données	61
14.2	Étape 2 : Fractionner l'ensemble de données	62
14.3	Étape 3 : Former le réseau	62
14.4	Étape 4 : Évaluation	63
14.5	Mesures de performances	63
15	Conclusion	65

Chapitre 3 Conception **66**

1	Introduction	66
2	Architecture du système	66
3	Redimensionnement de l'image	69
4	Détection de visage	69
4.1	Étape 1 : Le réseau de propositions (P-Net)	70
4.2	Étape 2 : Le réseau Refine (R-Net)	70
4.3	Étape 3 : Le réseau de sortie (O-Net)	71
5	Réseaux Inception	71

5.1	InceptionV1	72
5.2	InceptionV2	74
5.3	InceptionV3	76
5.4	Préparation de base de données	77
5.5	Architecture détaillé des bloc	77
5.6	Détail de l'architecture du reseau InceptionV3	78
6	Conclusion	81

Chapitre 4 Implémentions et résultats 82

1	Introduction	82
2	Matériel utilisé	82
2.1	Fiches techniques Raspberry Pi	82
2.2	Mise on marche de Raspberry Pi	85
2.3	Station de calcul	88
3	Logiciel utilisé	90
3.1	Python	90
3.2	OpenCV-Python	90
3.3	Numpy	90
3.4	Pandas	91
3.5	Matplotlib	91
3.6	TensorFlow	91
3.7	Keras	92
3.8	Tkinter	92
3.9	Os	92
4	Base d'apprentissage	92
5	Apprentissage et Test	93
5.1	Apprentissage du réseau 'InceptionV3'	93
5.2	Test de réseaux 'InceptionV3'	95
6	Interface graphique	95
7	Quelques tests sur des images dans les salles de cours	98
8	Conclusion	101

Conclusion générale 103

Bibliographie

106

Webographie

108

Table des figures

1.1	Les premiers masques de protection	18
1.2	Masques Chirurgicaux	20
1.3	Masques FPP	20
1.4	Masque en Tissu	21
1.5	Masque à Gaz	21
1.6	Masque à Oxygène	22
1.7	Virus COVID-19	23
1.8	principaux modes de transmission	23
1.9	Exemples d'images de l'ensemble de données 'RMFD'	24
1.10	Exemples d'images de l'ensemble de données 'SMFD'	25
1.11	Exemples d'images de l'ensemble de données 'LFW'	25
1.12	Exemples d'images de l'ensemble de données 'MD-2'	26
1.13	Exemples d'images de l'ensemble de données 'MD-3'	27
1.14	Exemples d'images de l'ensemble de données 'celebA'	28
1.15	Exemples d'images de l'ensemble de données 'MMD'	28
2.1	Relation entre "IA, ML, et le Deep Learning"	41
2.2	Algorithmes d'apprentissage automatique [w8]	44
2.3	Neurone biologique	45
2.4	Neurone formel	45
2.5	Machine Learning VS Deep Learning[w12]	48
2.6	Exemple de réseau convolutif (CNN)	50
2.7	Exemple d'une couche RNN simple à trois entrée et deux sorties.	51
2.8	Exemple de modèle génératif (CNN)	51
2.9	Exemple d'une convolution	53
2.10	Diffèrent type de convolution	54
2.11	Quelques fonctions d'activation.	55

2.12	Application de ReLU à partir d'une matrice.	55
2.13	Exemple des trois type de pooling cité.	56
2.14	Exemple de mise à plat des images finales en sortie	57
2.15	Exemple d'opération entièrement connecte (FC)	57
2.16	Architecture 'LeNet-5'	58
2.17	Architecture 'AlexNet'	59
2.18	Architecture 'VGG-16'	59
2.19	Architecture 'Inception-ResNet-V2'	60
2.20	Architecture 'ResNet-50'	61
2.21	Exemples de fractionnement commun des données de formation et de test.	62
2.22	la courbe ROC.	65
3.1	Architecture du système globale	67
3.2	Architecture de l'entraînement	68
3.3	Architecture de test	68
3.4	L'image d'entrée on forme de pyramide d'image	69
3.5	P-Net	70
3.6	R-Net	71
3.7	O-Net	71
3.8	Architecture naïf d'Inception	72
3.9	Architecture Inception avec reduction	72
3.10	Architicture Inception V1	73
3.11	Architecture inception v2	74
3.12	Factorisation de convolution 5x5	75
3.13	Factorisation de convolution 1xn et nx1	75
3.14	Banque de filtres d'inception	76
3.15	Architecture Inception-V3	77
3.16	Architecture du inception A	78
3.17	Architecture du Inception B	79
3.18	Architecture du Inception C	79
3.19	Architecture Réduction Inception A	80
3.20	Architecture Réduction Inception B	80
3.21	Architecture Bloc classificateur auxiliaire	81

4.1	Raspberry Pi	84
4.2	Les composants de Raspberry Pi 3	84
4.3	Caméra Raspberry Pi	85
4.4	L'interface Etcher	86
4.5	Bancher la caméra	87
4.6	Configuration de la caméra	88
4.7	Station de calcul du laboratoire LAIG	89
4.8	Matériels utilisé dans notre application	89
4.9	Exemples d'images de l'ensemble de données '12K-images'	93
4.10	le graphe de 'accuracy' pour apprentissage de 'InceptionV3'	94
4.11	le graphe de 'loss' pour d'apprentissage de 'InceptionV3'	94
4.12	Matrice de confusion	95
4.13	L'interface graphique	96
4.14	chargement de l'image dans l'interface	96
4.15	Détection de visages	97
4.16	Résultat du modèle	97
4.17	Amphi théâtre n°6 de l'université 8 Mai 1945 -Guelma-	98
4.18	Amphi théâtre de l'université 8 Mai 1945 -Guelma-	99
4.19	Salle de TD de l'université 8 Mai 1945 -Guelma-	99
4.20	Salle de TD de l'université 8 Mai 1945 -Guelma-	100
4.21	Salle de TP de l'université 8 Mai 1945 -Guelma-	100
4.22	Salle de TP de l'université 8 Mai 1945 -Guelma-	100
4.23	salle de conférence externe	101
4.24	salle de conférence externe	101

Liste des tableaux

1.1	Informations sur les ensembles de données 'MD-2'	26
1.2	Informations sur les ensembles de données 'MD-3'	26
1.4	Illustration des approches MFR les plus récentes	36
2.1	Matrice de confusion.	63
4.1	Base "Détection de masque facial-Images 12K" adaptée pour InceptionV3 .	93

Introduction générale

Face à la transmission de la maladie du coronavirus (COVID-19), l'Organisation mondiale de la santé (OMS) a exigé le port du masque. Avant COVID-19, seules quelques personnes portaient des masques pour se protéger de la pollution de l'air, et les professionnels de la santé lorsqu'ils pratiquaient dans les hôpitaux. Avec la transmission rapide du virus, les scientifiques ont conseillé le port du masques dans les lieux publics pour empêcher la transmission de la maladie.

Les techniques d'intelligence artificielle (IA) telles que l'apprentissage automatique "Machine Learning"(ML) et l'apprentissage en profondeur "Deep Learning" (DL) peuvent être utilisées de plusieurs manières pour prévenir la transmission de la maladie. Une solution intelligente pour détection de masque a été proposée.

L'objectif de notre travail est de crée une application embarquée sur Raspberry Pi 3 afin de repérer les étudiants qui portent des masques dans les salles de cours. pour cela nous avons utilisé le réseaux convolutif "InceptionV3" entraîne sur 3 classes "masque porté correctement", "masque mal porté" et "sans masque".

Nous avons partagé notre travail sur quatre chapitres :

- **Chapitre 1** : fait l'objet d'une étude détaillée sur les masques et leurs historiques suivi d'un état de l'art sur quelques études récentes sur la détection du port de masque.
- **Chapitre 2** : consacré à un étude détaillée sur les réseaux de neurones convolutif et l'apprentissage profond.
- **Chapitre 3** : dédié à la conception et l'architecture détaillée du système proposé.

- **Chapitre 4** : les principaux résultats obtenus et leurs interprétations seront détaillés dans ce chapitre.

Et nous clôturons cette étude par une conclusion et des perspectives pour des travaux futurs complémentaires à ce projet.

Chapitre 1

Etat de L'art

1 Introduction

Au cours des deux dernier années l'humanité vie sur l'empire d'un nouveau virus connu sous le nom de coronavirus(COVID-19) qui touche la santé humaine et ce propage rapidement entre les humains dans les lieux publics et fermes comme les supermarchés, les universités, les aéroports, ...etc.

Les recherches scientifiques, ont prouvé que le port de masque faciaux dans les lieux (publics et fermes) empêche la transmission du COVID-19. A fin d'empêcher la propagation de la maladie dans les lieux publics d'une principalement au non respect du port de masque, une solution intelligente à été proposé par les chercheurs c'est de mettre en oeuvre des caméras intelligente dans les lieux (publics et fermés) capable de détecter et de signaler les personnes qui ne portent pas de masque.

2 Le masque comme objet de protection

Fabriqués à partir de vessie animale, les premiers masques de protection apparaissent sous l'empire romain. Ils sont utilisés au fond des mines pour se protéger des vapeurs toxiques Figure 1.1a . Le même dispositif est aussi utilisé dans les ateliers où l'on réduit et affine le minium (ou cinabre).

Au XVIe siècle, Léonard de Vinci [w1] adapte le principe et propose l'utilisation d'un tissu imbibé d'eau à placer sur la bouche des navigateurs pour les prémunir d'éventuelles

attaques chimiques lors des batailles navales. La période du XIVE jusqu'à XVIIIe siècle, est marquée par plusieurs épidémies de peste qui secouent l'Europe. Pour les médecins de l'époque, la contagion se propage par la bouche et le nez. Charles de Lorme [w1], premier médecin de Louis XIII, imagine alors en 1619 un masque en carton bouilli doté d'un bec à deux trous permettant la respiration Figure 1.1b. Et pour en parfaire son efficacité, il peut contenir une grande variété de produits aux propriétés désinfectantes, tels que des fleurs séchées, des herbes aromatiques, des épices ou du camphre.

Dès le XVIIIe siècle, l'Italien Bernardino Ramazzini [w1], parmi les différents dispositifs mis en œuvre, recommande aux travailleurs dans des marais insalubres d'utiliser une étoffe ou une gaze sur le visage (sur le nez et la bouche) d'une éponge humide ou d'un tissu fin et serré en forme de cône creux Figure 1.1c. Il suffit d'ajouter à l'eau du vinaigre, ou d'autres produits comme l'eau de chaux pour protéger les fabricants de colle-forte, et même le personnel des hôpitaux. A la fin du XIXe siècle, le médecin allemand Carl Flügge [w1], soupçonnant la propagation de la maladie entre médecins et patients à travers la salive, plaide auprès de ses confrères pour le port du masque lors des opérations chirurgicales Figure 1.1d . Il semble que ce soit le chirurgien Paul Berger qui lors d'une intervention à l'hôpital Tenon à Paris en octobre 1897 enfile le premier un tel masque. Un accessoire devenu aujourd'hui un incontournable des salles d'opération [w1].



(a)



(b)



(c)



(d)

FIGURE 1.1 – Les premiers masques de protection

3 Le masque poursuit son développement

Composé généralement de couches de coton avec parfois une couche extérieure complémentaire imperméable aux gouttelettes, le masque se développe en mélangeant les propriétés antibactériennes, antipoussières et antigaz. De nombreux brevets furent déposés. Dans tous les cas, il est lavable et parfois stérilisable [1].

4 Masques et demi-masques (jetables ou réutilisables)

Il existe deux types de masques sont :

- Jetables :utilisés une seul fois comme les masques chirurgicaux ,masques de protection,..etc.
- Réutilisables(non jetables) : ce sont des masques lavables utilisés plusieurs fois comme les masques en tissu.

5 Types de masques

5.1 Masques chirurgicaux

Un masque chirurgical (aussi nommé antiprojections, ou d'hygiène ou à trois plis) est un masque respiratoire à usage unique conçu pour limiter la propagation vers l'environnement très proche des germes (bactéries, virus) depuis la bouche, le nez et les voies respiratoires de la personne qui le porte. Ce type de ce masque est porté par les professionnels de la santé durant une intervention chirurgicale pour protéger le champ opératoire stérile et empêcher une infection du patient qui s'y trouve. Bien qu'il ne soit pas conçu pour cela, il est aussi souvent porté pour se protéger d'une part contre la pollution atmosphérique et d'autre part contre d'éventuelles infections, comme lors de la pandémie de Covid-19. Cet usage généralisé cause aujourd'hui une pénurie mondiale de ces masques : c'est un problème central de la pandémie de coronavirus [w1].



FIGURE 1.2 – Masques Chirurgicaux

5.2 Masques de protection pièce faciale filtrante (FFP)

Le masque est 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 existe trois catégories de masques FFP [w2] :

- FFP1 Figure(1.3a) : Non-médicaux (anti-poussières), filtration d'au moins 80% des particules en suspension
- FFP2 Figure(1.3b) : médicaux, filtration d'au moins 94% des particules en suspension
- FFP3 Figure(1.3c) : médicaux, filtration d'au moins 99% des particules en suspension

les

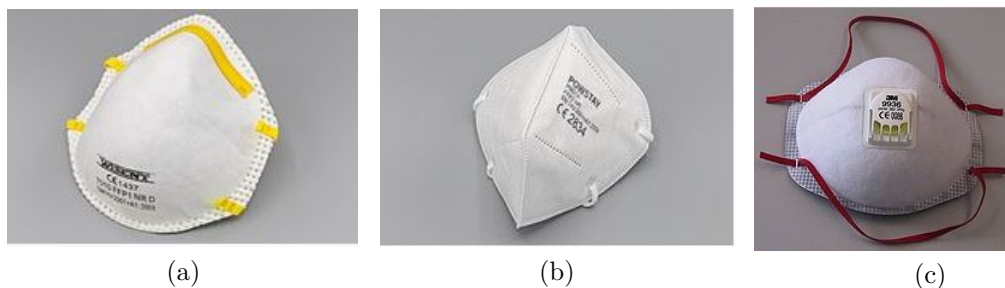


FIGURE 1.3 – Masques FFP

5.3 Les masques barrières (en Tissu)

Le masque barrière(pour le grand public) est un masque en tissu, lavable, destiné au public pour favoriser une protection collective. généralement des tissus en coton. Il est lavable et réutilisable. Il existe deux catégories de masques barrières [w3] :

- Usage Non Sanitaire UNS (catégorie 1) Figure(1.4a) : il assure au moins 90% de filtration de particules de 3 microns.
- Usage Non Sanitaire UNS (catégorie 2) Figure(1.4b) : il assure au moins 70% de filtration de particules de 3 microns.

La principale différence entre les deux catégories c'est le taux de filtration de particules, et les masques de catégorie 1 sont lavable mais les masques de catégorie 2 sont jetable.



(a)



(b)

FIGURE 1.4 – Masque en Tissu

5.4 Masque à gaz

Un masque à gaz, ou masque anti-gaz, est une protection contre certaines armes chimiques et armes biologiques et contre les agents radioactifs. Il se porte sur le visage et couvre les entrées des voies respiratoires (bouche, nez) et souvent les yeux ainsi que d'autres tissus sensibles du visage [w4].



FIGURE 1.5 – Masque à Gaz

5.5 Masque à oxygène

Un masque à oxygène est un masque fournissant de l'oxygène aux poumons à partir d'un réservoir de stockage. Il couvre généralement le nez et la bouche, voire parfois toute la tête et peut être en plastique, en silicone ou en caoutchouc [w5].



FIGURE 1.6 – Masque à Oxygène

6 Coronavirus(COVID-19)

Une épidémie de pneumonies, décrite à l'époque comme d'allure virale de cause inconnue a émergé dans la ville de Wuhan (province de Hubei, Chine) en décembre 2019. Le 9 janvier 2020, la découverte d'un nouveau coronavirus Figure 1.7, a été annoncée officiellement par les autorités sanitaires chinoises et l'Organisation mondiale de la santé (OMS). D'abord appelé 2019-nCoV puis SARS-CoV-2, ce virus est différent du virus SARS-CoV responsable de l'épidémie de SRAS en 2003. Il est également différent du virus MERS-CoV responsable d'une épidémie évoluant depuis 2012 au Moyen-Orient. Ce nouveau virus est l'agent responsable de cette nouvelle maladie infectieuse respiratoire appelée Covid-19 (pour CoronaVirus Disease 2019) [w6].

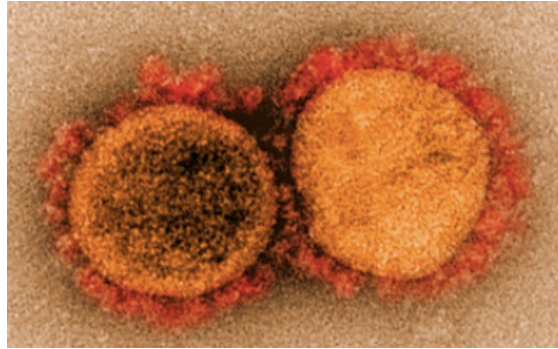


FIGURE 1.7 – Virus COVID-19

7 La transmission du COVID-19

La transmission se fait essentiellement par voie aérienne (gouttelettes de postillons émises au cours des efforts de toux mais aussi lors de la parole) et passe par un contact rapproché (moins d'un mètre) et durable (au moins 15 minutes) avec un sujet contagieux Figure 1.8. Le virus peut conserver une infectiosité pendant quelques heures sur des surfaces inertes d'où il peut être transporté par les mains ce qui justifie une bonne hygiène des mains [w6].

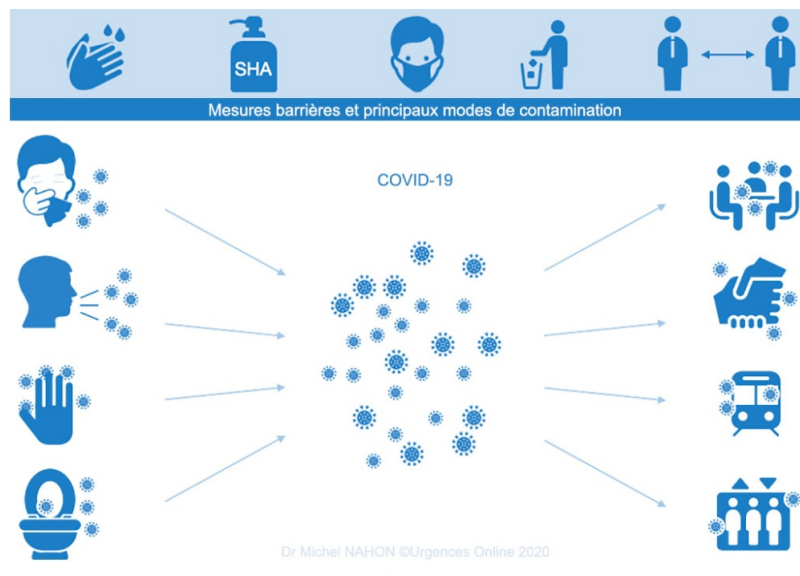


FIGURE 1.8 – principaux modes de transmission

8 Base des données

c'est un jeu de données utilisé dans l'entraînement pour la détection de masque. Nous présentons dans ce qui suit les 'data sets' les plus utilisés dans l'état de l'art.

8.1 Base 'RMFD'

Le jeu de données de visage masqué du monde réel (RMFD) [2], comprend 5000 visages masqués et 90 000 visages non masqués. Figure 1.9 illustre des échantillons des exemples d'images du RMFD [2].

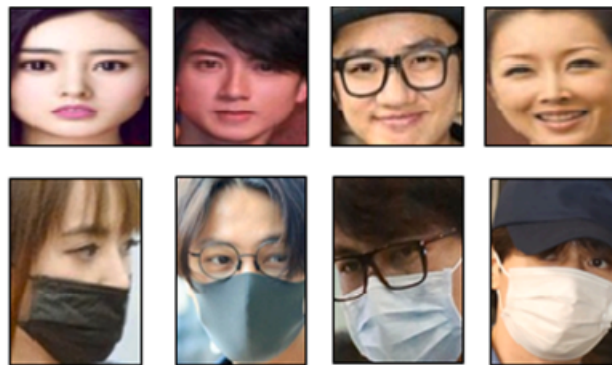


FIGURE 1.9 – Exemples d'images de l'ensemble de données 'RMFD'

8.2 Base 'SMFD'

Le jeu de données de visage masqué simulé (SMFD) [2], se compose de 1570 images, 785 pour les visages masqués simulés, 785 pour les visages non masqués. Figure 1.10 illustre des échantillons des exemples d'images du SMFD [2].



FIGURE 1.10 – Exemples d’images de l’ensemble de données ‘SMFD’

8.3 Base ‘LFW’

Le Labeled Faces in the Wild (LFW) [1]. Il s’agit d’un ensemble de données de visage masqué simulé qui contient 13000 visages masqués pour des célébrités du monde entier. La figure 1.11 illustre des échantillons des exemples d’images du LFW [1].



FIGURE 1.11 – Exemples d’images de l’ensemble de données ‘LFW’

8.4 Base ‘MD-2’

Le jeu de données Mask Detection-2 (MD-2) [4] contient deux catégories visage (a1) et masque facial (a2) se référant respectivement aux visages et aux visages avec masques[4]. Tableau 1.1 fournit la distribution par catégorie de chaque base de données. La figure 1.12 illustre quelques échantillons de la base de données ‘MD-2’.

Division de données	Nombres de données
Ensemble d'entraînement	6362
Ensemble de validation	1590
Catégorie a1 de l'ensemble d'entraînement	9937
Catégorie a2 de l'ensemble d'entraînement	3212
Moyenne de cases par image	2.067

TABLE 1.1 – Informations sur les ensembles de données 'MD-2'



FIGURE 1.12 – Exemples d'images de l'ensemble de données 'MD-2'

8.5 Base 'MD-3'

Le jeu de données Mask Detection-3 (MD-3) [4] comprend trois catégories, qui sont sans masque (b1), avec masque (b2) et masque porté incorrect (b3), où le dernier indique que le masque est mal porté [4]. Tableau 1.2 fournit la distribution par catégorie de chaque base de données. La figure 1.13 illustre quelques échantillons de la base de données 'MD-3'.

Division de données	Nombres de données
Ensemble d'entraînement	683
Ensemble de validation	170
Catégorie b1 de l'ensemble d'entraînement	567
Catégorie b2 de l'ensemble d'entraînement	2388
Catégorie b3 de l'ensemble d'entraînement	107
Moyenne de cases par image	4.490

TABLE 1.2 – Informations sur les ensembles de données 'MD-3'



FIGURE 1.13 – Exemples d'images de l'ensemble de données 'MD-3'

8.6 Base 'CelebFaces (Synthetic CelebA)'

CelebA [8] est un ensemble de données d'attributs de visage à grande échelle avec plus de 200 000 images de célébrités. Il a été construit à l'aide de 50 types de masques synthétiques de différentes tailles, formes, couleurs et structures. Lors de la construction des échantillons synthétiques, le visage a été aligné à l'aide des coordonnées oculaires pour toutes les images, puis le masque a été placé au hasard sur le visage à l'aide d'Adobe Photoshop [8]. La figure 1.14 illustre des échantillons des exemples d'images du jeu de données CelebA.



FIGURE 1.14 – Exemples d'images de l'ensemble de données 'celebA'

8.7 Base 'MMD'

Le premier ensemble de données est Medical Masks Dataset (MMD) [10] publié par Mikolaj Witkowski. L'ensemble de données MMD se compose de 682 images avec plus de 3 000 visages masqués médicaux portant des masques. 2 illustre des exemples d'images en MMD [10]. La figure 1.15 illustre des échantillons des exemples d'images du jeu de données MMD.



FIGURE 1.15 – Exemples d'images de l'ensemble de données 'MMD'

9 Quelques recherches dans le domaine

La détection du port de masques faciaux a été le centre de la recherche depuis la pandémie du coronavirus, les chercheurs ont commencé leurs recherches par les algorithmes de la reconnaissance de visage base sur l'apprentissage profond(deep learning) pour les modifier et les adapter au problème. plusieurs études ont été proposer et publiées dans les revues dont nous avons selectionné dix (parmi les plus importantes) pour une étude de l'état de l'art.

9.1 Article 01 [2]

- (an) **A deep transfer learning hybrid model with learning methods automatic for detecting face masks in the era of the COVID-19 pandemic**
- (fr) **Un modèle hybride d'apprentissage par transfert en profondeur avec des méthodes d'apprentissage automatique pour la détection de masques faciaux à l'ère de la pandémie de COVID-19**

Résumé

Dans cet article, un modèle hybride utilisant l'apprentissage automatique profond et classique pour la détection de masques faciaux a été présenté. Le modèle proposé comprend deux volets. Le premier composant est conçu pour l'extraction de caractéristiques à l'aide de Resnet50. Alors que le deuxième composant est conçu pour le processus de classification des masques faciaux à l'aide d'arbres de décision, d'une machine à vecteurs de support(SVM)et d'un algorithme d'ensemble. En fonction des résultats obtenus par cette recherche ils ont conclu que le classificateur d'ensemble est meilleur que les arbres de décision et le classificateur SVM en termes de validation, de précision des tests, de métriques de performance lorsque l'entraînement est dans L'ensemble de données dataSete3 (DS3) et une combinés de L'ensemble de données RMDF [2] et L'ensemble de données SMDF [2] (DS3). Alors que, lorsque la formation est terminée sur L'ensemble de données SMFD [2] (DS2), le classificateur SVM surpasse les autres classificateurs. De plus, en termes de temps consommé pour la formation, le classificateur SVM est le moins consommateur de

temps il a atteint de 0,33s tandis que pour l'ensemble et le classificateur d'arbres de décision étaient de 29,2s et 0,79s pour l'apprentissage de DS3. Pour sélectionner le meilleur classificateurs les auteurs ont utilisé les matrices de confusion. En ce qui concerne les arbres de décision, les matrices de confusion ne sont pas incluses dans cette section car elles ont atteint la précision de test la plus faible.

En fonction des résultats obtenus des matrice de confusion et précision de classe : le SVM et le classificateur d'ensemble ont été entrainer sur DS3 ont donné des résultats de précision très proches uniquement 0,01% de différence. A la fin la sélectionner été pour le classificateur SVM pour la raison du temps. Le seul inconvénient est de ne pas utiliser la plupart des méthodes d'apprentissage automatique classiques pour obtenir le temps de consommation le plus bas et la plus grande précision.

Base de données utilisée

Cette recherche a mené ses expérimentations sur trois ensembles de données originaux. Le premier jeu de données est le jeu de données de visage masqué du monde réel (RMFD) [2], 5000 images de visages avec et sans masques ont été utilisées avec un total de 10000 images. Le deuxième jeu de données est un jeu de données de visage masqué simulé(SMFD) [2], 1570 images, 785 images de visages avec et sans masques simulés. Les ensembles de données RMDF et SMFD (70% pour la formation, 10% pour la validation, 20% pour la phase du test).Le troisième ensemble de données utilisé dans cette recherche est le Labeled Faces in the Wild(LFW) [2]. Il s'agit d'un ensemble de données de visage masqué simulé qui contient 13000 visages masqués pour des célébrités du monde entier. L'ensemble de données LFW utilisé pour la phase de test uniquement.

9.2 Article 02 [3]

- (an) **Face Mask Detection Using Transfer Learning of InceptionV3**
- (fr) **Détection de masque facial à l'aide du transfert Apprentissage d'InceptionV3**

Résumé

Dans cet article, un modèle d'apprentissage par transfert est proposé pour automatiser le processus d'identification des personnes qui ne portent pas de masque. Le modèle proposé est construit en affinant le modèle d'apprentissage en profondeur de pointe pré-entraîné InceptionV3. L'objectif principal de cette recherche est de développer une approche automatisée pour aider les responsables gouvernementaux de divers pays à surveiller et imposer aux citoyens de porter des masques dans les lieux publics.

Dans ce travail, des techniques d'augmentation d'image sont utilisées pour améliorer les performances du modèle car elles augmentent la diversité des données d'apprentissage. Le modèle d'apprentissage par transfert proposé a atteint une précision et une spécificité de 100%, 100% pendant la formation et 99,92%, 99,9% pendant les tests sur l'ensemble de données SMFD [2].

Base de données utilisée

Cette recherche a mené ses expérimentations sur un jeu de données : SMFD [2] avec des visages masqués simulé comprenant 1570 images, 785 images de visages avec et sans masques simulés (70% pour l'entraînement et 30% pour le test du modèle).

9.3 Article 03 [4]

- (an) **FMD-Yolo : An efficient face mask detection method for COVID-19 prevention and control in public**
- (fr) **FMD-Yolo : une méthode efficace de détection des masques faciaux pour la prévention et le contrôle du COVID-19 en public**

Résumé

Dans cet article deux ensembles de données de masques faciaux ont été sélectionnés pour vérifier l'efficacité de FMD-Yolo (face mask detection FMD).

Le réseau FMD-Yolo est construit par trois composants principaux dans le réseau profond, qui sont Im-Res2Net-101, En-PAN et Classification + Régression. En particulier, Im-Res2Net-101 avec les bases de données MD-2 et MD-3. Sur la base de données MD-2, où

FMD-Yolo proposé avec huit autres algorithmes de détection sont : Base de référence sur RCNN plus rapide, RCNN plus rapide avec FPN, Yolo V3, Yolo V4, RetinaNet, FCOS, Dét Efficace et Réseau RH sont comparés sur cinq métriques (indicateurs) d'évaluation de carte, PA50, PA75, Mar et RA50. les resultats sur la base de données MD-2 l'algorithme FMD-Yolo fonctionne mieux sur tous les indicateurs carte, PA50, PA75, Mar et RA50 a atteint de 66,4%,92,0%, 80,0%, 77,4% et 95,5% par rapport aux autres algorithmes ; Il a donne une une amélioration de 4,6%, 1,8%, 5,5%, 10,0% et 2,3%. Même dans la base de données MD-3 l'algorithme FMD-Yolo fonctionne mieux sur tous les indicateurs carte, PA50, PA75, Mar et RA50 il a atteint de 57,5%, 88,4%, 64,3%, 69.9% et 92,9% par rapport aux autres algorithmes ; Et donne une amélioration de 3,9%, 3,8%, 3,9%, 7,1% et 0,2%.

Base de données utilisée

Cette recherche a mené ses expérimentations sur deux ensembles : MD-2 [4] avec 7952 images appartenant à 2 classes((a1)_sans masque et (a2)_avec masque), et MD-3 [4] avec 853 images appartenant à 3 classes ((b1)_sans masque, (b2)_avec masque, (b3)_masque porter incorrecte).

9.4 Article 04 [5]

- (an) **Increasing face mask detection with YOLO on a new dataset**
- (fr) **Augmenter la détection des masques faciaux avec YOLO sur un nouvel ensemble de données**

Résumé

L'idée de base de cet article tourne autour de l'ensemble de données [5] utilise considire comme nouveau car il n'a jamais été utilise sur des études avant la pandémie et la nouvelle génération de yolo proposée.

La nouveauté de ce travail réside également dans les expériences exhaustives menées pour effectuer la détection de masques faciaux sur un grand ensemble de données contenant des images de contraste et de classes variables.De plus, des modifications sont proposées dans de minuscules variantes de l'algorithme YOLO qui sont capables de s'intégrer aux systèmes

de reconnaissance faciale pour effectuer une détection de masques faciaux en temps réel le problème a été résolu en créant un ensemble de données richement annoté qui peut aider à faire progresser la recherche sur la détection des masques faciaux. Proposé en tant que détecteur de masques faciaux efficace et amélioré dans les applications du monde réel en raison de son réseau d'extraction de caractéristiques optimisé et amélioré le système proposé. a atteint la précision moyenne la plus élevée pour les masques faciaux avec une valeur de 70,37% qui le rend t-il est capable de détecter les masques avec précision.

Base de données utilisée

Cette recherche a mené ses expérimentations sur un nouvel ensemble de données [5] proposé se compose de 52 635 images (80% pour l'entraînement, 10% pour la validation, 10% pour la phase du test) de personnes portant des masques faciaux, de personnes ne portant pas de masques faciaux, de personnes portant des masques faciaux de manière incorrecte et, la zone de masque dans les images où un masque facial est présent.

9.5 Article 05 [6]

- (an) **IoT-enabled smart doors for body temperature monitoring and mask detection facials**
- (fr) **Portes intelligentes compatibles IoT pour la surveillance de la température corporelle et la détection des masques faciaux**

Résumé

Dans cet article, une approche de L'Internet of Things (IoT) qui utilise un modèle de deux phases la premier et la détection de masque et la deuxieme un détection de température. Ce modele proposé et composer de deux categories la premier et la detection de masque faciaux a laide de la methode de réseaux de neurones à convolution (CNN), où les auteurs ont implémenter l'algorithme de détection de masque en fonction de la bibliothèque de flux Tensor, qui contient deux parties : le modèle Face Detection et les différentes phases du modèle de détection de masque facial il a a atteint une Précision dans la validation de 100% pour les personne sans masque et 99% pour les personnes avec masques. Et

la deuxième catégorie la détection de température à l'aide d'un capteur de température MLX90614 est connecté à la broche GPIO du RPi et le code approprié est écrit pour le capteur.

Base de données utilisée

Cette recherche a mené ses expérimentations sur un ensemble de données [6] obtenues sur internet conçu par Prajna Bhandary qui contient 1376 photos réparties en deux classes : 690 photos avec masques et 686 photos sans masques.

9.6 Article 06 [7]

- (an) **Face mask detection using deep learning : an approach to reduce the risk of spread of coronavirus**
- (fr) **Détection de masque facial à l'aide de l'apprentissage en profondeur : une approche pour réduire le risque de propagation du coronavirus**

Résumé

Dans cet article ils proposent un modèle ResNet50 sur lequel ils ont appliqué le concept d'apprentissage par transfert pour fusionner des informations sémantiques de plus, ils ont proposé une transformation de boîte englobante basé sur le benchmark de reconnaissance d'objets donné dans [7]. Selon ce benchmark, toutes les tâches liées à un problème de reconnaissance d'objets peuvent être regroupées sous trois composants principaux : la colonne vertébrale, le cou et la tête la technique proposée atteint une précision élevée 98,92%. Par ailleurs, le modèle proposé a donné des taux de détection de masque élevé par rapport au récent modèle de référence public publié sous le nom de détecteur RetinaFaceMask (Réseau mobile et ResNet) ils ont atteint une grande précision de 98%. Les performances exceptionnelles du modèle proposé sont parfaitement adaptées aux appareils de vidéosurveillance.

Base de données utilisé :

Cette recherche a mené ses expérimentations dans l'ensemble de données MAFA [7].

9.7 Article 07 [8]

- (an) **Masked Face Recognition Using Deep Learning : A Review**
- (fr) **Reconnaissance des visages masqués à l'aide de l'apprentissage en profondeur : un examen**

Résumé

Dans cette article l'objectif est d'examiner et de discuter des techniques d'apprentissage en profondeur utilisées dans le domaine de Reconnaissance des visages masqués MFR.

Un grand nombre de modèles intelligents de reconnaissance faciale masquée (MFR) ont été récemment présentés et appliqués dans divers domaines. Cette étude approfondie considère une grande variété d'approches et de réalisations récentes, visant à façonner une vue globale du domaine de la MFR. De nombreux défis et axes de recherche prometteurs sont mis en évidence. Le tableau 2.1 illustre l'ensemble des approches MFR étudiées dans cet article.

Modèle	Méthode	Exigences	Base ed données
GAN	Carte et modules d'édition	VGG-19	CelebA
CNN pré-formés	etude comparative sur les CNN	VGGface, Face-cenet, Face ouverte, Face profonde	RMFRD.
MTArcFace	Combinaison de perte ArcFace et l'utilisation du masque perte de classement	ArcFace	LFW, CFP, Argedb
VGG-16, Alex-Net, ResNet-50	Caractéristique profondes des zones du visage	VGG-16, Alex-Net, ResNet-50	RMFRD, SMFRD
VGG-16 et Face-Net	Apprentissage de la distance en corsinus	-	Ensemble de données collectées

ResNet-50, MobileFaceNet	Intégration modèle de démasquage	FCNN	MS1MV2, MFR, MRF2
FaceMaskNet-21	Apprentissage métrique approfondi	FaceMaskNet	–
Basé sur l'attention	Face-yeux multi-granularité	-	MFDD, RMFRD
MasqueLeVisage	MaskTheFace avec FaceNet	FaceNet	VGGface2-mini-SM, LFW-SM
WearMask3D	Perte softmax normalisé	ResNet	MFR2, MFW-mini
ResNet-50	Adaptation de domaine	-	RMFRD
MTCNN	CNN multitache en cascade	FaceNet	MFD
GAN	JAMGAN avec DCR	-	MFSR, CASIA-WebFace, VGGface2
GAN	Distillation de descocclusion	-	Celeb-A, LFW, AR
CNN à deux succursales	Détection de partie latente	ResNet-50	MFV, IMF, LFW
MTCNN	Réidentification des piétons	FaceNet	Images de piétons
Réseaux siamois	Hétérogène formation semi-siamois	ResNet-50	Oulu-CACIA NIR-VIS, BUAA-VisNir
ResNet	Attention-sage	-	RMFRD, SMFFRD
CBAM	Recadrage du visage	CBAM	Webface, AR, Yela B, LFW

TABLE 1.4 – Illustration des approches MFR les plus récentes

9.8 Article 8 [9]

- (an) **Application of deep learning and machine learning models to detect COVID-19 Face Masks - A Review**
- (fr) **Application de modèles d'apprentissage en profondeur et d'apprentissage automatique pour détecter les masques faciaux COVID-19 - Un examen**

Résumé

Dans cet article les auteurs ont mené une étude de comparaison pour toutes les données pertinentes des articles sélectionnés ont été extraites et synthétisées en réponse aux questions de recherche. Après la sélection des articles pertinents, les auteurs ont évalué les résumés des articles de manière indépendante, et 116 articles ont été considérés pour l'éligibilité à la revue. ils ont en outre évalué l'éligibilité des articles en texte intégral et 45 articles ont été supprimés. Seuls 32 articles ont été pris en compte dans cette étude. Les modèles d'intelligence artificielle appliqués pour détecter le masque facial dans les 32 articles.

9.9 Article 09 [10]

- (an) **Fight against COVID-19 : a new deep learning model based on YOLO-v2 with ResNet-50 for detection of medical masks**
- (fr) **Lutte contre COVID-19 : un nouveau modèle d'apprentissage en profondeur basé sur YOLO-v2 avec ResNet-50 pour la détection de masques médicaux**

Résumé

Dans cet article un modèle d'apprentissage en profondeur basé sur YOLO- v2 avec ResNet-50 pour la détection de masques médicaux.

L'objectif de cet article est d'annoter et de localiser le masque facial médical dans des images réelles, à l'aide de deux composants le premier composant ResNet-50 utilisé comme modèle de transfert en profondeur pour l'extraction de caractéristiques, Alors que le

deuxième composant est conçu pour la détection de masques médicaux basés sur YOLO-v2. le modele propose a obtenu une précision moyenne de 81%.

Base de données utilisée

Cette recherche a mené ses expérimentations sur deux ensembles de données ont été combinés en un seul ensemble de données. MMD [10] se compose de 682 images avec plus de 3 000 visages masqués médicaux, et FMD [10] se compose de 853 images l'ensemble de données a été partagé en (70% d'entraînement, 10% validation et 20% test).

9.10 Article 10 [11]

- (an) **New face mask detection technique using machine learning to control the COVID'19 pandemic**
- (fr) **Nouvelle technique de détection de masque facial utilisant l'apprentissage automatique pour contrôler la pandémie de COVID'19**

Résumé

Cet article propose un détecteur de masque qui utilise un système de catégorisation faciale d'apprentissage automatique pour déterminer si une personne porte un masque ou non. Dans cette recherche, les auteurs ont utilisé un CNN C2D profond, ainsi que des données entières sont projetées dans un nouvel espace vectoriel, puis les vecteurs avec une variance individuel le maximale et le reste des caractéristiques sont supprimés pour réduire les dimensions. L'approches utilisés, a atteint une précision de 99%.

Base de données utilisée

Cette recherche a mené ses expérimentations sur l'ensemble de données de visage masqué du monde réel (RMFD) [2] et un autre ensemble de données qui attributs des visages de célébrités (CelebA) [11].

10 Conclusion

Depuis le début de la pandémie du corona virus un grand intérêt à été donné au port de masque dans les lieux publics et fermés pour se protéger du virus et afin de limiter la propagation de la maladie.

la communauté scientifique a aussi orienté ses recherches sur les solutions possibles pour la limitation de la contagion à travers la création de système intelligents de détection du port de masques dans les lieux publics.

Chapitre 2

Apprentissage Profond et Réseaux de Neurones Convolutifs

1 Introduction

Une personne peut distinguer automatiquement beaucoup de choses, et cela du à la vitesse de collecte et de déduction des informations. Ces caractéristiques chez l'homme ont inspiré les scientifiques à créer un modèle similaire au réseau neuronal humain et à l'utiliser dans divers domaines. Les réseaux neuronaux ont souvent été appliqués pour traiter des problèmes de vision par ordinateur et de traitement d'image tels que la détection d'objet, la classification et la reconnaissance de formes d'une façon générales. Parmi les points forts de ces méthodes leur résistance au bruit et leur capacité à traiter des données de grande dimension.

2 Relation entre IA/ML/DL

L'Intelligence Artificielle (IA) est basée sur une démarche d'apprentissage afin de reproduire une partie de l'intelligence humaine à travers une application, un système ou un processus. La reconnaissance faciale, la perception visuelle et autres sont des exemples de systèmes d'intelligence artificielle.

La machine Learning (ML) est un sous-domaine de l'IA qui utilise les réseaux neuronaux artificiels (ANN) pour imiter la façon dont les êtres humains prennent des décisions. La

machine Learning permet aux ordinateurs de développer des modèles d'apprentissage par eux-mêmes, sans aucune programmation, à partir de gros ensembles de données.

Le Deep Learning (DL) ou apprentissage profond est un type d'intelligence artificielle dérivé du machine learning (apprentissage automatique) où la machine est capable d'apprendre par elle-même, contrairement à la programmation où elle se contente d'exécuter à la lettre des règles prédéterminées, est l'une des nombreuses approches du machine Learning qui a connu un grand succès dans ces cinq dernières années, la figure 2.1 illustre la relation entre IA, ML et DL.

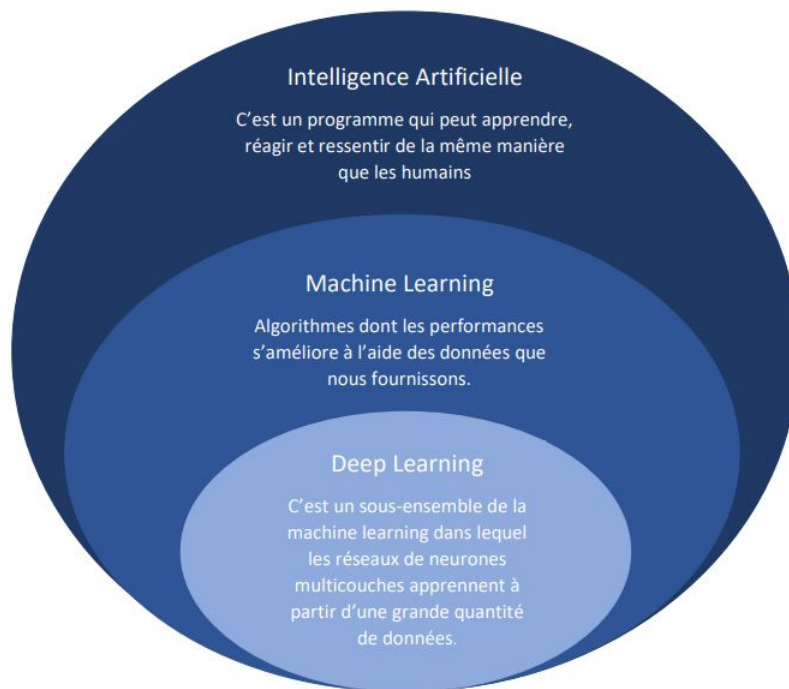


FIGURE 2.1 – Relation entre "IA, ML, et le Deep Learning"

3 Apprentissage automatique (Machine Learning)

Le Machine Learning (ML) a émergé dans la seconde moitié du 20^{ème} siècle du domaine de l'intelligence artificielle et correspond à l'élaboration d'algorithmes capables d'accumuler de la connaissance et de l'intelligence à partir d'expériences, sans être guidés au cours de leur apprentissage, ni explicitement programmés pour gérer telle ou telle expérience ou donnée spécifique [w7].

L'apprentissage automatique est un sous-ensemble de l'intelligence artificielle globale et le terme général désignant le moment où les ordinateurs apprennent à partir des données.

Il décrit l'intersection de l'informatique et des statistiques où les algorithmes sont utilisés pour effectuer une tâche spécifique sans être explicitement programmés ; au lieu de cela, ils reconnaissent les modèles dans les données et font des prédictions une fois que de nouvelles données arrivent. Finalement, c'est le mode d'apprentissage qui est le plus souvent utilisé pour catégoriser l'apprentissage automatique en trois grandes catégories [w7] et illustre dans la figure 2.2 :

- L'apprentissage supervisé.
- L'apprentissage non-supervisé.
- L'apprentissage par renforcement.

3.1 Apprentissage supervisé

L'apprentissage supervisé utilise une relation connue entre l'entrée et la sortie. C'est là que les données étiquetées entrent en jeu : le but de l'algorithme est d'apprendre à partir des "réponses correctes" dans les données de formation et d'utiliser les informations pour faire des prédictions lorsqu'on leur donne de nouvelles données d'entrée [w7].

L'apprentissage supervisé comporte deux tâches principales :

- Régression : Prédire une valeur numérique continue (combien ?).
- Classification : attribuez une étiquette (quelle classe?).

3.2 Apprentissage non-supervisé

Alors que l'apprentissage supervisé nécessite des paires d'entrée-sortie (ou des données étiquetées) pour apprendre, les algorithmes d'apprentissage non supervisé utilisent uniquement des données d'entrée (données non étiquetées). Bien qu'un peu moins intuitive, cette méthode d'apprentissage convient aux problèmes où nous avons peu ou pas d'idée à quoi devraient ressembler nos résultats. L'objectif est d'acquérir des connaissances et de trouver une structure dans les données [w7].

3.3 Apprentissage par-renforcement

Dans l'apprentissage par renforcement, l'algorithme (dans ce contexte également souvent appelé agent) apprend par essais et erreurs en utilisant la rétroaction sur ses propres

actions. Les récompenses et les punitions fonctionnent comme des signaux pour un comportement désiré et non désiré.

Ce type d'apprentissage est souvent utilisé dans le cadre de la robotique, de la théorie des jeux et des véhicules autonomes [w7].

4 Avantages de l'apprentissage automatique

Parmi les avantages de l'apprentissage automatique :

- Identification des tendances et des modèles.
- Amélioration continue.
- Gérer la grande quantité de données multidimensionnelles.
- Large application

5 Inconvénients de l'apprentissage automatique

Parmi les inconvénients de la Machine Learning :

- L'acquisition des données.
- Temps et ressources.
- Interprétation.
- Sensibilité élevée aux erreurs.

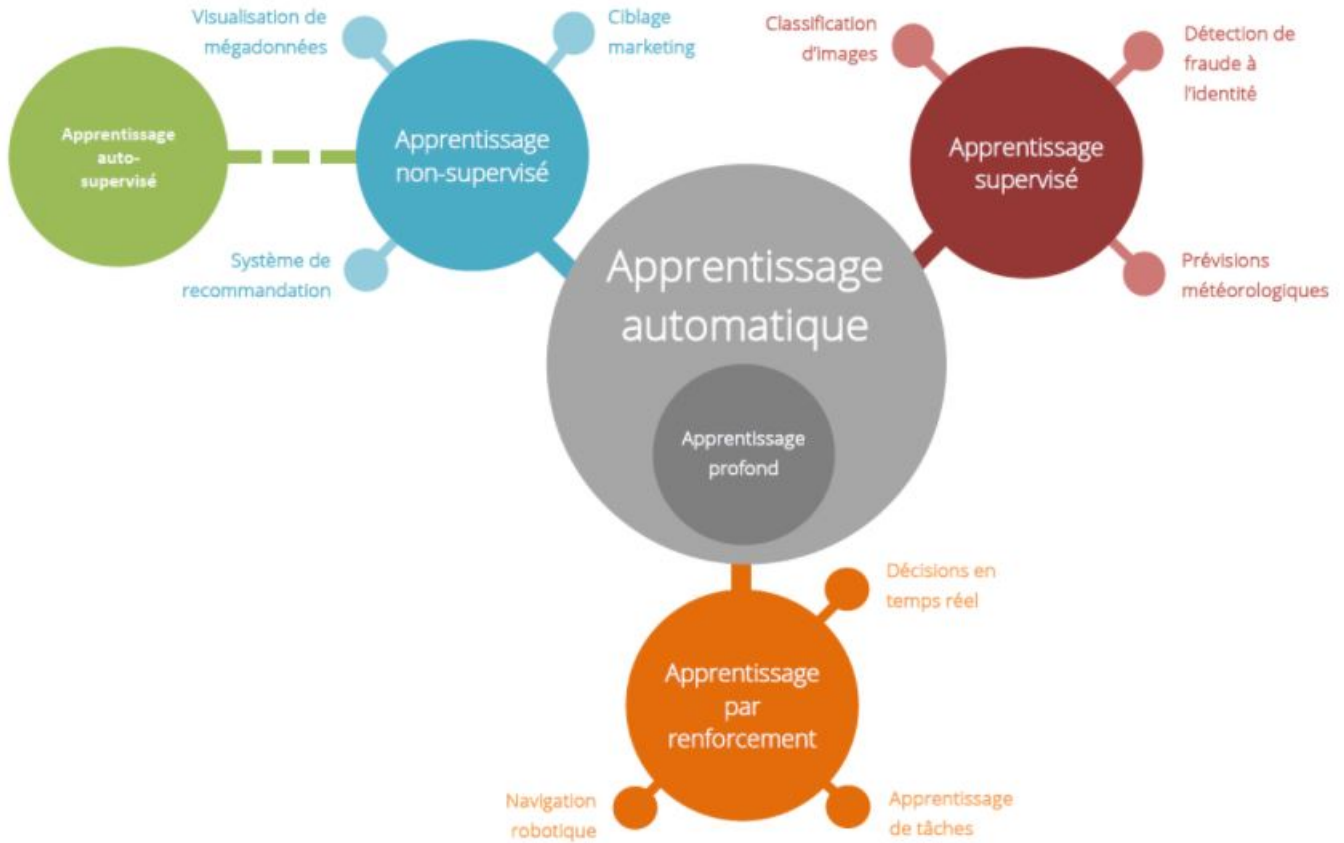


FIGURE 2.2 – Algorithmes d'apprentissage automatique [w8]

5.1 Neurone biologique

Les réseaux de neurones artificiels sont des modèles mathématiques inspirés de la biologie. L'élément de base de ces réseaux, le neurone artificiel, était issu au départ d'une volonté de modélisation du fonctionnement d'un neurone biologique. La figure 2.3 illustre un neurone biologique [w9]. Schématiquement, on peut décomposer le neurone en 3 grandes entités :

- un corps cellulaire, appelé péricaryon ;
- un ensemble de dendrites (de l'ordre de 7 000) ;
- un axone.

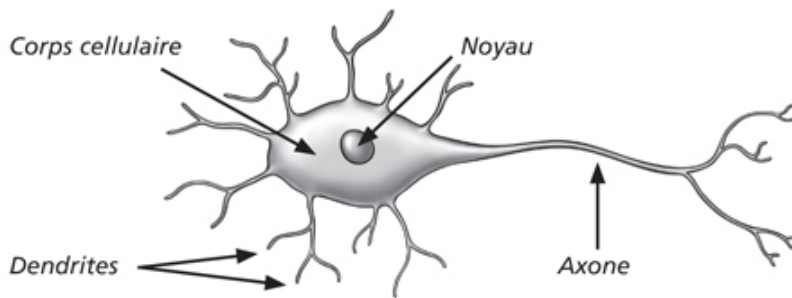


FIGURE 2.3 – Neurone biologique

5.2 Neurone formel

Le perceptron, encore appelé neurone artificiel ou neurone formel, cherche à reproduire le fonctionnement d'un neurone biologique. Il existe différents niveaux d'abstraction, suivant la précision de la modélisation voulue. Dans ce cas, nous n'allons pas chercher à reproduire exactement et exhaustivement tous les processus biologiques en œuvre dans un neurone biologique. Il s'agit d'une version simplifiée [Figure :2.4] [w9]. Nous allons considérer les entités suivantes :

- Des entrées, notées x sous formes de vecteurs, représentant les dendrites ;
- Une sortie, notée y , représentant l'axone ;
- Des paramètres, notés w et b , influençant le fonctionnement du neurone.

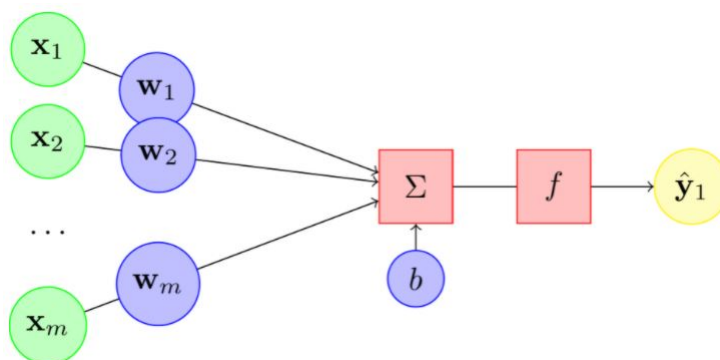


FIGURE 2.4 – Neurone formel

6 Apprentissage en profondeur (Deep Learning)

6.1 Historique

Depuis 2012, les algorithmes à base de deep learning (apprentissage profond) ont participé à la résolution de beaucoup de problèmes : reconnaître des visages (DeepFace), la conduite de voitures autonomes ou encore la recherche de cellules cancéreuses. Pourtant, les fondements de ces méthodes ne sont pas si récents : le deep learning a été formalisé en 2007 à partir des nouvelles architectures de réseaux de neurones dont les précurseurs sont McCulloch et Pitts en 1943. Suivant de nombreux développements comme le perceptron, les réseaux de neurones convolutifs de Yann Le Cun et Yoshua Bengio en 1998 et les réseaux de neurones profonds qui en découlent en 2012 et ouvrent la voie à de nombreux champs d'application comme la vision, le traitement du langage ou la reconnaissance de la parole [13].

6.2 Définition

Le Deep Learning (ou apprentissage profond) est un ensemble de méthodes d'apprentissage automatique conçues sur la base de réseaux de neurones profonds, visant à mimer la profondeur des couches d'un cerveau : le cerveau humain est profond, dans le sens où chaque action est le résultat d'une longue chaîne de communications synaptiques avec de nombreuses couches de traitement. Le deep learning réunit une classe d'algorithmes d'apprentissage correspondants à ces architectures profondes. Il est souvent utilisé pour un apprentissage «de bout en bout», c'est à dire l'apprentissage simultané des caractéristiques utiles des données, et de la meilleure façon de les utiliser.

Parmi les exemples récents les plus remarquables, citons le programme de reconnaissance des visages de Facebook, la victoire d'AlphaGo ou encore les outils d'aide à la conduite assistée et autonome (ADAS), la santé avec la recherche de cellules cancéreuses par la start-up DreamQuark, ou la reconnaissance de parole [w10].

7 Machine Learning VS Deep Learning

L'apprentissage automatique signifie que les ordinateurs apprennent à partir de données à l'aide d'algorithmes pour effectuer une tâche sans être explicitement programmés. L'apprentissage en profondeur utilise une structure complexe d'algorithmes modélisés sur le cerveau humain. Cela permet le traitement de données non structurées telles que des documents, des images et du texte. Bien qu'il existe de nombreuses différences entre ces deux sous-ensembles d'intelligence artificielle, voici quatre des plus importantes : Intervention humaine l'apprentissage automatique nécessite une intervention humaine plus continue pour obtenir des résultats. Le deep learning est plus complexe à mettre en place mais nécessite une intervention minimale par la suite. Matériel les programmes d'apprentissage automatique ont tendance à être moins complexes que les algorithmes d'apprentissage en profondeur et peuvent souvent fonctionner sur des ordinateurs conventionnels, mais les systèmes d'apprentissage en profondeur nécessitent un matériel et des ressources beaucoup plus puissants. Cette demande de puissance a entraîné une utilisation accrue des unités de traitement graphique. Les GPU sont utiles pour leur mémoire à bande passante élevée et leur capacité à masquer la latence (retards) dans le transfert de mémoire en raison du parallélisme des threads (la capacité de nombreuses opérations à s'exécuter efficacement en même temps). fois les systèmes d'apprentissage automatique peuvent être mis en place et fonctionner rapidement, mais peuvent être limités dans la puissance de leurs résultats. Les systèmes d'apprentissage en profondeur prennent plus de temps à mettre en place, mais peuvent générer des résultats instantanément (bien que la qualité soit susceptible de s'améliorer avec le temps à mesure que davantage de données deviennent disponibles). Approche l'apprentissage automatique a tendance à nécessiter des données structurées et utilise des algorithmes traditionnels comme la régression linéaire. L'apprentissage en profondeur utilise des réseaux de neurones et est conçu pour accueillir de gros volumes de données non structurées [w11].

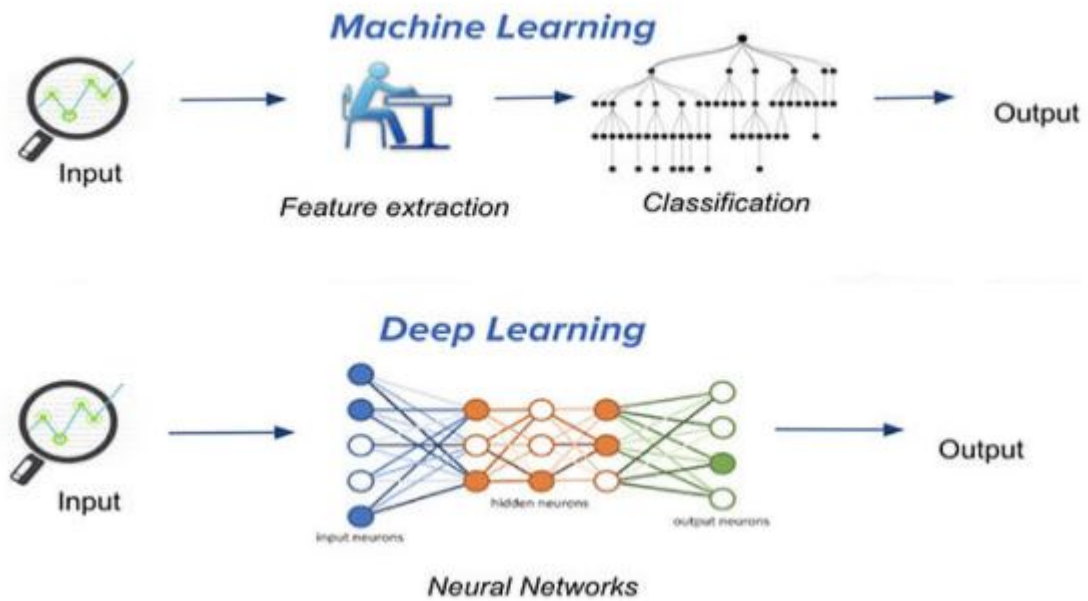


FIGURE 2.5 – Machine Learning VS Deep Learning[w12]

8 Avantages du Deep Learning

Ils existent plusieurs avantages qui distinguent le Deep Learning, on cite :

- La robustesse pour comprendre et utiliser de nouvelles données.
- Il obtient ses résultats plus rapidement. Il apprend au fil du temps.
- Il peut gérer de grandes quantités de données pour de petits réseaux avec un coût d'apprentissage bien moindre.
- De meilleurs résultats qu'avec d'autres méthodes d'apprentissage machine.
- Le traitement des données non structurées.

9 Inconvénient du Deep Learning

- Nécessite une grande puissance de calcul.
- Une technologie coûteuse à mettre en place.
- Nécessite une vaste base de données.

10 Domaines d'application de Deep Learning

Aujourd'hui, les géants de la technologie investissent massivement dans ce domaine de l'intelligence artificielle et développement de multiples applications du Deep learning.

- La traduction.
- Le Deep Learning au service de la rédaction de textes.
- Le Deep Learning dans le secteur du marketing et de la vente.
- Dans le domaine de l'agriculture.
- Le domaine de l'informatique appliquées aux Nouvelles Technologie d'Information et de Communication (reconnaissance visuelle - par exemple d'un panneau de signalisation par un robot ou une voiture autonome - et vocale notamment).
- La robotique.
- Reconnaissance ou comparaison de formes.
- La sécurité.
- La santé.

11 Différentes Architectures du Deep Learning

Bien qu'il existe un grand nombre de variantes d'architectures profondes. Il n'est pas toujours possible de comparer les performances de toutes les architectures, car elles ne sont pas toutes évaluées sur les mêmes ensembles de données. Le Deep Learning est un domaine à croissance rapide, et de nouvelles architectures, variantes ou algorithmes apparaissent toutes les semaines.

11.1 Réseaux neuronaux convolutifs (CNN)

Les réseaux convolutifs sont une forme particulière de réseaux neuronaux multicouches (aussi appelés CNN) sont caractérisés par leurs premières couches convolutionnelles. Une couche convolutive, est basée sur le principe mathématique de convolution, et cherche à repérer la présence d'un motif (dans une image le plus souvent, ou un signal). Les réseaux neuronaux convolutifs ont de nombreuses applications dans la reconnaissance d'images, de vidéos ou le traitement du langage naturel[11].

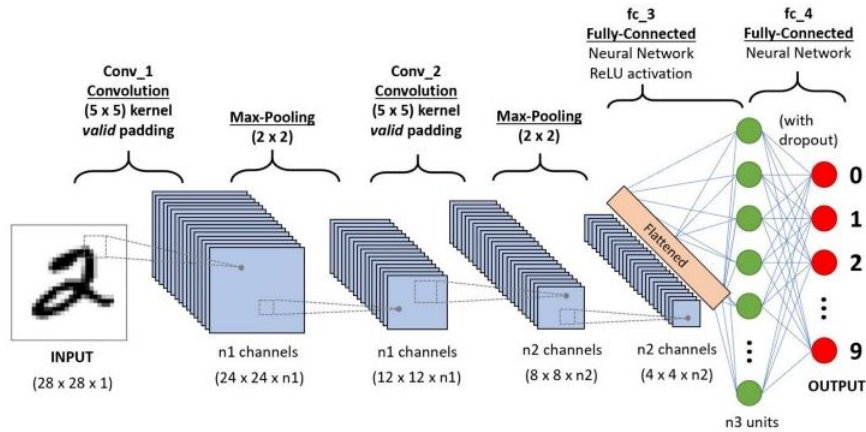


FIGURE 2.6 – Exemple de réseau convolutif (CNN)

11.2 Réseau de neurones récurrents (RNN)

Les réseaux récurrents (ou RNN pour Recurrent Neural Networks) sont des réseaux de neurones dans lesquels l'information peut se propager dans les deux sens, y compris des couches profondes aux premières couches. En cela, ils sont plus proches du vrai fonctionnement du système nerveux. Pour cette raison, les RNNs sont particulièrement adaptés aux applications faisant intervenir le contexte, et plus particulièrement au traitement des séquences temporelles comme l'apprentissage et la génération de signaux, c'est à dire quand les données forment une suite et ne sont pas indépendantes les unes des autres. Néanmoins, pour les applications faisant intervenir de longs écarts temporels, cette « mémoire à court-terme » n'est pas suffisante. En effet, les RNNs « classiques » ne sont capables de mémoriser que le passé dit proche, et commencent à « oublier » au bout d'une cinquantaine d'itérations environ. Le schéma Figure(2.7) montre le détail d'une couche récurrente. où $[X_i^t, X_j^t]$ désignent respectivement les entrées et les sorties de la couche à l'instant t, la présence de poids $[W_{ji}]$ reliant les entrées à la sortie (comme pour un MLP classique). Les connexions récurrentes sont notées en rouge [14].

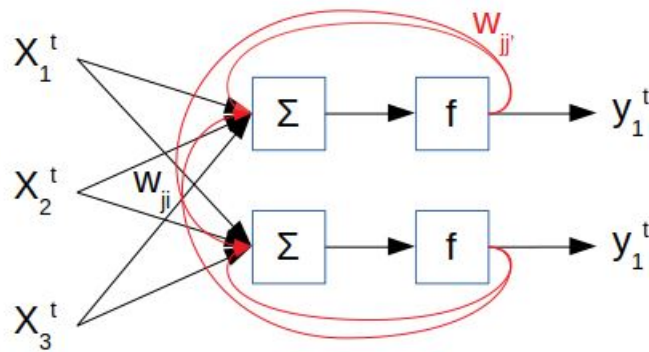


FIGURE 2.7 – Exemple d’une couche RNN simple à trois entrée et deux sorties.

11.3 Modèle génératif

Si les modèles discriminatifs comme (CNN, RNN) sont utilisés pour prédire les données du label et de l’entrée, tant dis que le modèle génératif, quant à lui, a une tâche beaucoup plus complexe à accomplir. Il doit comprendre la distribution à partir de laquelle les données sont obtenues et doit ensuite utiliser cette compréhension pour effectuer la tâche de classification. Voici quelques exemples de modèle génératif [8] :

- Boltzmann Machines.
- Restricted Boltzmann Machines.
- Deep Belief.
- Generative Adversarial Networks (GAN).

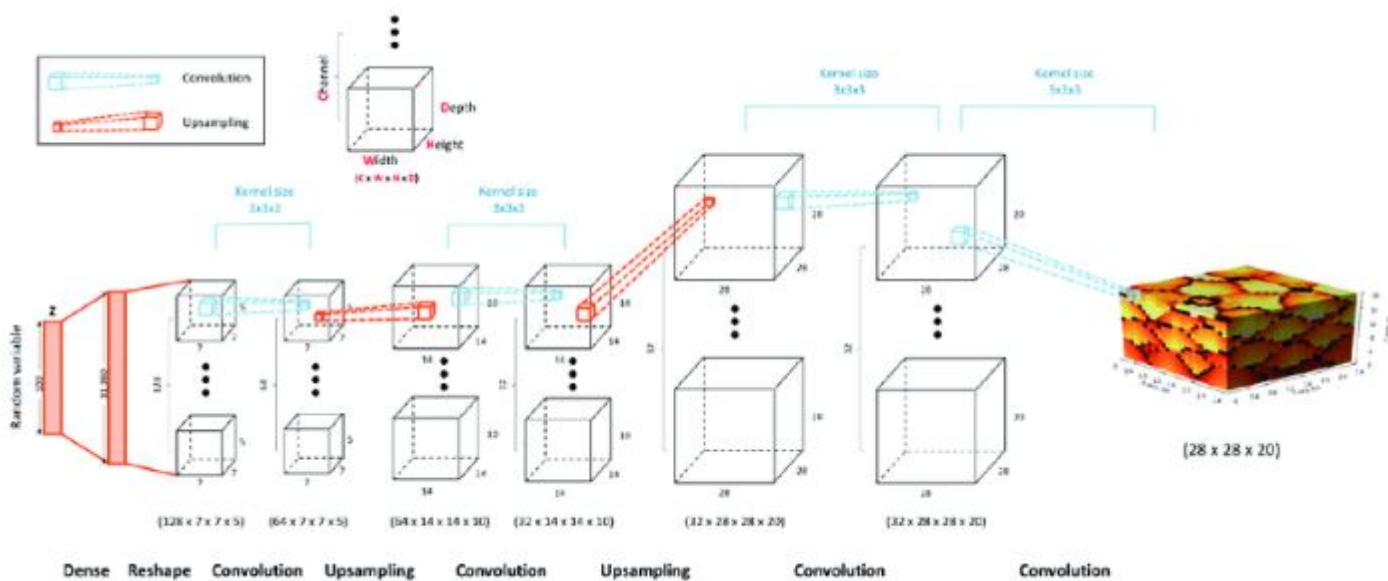


FIGURE 2.8 – Exemple de modèle génératif (CNN)

12 Structure des réseaux de neurones convolutifs

Nous donnons un intérêt spécifique à ce type de réseau vu le contexte de notre application

12.1 Architecture des réseaux de neurones convolutifs

Les CNN sont composés de trois types de couches : des couches convolutives et des couches entièrement connectées. Lorsque ces couches sont empilées, une architecture CNN a été formée. Ces couches sont traitées indépendamment [16] :

- La couche de convolution (CONV), qui traite les données d'un champ récepteur.
- La couche de pooling (POOL), qui permet de compresser l'information en réduisant la taille de l'image intermédiaire (souvent par sous-échantillonnage).
- La fonction d'activation (Relu), souvent appelée par abus 'Relu' en référence à la fonction d'activation (Unité de rectification linéaire).
- La couche "entièrement connectée" (FC), qui est une couche de type perceptron.
- La couche de perte (LOSS).

12.2 La couche de convolution

La convolution est le cœur du réseau de neurones convolutif. À l'origine, une convolution est un outil mathématique (on parle de produit de convolution) très utilisé en retouche d'image, car il permet d'en faire ressortir l'extraction des caractéristiques à partir des images d'entrées, afin d'appliquer un bon filtre. En fait, une convolution prend simplement en entrée une image et un filtre (qui est une autre image), effectue un calcul, puis renvoie une nouvelle image (généralement plus petite) [15].

Trois hyper paramètres permettent de dimensionner le volume de la couche de convolution (aussi appelé volume de sortie) : la profondeur, le pas et la marge.

- Profondeur de la couche : nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).
- Le pas : contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.

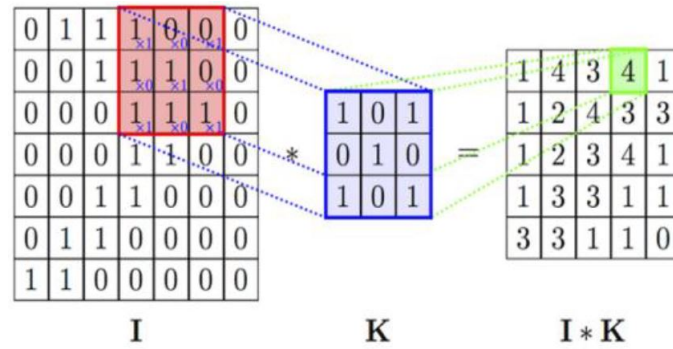


FIGURE 2.9 – Exemple d’une convolution

- La marge (à 0) ou zéro padding : parfois, il est commode de mettre des zéros à la frontière du volume d’entrée. La taille de ce ‘zéro-padding’ est le troisième hyper paramètre. Cette marge permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d’entrée.

Des filtres sont appliqués à chaque image utilisée pour l’apprentissage à différentes résolutions, et la sortie de chaque image convoluée est utilisée comme entrée de la couche suivante.

Types de convolutions

Il existe plusieurs types de convolutions, même si en général on utilise celle de base, il peut s’avérer utile de connaître les outils à notre disposition [14].

- La convolution classique : qui représente le décalage du noyau entre chaque calcul, et le padding qui est la manière dont on peut « dépasser » de l’image pour appliquer la convolution.
- La convolution dilatée : identique à la convolution à ceci près que le noyau est élargi (on prend, par exemple, un pixel sur deux pour calculer la convolution). Il y a un paramètre supplémentaire : la dilation rate, qui est le nombre de pixels à ignorer.
- Convolution transposée : qui construit la sortie comme si on inversait une convolution sur l’image.

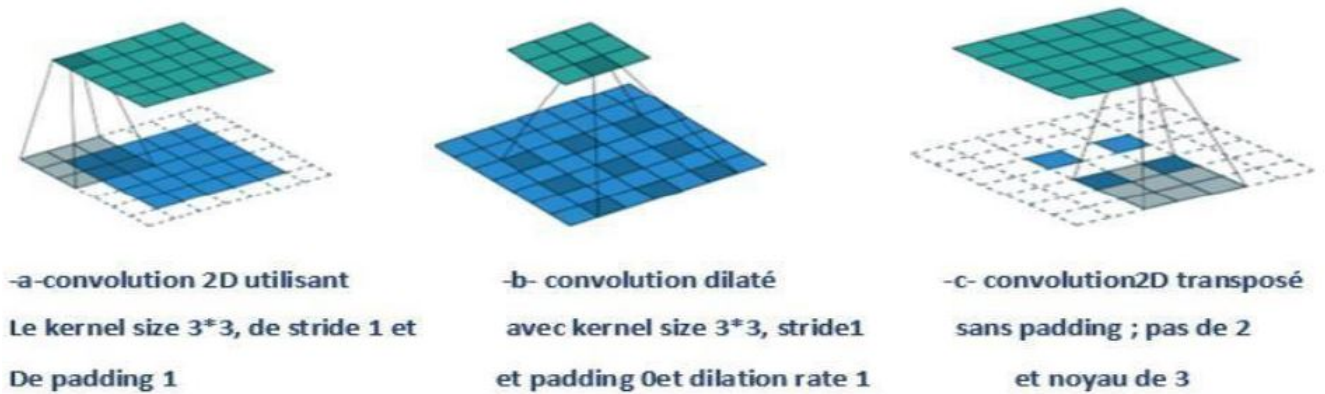


FIGURE 2.10 – Diffèrent type de convolution

12.3 Fonctions d’activations

La sortie de chaque couche convolutionnelle est envoyée à une couche de fonction d’activation. La couche de fonction d’activation se compose d’une fonction d’activation qui prend la carte de caractéristiques produite par la couche de convolution et génère la carte d’activation comme sortie, elle est utilisée pour transformer le niveau d’activation d’un neurone en un signal de sortie. En général, 3 propriétés sont attendues de la fonction d’activation [w9] :

- Non linéarité : c’est la propriété cruciale de la fonction d’activation. Grâce à cette propriété le réseau de neurones peut être utilisé pour résoudre des problèmes non linéaires ;
- Différentiable - ce qui signifie que nous avons une dérivée continue du premier ordre. C’est une propriété souhaitable pour activer des méthodes d’optimisation basées sur un gradient comme la rétropropagation Rumelhart et al. (1986) [w9] ;
- Monotone Cette caractéristique aide le réseau de neurones à converger plus facilement vers un modèle plus précis. De nombreuses fonctions d’activation sont utilisées comme l’indique. La figure(2.11) illustre quelques fonctions d’activation [w13].

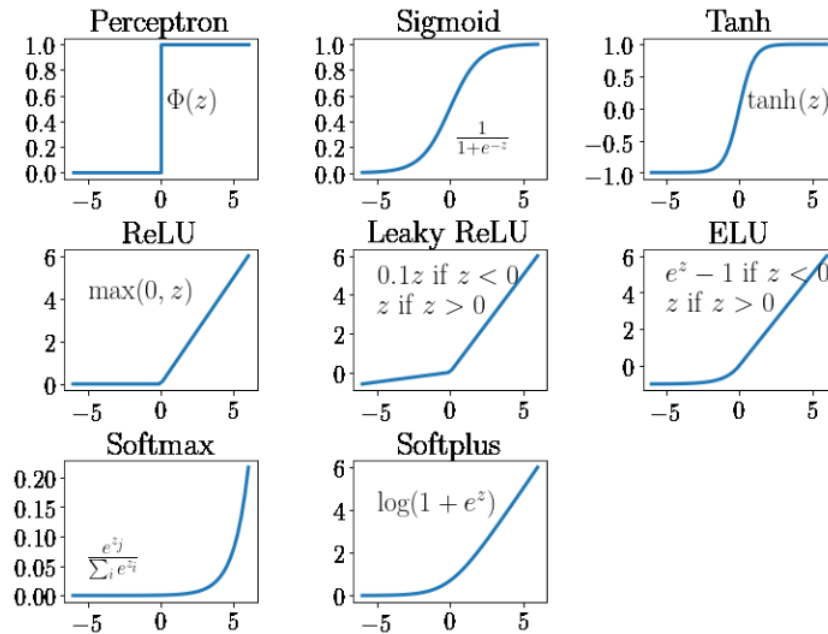


FIGURE 2.11 – Quelques fonctions d’activation.

Remarque

En pratique, on utilise le ReLU comme activation pour les couches cachées car cette dernière accélère le processus d’entraînement.

La couche ReLU

Rectified Linear Unit (ReLU) : Elle est utilisée après chaque opération de convolution, ou toutes les valeurs de pixels négatifs sont mises à zéro. Le but de ReLU est d’introduire la non-linéarité dans notre CNN, puisque la plupart des données du monde réel, et les caractéristiques appliquées à l’une des cartes d’entrée donne une carte de sortie qui est également appelée carte de caractéristiques rectifiées [15].

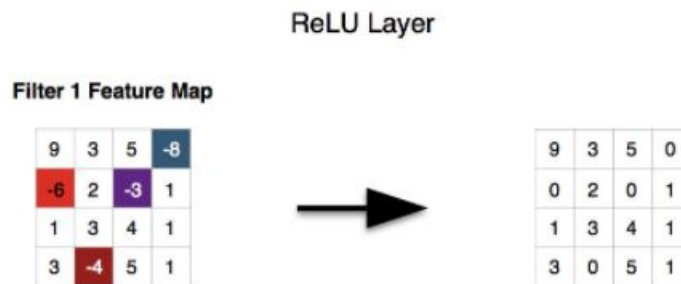


FIGURE 2.12 – Application de ReLU à partir d’une matrice.

12.4 Couche de Pooling

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs feature maps, et applique à chacune d'entre elles l'opération de pooling. L'opération de pooling (ousub-sampling) consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes. Pour cela en pratique, on utilise souvent des cellules carrées de petite taille (2 x 2 pixels ou 3 x 3 pixels) pour ne pas perdre trop d'informations. La couche de pooling permet de réduire le nombre de paramètres et de calculs dans le réseau. On améliore ainsi l'efficacité du réseau et on évite le sur-apprentissage [14]. Il existe plusieurs types de pooling : Max, Moyenne, Somme, ...etc.

- Max pooling : qui revient à prendre la valeur maximale de la sélection. C'est le type le plus utilisé, car il est rapide à calculer (immédiat), et permet de simplifier efficacement l'image.
- Le min pooling : qui revient à prendre la valeur minimale de la sélection.
- Le mean pooling : (ou average pooling), soit la moyenne des pixels de la sélection : on calcul la somme de toutes les valeurs et on divise par le nombre de valeurs. On obtient ainsi une valeur intermédiaire pour représenter ce lot de pixels.



FIGURE 2.13 – Exemple des trois type de pooling cité.

Mise à plat (Flattening)

La dernière étape de la partie “extraction des informations”, le flattening consiste simplement à mettre bout à bout toutes les images (matrices) que nous avons (résultat de pooling) pour en faire un (long) vecteur. Les pixels (en réalité ce ne sont plus des images ou des pixels, mais des matrices de nombres, donc les pixels sont ces nombres) sont récupérés ligne par ligne et ajoutés au vecteur final [14].

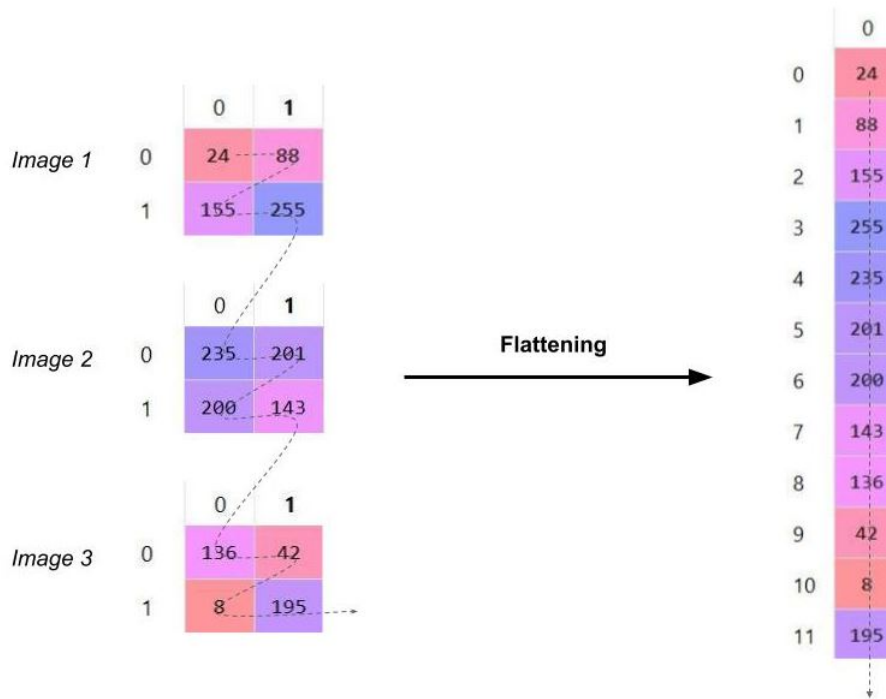


FIGURE 2.14 – Exemple de mise à plat des images finales en sortie

12.5 Couche entièrement connectée (FC :Fully Connected)

Le but de la couche entièrement connectée est de pouvoir utiliser ces fonctions pour classer l'image d'entrée dans différentes classes en fonction de l'ensemble de données d'apprentissage [16].

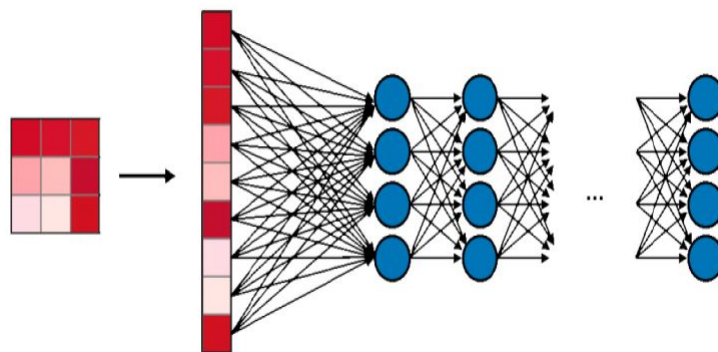


FIGURE 2.15 – Exemple d'opération entièrement connecte (FC)

12.6 La couche de perte (LOSS)

La couche de perte spécifie comment l'entraînement du réseau pénalise l'écart entre le signal prévu et réel. Elle est normalement la dernière couche dans le réseau. Diverses fonctions adaptées à différentes tâches peuvent y être utilisées. La perte « Soft max » est utilisée pour prédire une seule classe parmi K classes mutuellement exclusives. La perte par entropie croisée sigmoïde est utilisée pour prédire K valeurs de probabilité indépendante dans [0,1]. La perte euclidienne est utilisée pour régresser vers des valeurs réelles [14].

13 Quelques réseaux convolutifs célèbres

13.1 LeNet-5 [17]

LeNet-5 est l'un des premiers modèles pré-formés proposés par Yann LeCun et al [17] en 1998, et l'une des architectures les plus simples. Il a 2 couches convolutives et 3 couches entièrement connectées (d'où LeNet-5 il est très courant que les noms des réseaux de neurones soient dérivés du nombre de couches convolutives et entièrement connectées dont ils disposent). La couche de mise en commun moyenne telle que nous la connaissons maintenant s'appelait une couche de sous-échantillonnage et elle avait des poids entraînaibles (ce qui n'est pas la pratique actuelle de conception de CNN de nos jours). La figure 2.16 illustre l'architecture du réseau LeNet-5.

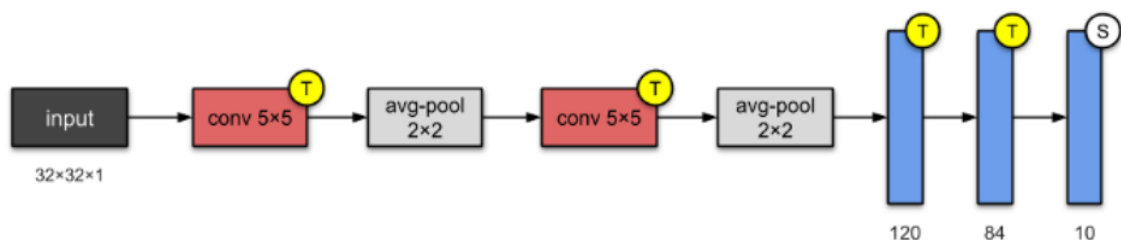


FIGURE 2.16 – Architecture 'LeNet-5'

13.2 AlexNet [18]

AlexNet il à été proposés par Alex Krizhevsky et al [18] en 2012, il contenait huit couches ; les cinq premières étaient des couches convolutives, certaines d'entre elles suivies de

couches de regroupement maximal, et les trois dernières étaient des couches entièrement connectées. Il a utilisé la fonction d'activation ReLU non saturante, qui a montré des performances d'entraînement améliorées par rapport à tanh et sigmoïde. La fonction d'activation utilisée dans la couche de sortie est Softmax. La figure 2.17 illustre l'architecture du réseau AlexNet.

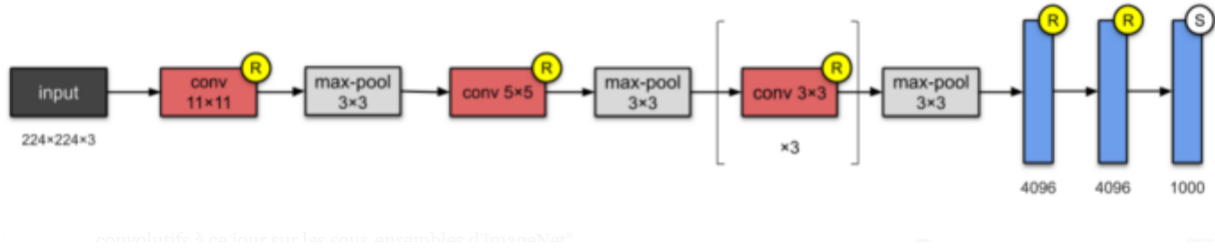


FIGURE 2.17 – Architecture 'AlexNet'

13.3 VGG-16 [19]

VGG-16 il à été proposés par Karen Simonyan et.al [19] en 2014, Les gens de Visual Geometry Group (VGG) ont inventé le VGG-16 qui a 13 couches convolutionnelles et 3 entièrement connectées, emportant avec eux la tradition ReLU d'AlexNet. Ce réseau empile plus de couches sur AlexNet et utilise des filtres de plus petite taille (2×2 et 3×3). La figure 2.18 illustre l'architecture du réseau VGG-16.

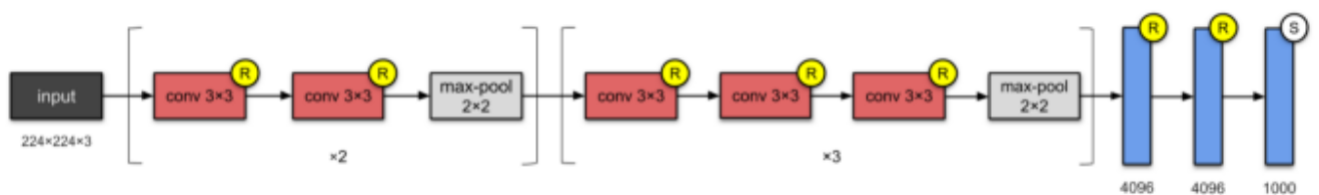


FIGURE 2.18 – Architecture 'VGG-16'

13.4 Inception-v4 [20]

Inception-v4 est une architecture de réseau neuronal convolutif qui s'appuie sur les itérations précédentes de la famille Inception en simplifiant l'architecture et en utilisant plus de modules d'inception qu'Inception-v3. La principale différence est le groupe Stem et

quelques changements mineurs dans le module Inception-C. L'architecture sera détaillée dans le chapitre 3.

13.5 Inception-ResNet-V2 [20]

Inception-ResNet-v2 est une architecture neuronale convolutive qui s'appuie sur la famille d'architectures Inception mais intègre des connexions résiduelles (remplaçant l'étape de concaténation de filtres de l'architecture Inception). La figure 2.19 illustre l'architecture du réseau Inception-ResNet-V2.

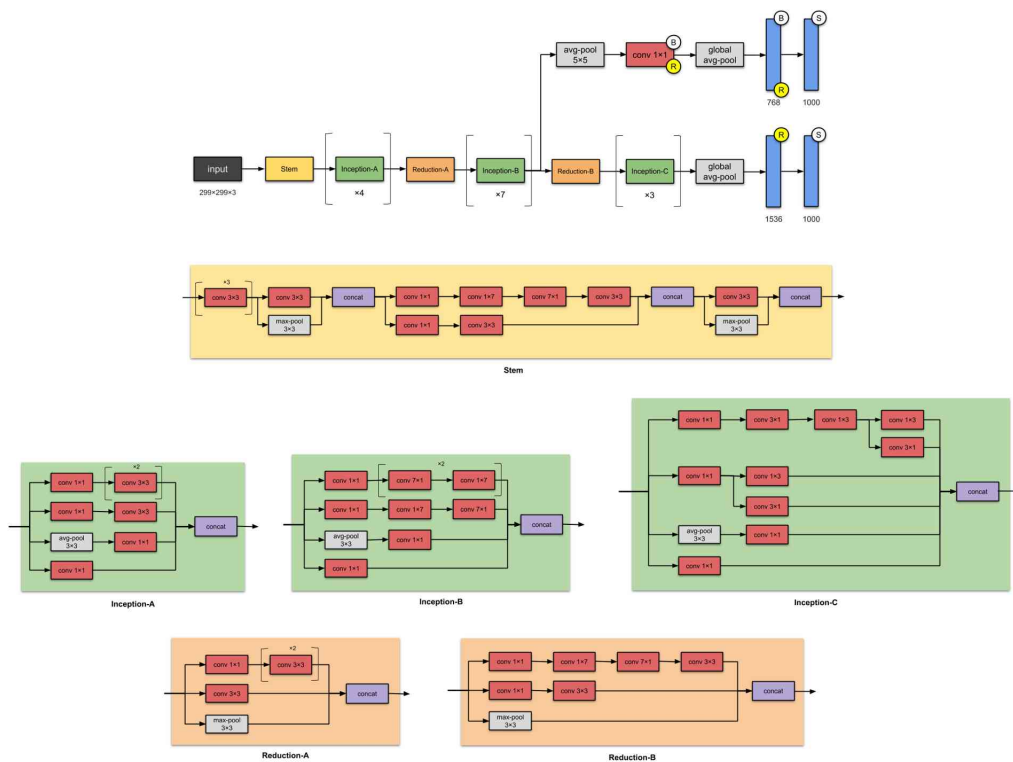


FIGURE 2.19 – Architecture 'Inception-ResNet-V2'

13.6 ResNet-50 [21]

Au cours des dernières CNN, nous n'avons rien vu d'autre qu'un nombre croissant de couches dans la conception et l'obtention de meilleures performances. Mais "avec l'augmentation de la profondeur du réseau, la précision devient saturée (ce qui n'est peut-être pas surprenant) puis se dégrade rapidement". Les gens de Microsoft Research ont résolu

ce problème avec ResNet - en utilisant des connexions de saut (c'est-à-dire des connexions de raccourci, des résidus) tout en créant des modèles plus profonds. La figure 2.20 illustre l'architecture du réseau ResNet-50.

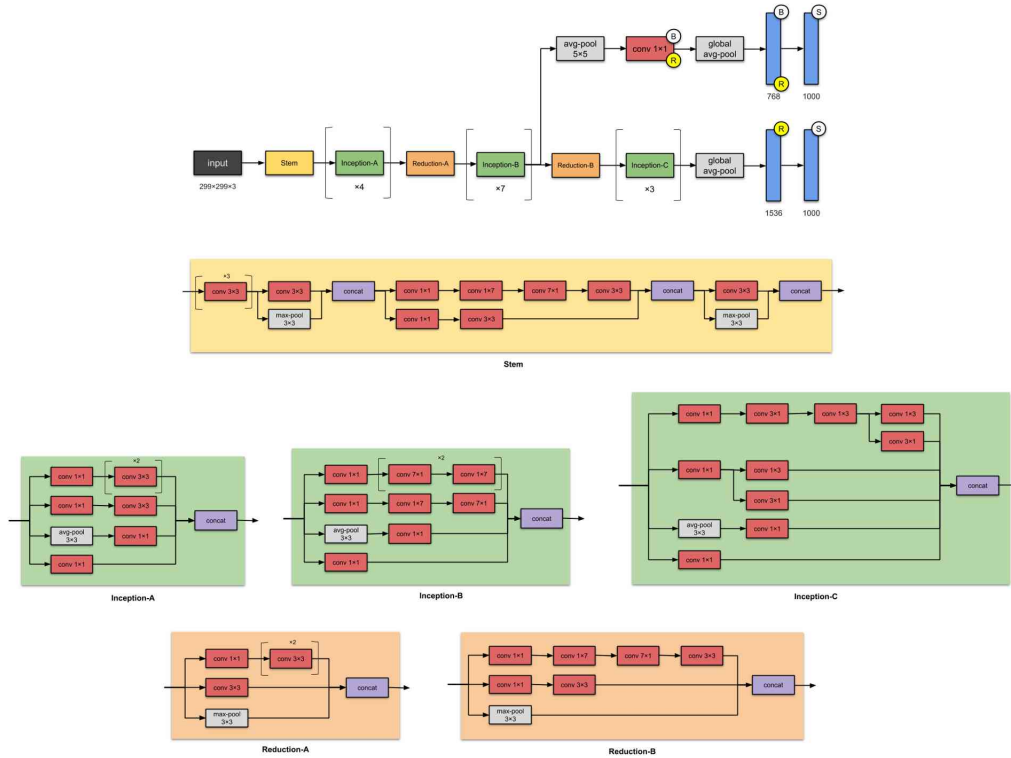


FIGURE 2.20 – Architecture 'ResNet-50'

14 Construction d'un modèle d'apprentissage en profond

Pour construire un modèle d'apprentissage en profondeur il est indispensable de suivre minsiueusement les étapes suivantes :

14.1 Étape 1 : Rassembler l'ensemble de données

La première composante de la construction d'un réseau d'apprentissage en profondeur consiste à rassembler notre ensemble de données initial. Nous avons besoin des images elles-mêmes ainsi que des étiquettes associées à chaque image. Ces étiquettes doivent provenir d'un ensemble fini de catégories, telles que : les catégories (chien, chat, souris).

De plus, le nombre d'images pour chaque catégorie doit être approximativement uniforme (c'est-à-dire le même nombre d'exemples par catégorie). Si nous avons deux fois plus d'images de chats

que d'images de chiens et cinq fois plus d'images de souris que d'images de chats, alors notre classificateur deviendra naturellement biaisé ou sur-ajuster dans ces catégories fortement représentées. Le déséquilibre des classes est un problème courant dans l'apprentissage automatique et il existe un certain nombre de façons de le surmonter [22].

14.2 Étape 2 : Fractionner l'ensemble de données

Divisez l'ensemble de données en deux parties : un ensemble d'entraînement, un ensemble de test. Un ensemble de formation est utilisé par notre classificateur pour « apprendre » à quoi ressemble chaque catégorie en faisant des prédictions sur les données d'entrée, puis se corriger lorsque les prédictions sont fausses. Une fois le classificateur formé, nous pouvons évaluer les performances sur un ensemble de tests. Il est extrêmement important que l'ensemble de formation et l'ensemble de tests soient indépendants l'un de l'autre. Les tailles divisées communes pour les ensembles de formation et de test incluent [(66.633.3%) ; (75%|25%) ; (90%|10%)] respectivement [22].



FIGURE 2.21 – Exemples de fractionnement commun des données de formation et de test. [w19]

14.3 Étape 3 : Former le réseau

Création d'une architecture de modèle en définissant le nombre de nœuds dans chaque couche 'layer', le nombre de couches cachées 'hidden layers', le nombre de nœuds dans la dernière couche 'output layer', le type de fonction d'activation ...etc [22].

14.4 Étape 4 : Évaluation

Faire un test pour chacune des images de l'ensemble de test. Puis tabule les prédictions du modèle pour une image dans l'ensemble de test. Enfin, ces prédictions de modèle sont comparées aux étiquettes de vérité terrain de l'ensemble de test. Les étiquettes de vérité terrain représentent ce qu'est réellement la catégorie d'image. À partir de là, on peut calculer le nombre de prédictions que notre classificateur a obtenu correctement et calculer des rapports agrégés tels que la précision, le rappel et la f-mesure (mesures des performances), qui sont utilisés pour quantifier les performances du réseau dans son ensemble [22].

14.5 Mesures de performances

Les évaluations de réseaux mise en œuvre nécessite le calcul d'un certain nombre de paramètres, on note :

- a- Vrai positif 'True positive TP' : Le modèle prédit correctement la classe positive.
- b- Vrai négatif 'True negative TN' : Le modèle prédit correctement la classe négative
- c- Faux positif 'False positive FP' : Le modèle prédit incorrectement la classe positive.
- d- Faux négatif 'False negative FN' : Le modèle prédit incorrectement la classe négative.

Matrice de confusion

Le Tableau 2.1 illustre la matrice de confusion d'un système de classification binaire :

	Classe réelle positive	Classe réelle négative
Classe prédite positive	Vrai positif (VP)	Faux positif (FP)
Classe prédite négative	Faux négatif (FN)	Vrai négatif (VN)

TABLE 2.1 – Matrice de confusion.

Exactitude ‘Accuracy’ [w13]

Taux de bonnes prédictions.

$$Accuracy = \frac{VP + VN}{VP + VN + FN + FP} \quad (2.1)$$

Taux vrai positif ou Sensitivité ‘Recall’ [w13]

Probabilité d’un cas positif si la prédiction est positive.

$$Taux\ vrai\ positif = \frac{VP}{VP + FP} = \frac{VN}{P} \quad (2.2)$$

Taux vrai négatif ou Spécificité [w13]

Probabilité d’une prédiction négative dans un cas négatif.

$$Taux\ vrai\ negatif = \frac{VN}{VN + FP} = \frac{VN}{N} \quad (2.3)$$

Taux faux positif [w13]

$$Taux\ faux\ positif = 1 - \frac{VN}{VN + FP} = 1 - \frac{VN}{N} = 1 - \text{taux vrai negatif} \quad (2.4)$$

Taux faux négatif [w13]

$$Taux\ faux\ negatif = 1 - \frac{VP}{VP + FP} = 1 - \frac{VP}{P} = 1 - \text{taux vrai positif} \quad (2.5)$$

F-mesure ‘moyenne harmonique’ [w13]

C’est la moyenne harmonique entre le rappel et la précision.

$$FM = \frac{1}{\frac{1}{P} + \frac{1}{R}} = 2 * \frac{P * R}{P + R} \quad (2.6)$$

Pour l'apprentissage de plusieurs classes, on généralise :

$$\mathbf{rappel} = \frac{\text{somme des rappels de chaque classe}}{\text{nombre de classes}} \quad (2.7)$$

$$\mathbf{précision} = \frac{\text{somme des précisions de chaque classe}}{\text{nombre de classes}} \quad (2.8)$$

La courbe ROC

La courbe ROC 'Receiver Operating Characteristic' permet d'analyser l'influence d'un paramètre du modèle : on fait varier le paramètre et on trace la courbe du rappel en fonction de 1-spécificité (taux de vrais positifs en fonction du taux de faux positifs) [w14].

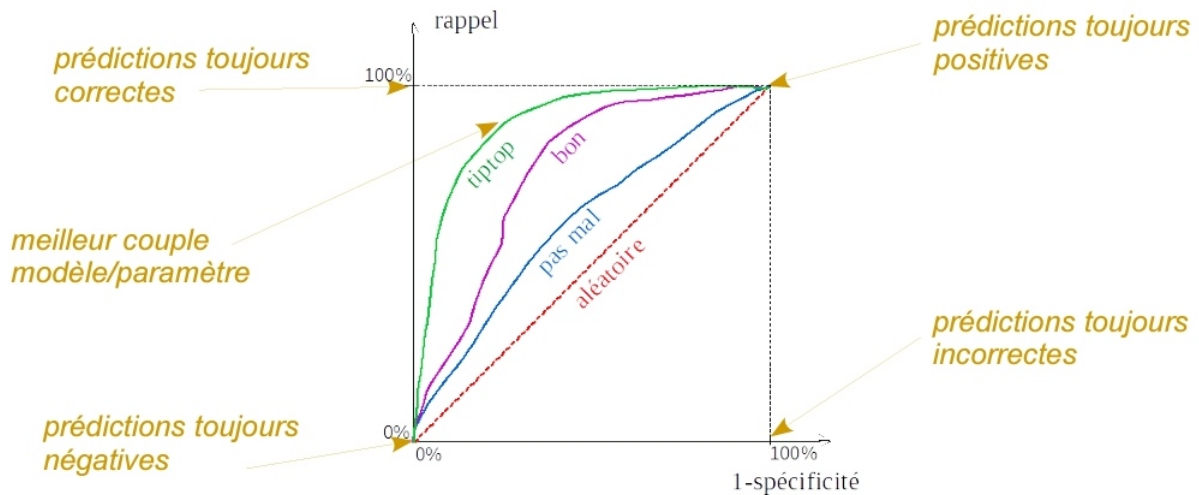


FIGURE 2.22 – la courbe ROC.

15 Conclusion

Les CNN est une technique d'apprentissage profond pour les tâches actuelles de détection d'objets. Comme toutes les techniques d'apprentissage profond, ils sont très dépendants de la taille et de la qualité des données d'entraînement. Avec un jeu de données bien préparé, les CNN sont capables de surpasser les humains dans les tâches de détection d'objets.

Chapitre 3

Conception

1 Introduction

L'un des domaines de la vision par ordinateur qui ont eu récemment un énorme intérêt, est la détection et la classification d'objets en images, car il est le cœur de nombreux systèmes importants comme les systèmes de reconnaissance faciale, les systèmes de sécurité et les systèmes de port de masque dans les lieux publics pendant la pandémie "COVID-19".

Devant le nombre croissant de caméras et l'impossibilité de placer un opérateur humain derrière chacune d'elles, la demande et le besoin d'outils d'analyse automatique des données récupérées ont fortement augmenté. Notre projet entre dans le cadre de 'la santé publique' dont l'objectif principal est la détection du port de masque sur des images dans les salles des cours. Pour ce faire nous avons utilisé les réseaux de neurones convolutifs connus pour leur performance dans ce type d'application.

2 Architecture du système

La figure 3.1 illustre l'architecture globale qui sera détaillée par la suite bloc par bloc dans les parties qui suivent.

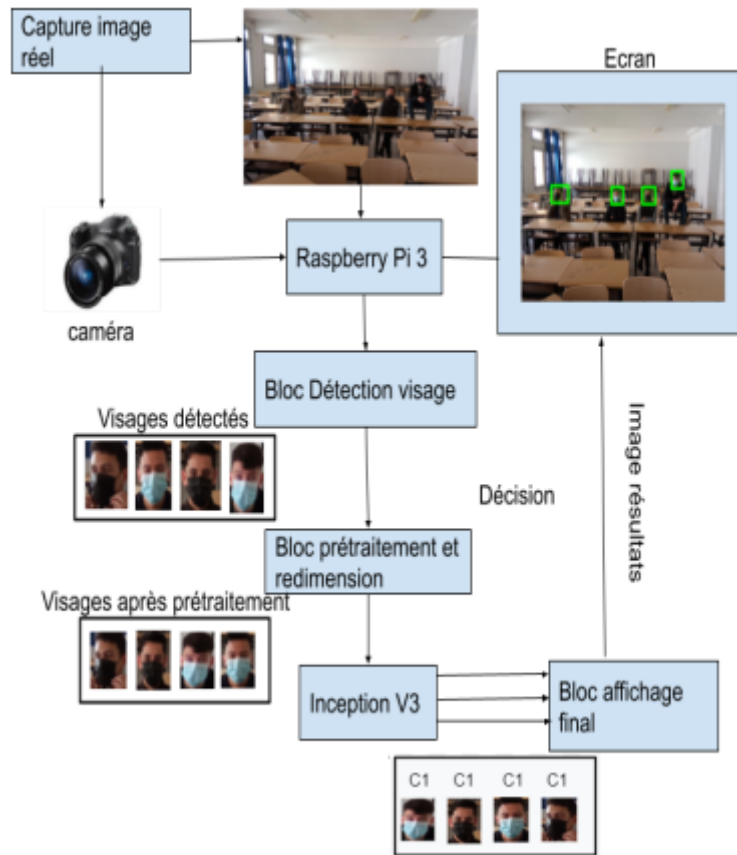


FIGURE 3.1 – Architecture du système globale

La figure 3.2 illustre l'architecture de l'entraînement du système.

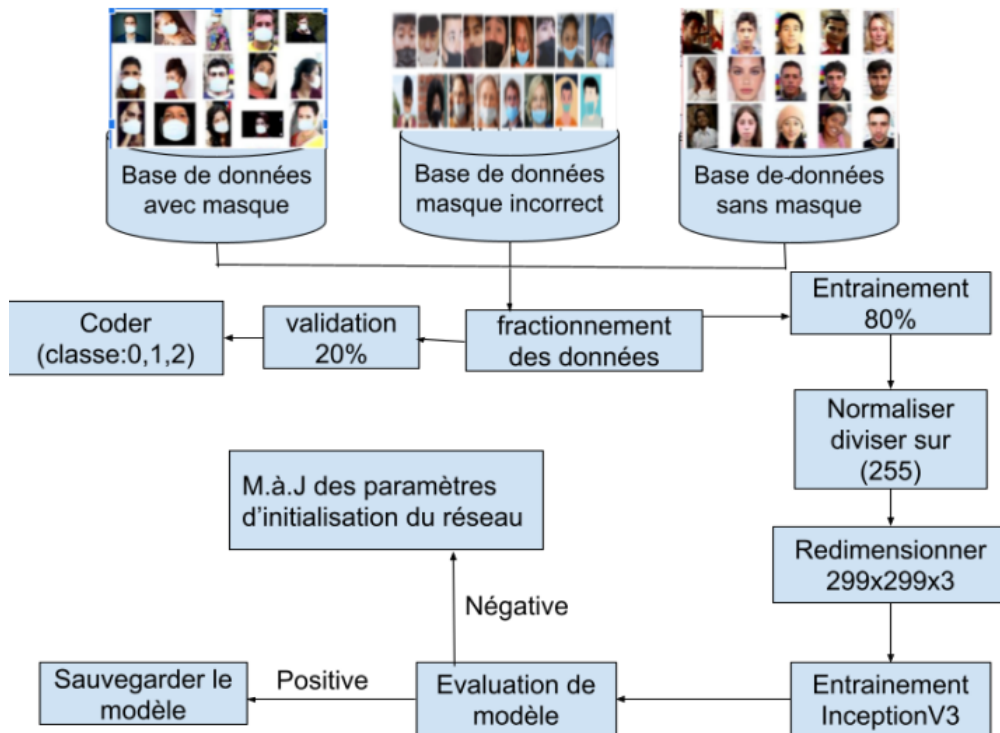


FIGURE 3.2 – Architecture de l’entraînement

La figure 3.3 illustre l’architecture de test du système.

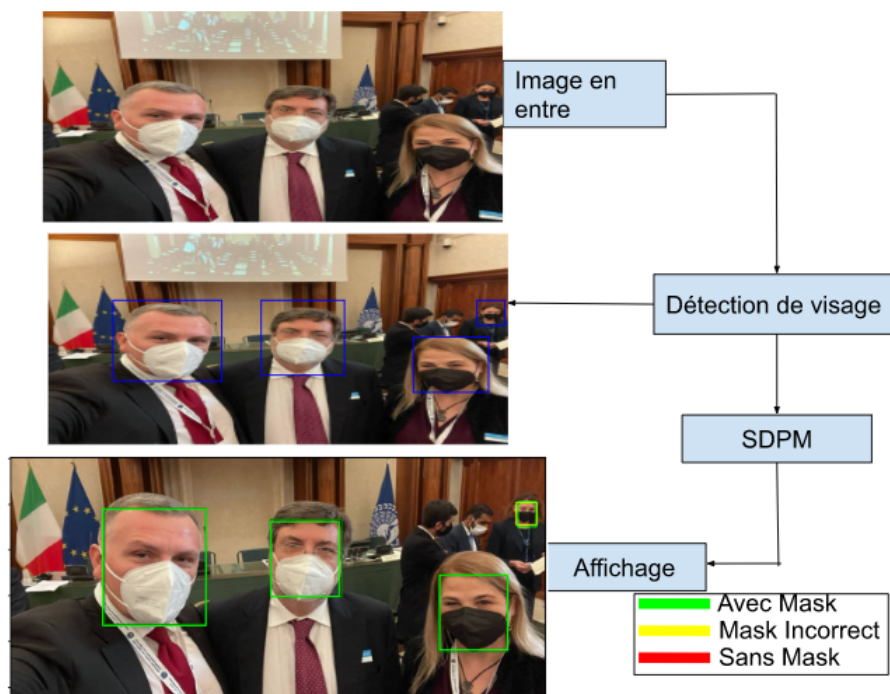


FIGURE 3.3 – Architecture de test

3 Redimensionnement de l'image

Tous les images ont été redimensionnées à une taille variable entre (75x75) et (299x299) afin de les adapter au réseaux utilise.

4 Détection de visage

Pour le test finale nous avons importé une bibliothèque python "Multi-task Cascaded Convolutional Neural Networks"(MTCNN) comme solution pour la détection et l'alignement des visages. Le processus consiste en trois étapes de réseaux convolutifs capables de reconnaître les visages et les points de repère tels que les yeux, le nez et la bouche [23].

Les trois étapes du MTCNN sont : La première étape consiste à prendre l'image et à la redimensionner à différentes échelles afin de construire une pyramide d'images, qui est l'entrée du réseau en cascade à trois étages suivant.

La figure 3.4 illustre l'image d'entrée est redimensionnée à différentes échelles pour créer une pyramide d'images [23].

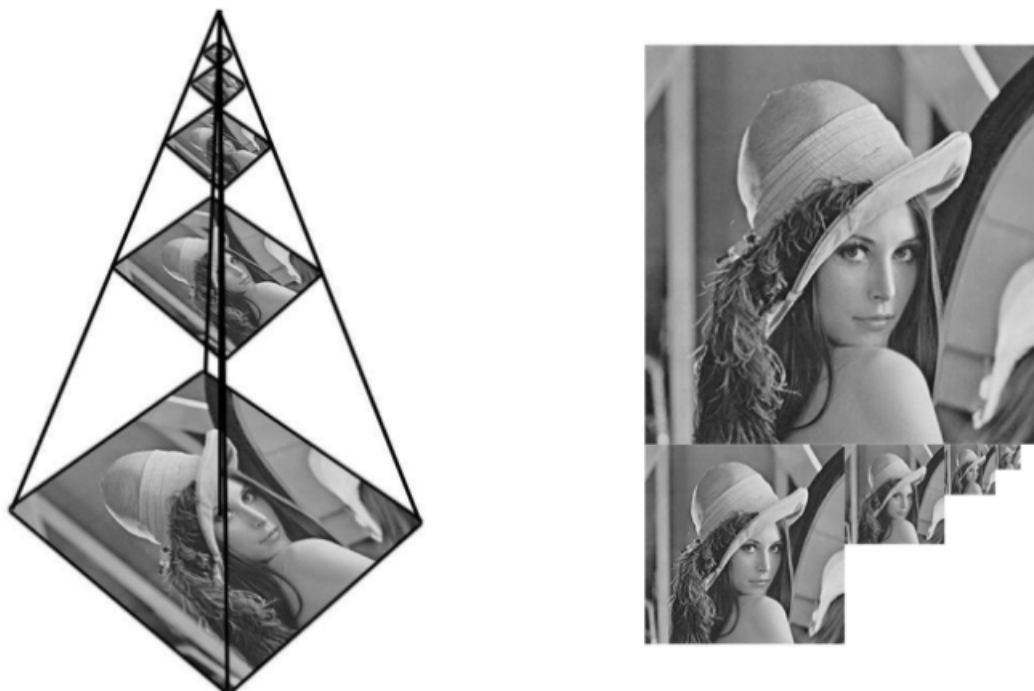


FIGURE 3.4 – L'image d'entrée on forme de pyramide d'image

4.1 Étape 1 : Le réseau de propositions (P-Net)

Cette première étape est un réseau entièrement convolutif (FCN). La différence entre un CNN et un FCN est qu'un réseau entièrement convolutif n'utilise pas de couche dense dans le cadre de l'architecture. Ce réseau de proposition est utilisé pour obtenir des fenêtres candidates et leurs vecteurs de régression de boîte englobante. Le résultat final de cette étape est toutes les fenêtres candidates après raffinement pour réduire le volume de candidats [23]. La figure 3.5 illustre l'architecture de l'étape de réseau 'P-Net'.

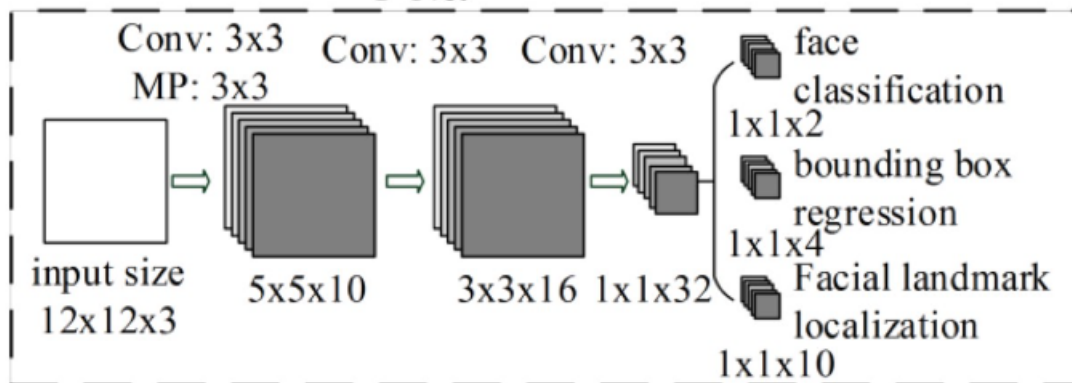


FIGURE 3.5 – P-Net

4.2 Étape 2 : Le réseau Refine (R-Net)

Le P-Net est un réseau CNN, et non un FCN comme le précédent car il existe une couche dense au dernier étage de l'architecture du réseau qui est illustrée dans la figure 3.6. Le R-Net réduit encore le nombre de candidats, effectue un calibrage avec une régression de boîte englobante et utilise la suppression non maximale (NMS) pour fusionner les candidats qui se chevauchent. Les sorties R-Net indiquent si l'entrée est un visage ou non, un vecteur à 4 éléments qui est la boîte englobante du visage et un vecteur à 10 éléments pour la localisation des repères faciaux [23].

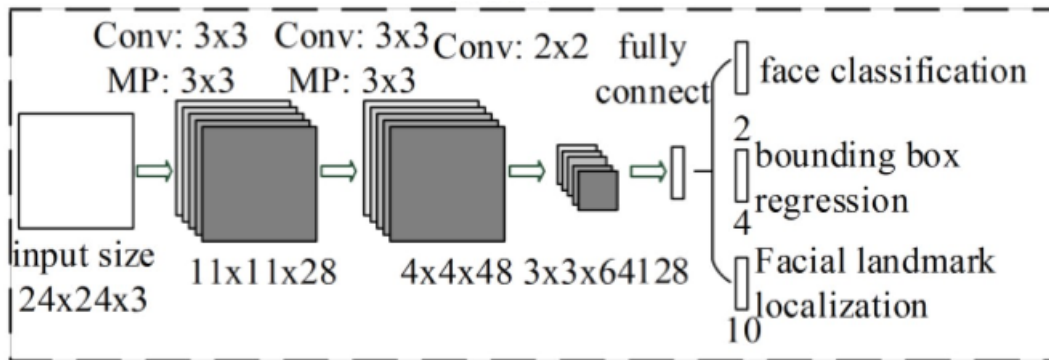


FIGURE 3.6 – R-Net

4.3 Étape 3 : Le réseau de sortie (O-Net)

Cette étape est similaire au R-Net, mais ce réseau de sortie vise à décrire le visage plus en détail et à produire les positions des cinq repères faciaux pour les yeux, le nez et la bouche. La figure 3.7 illustre l'architecture de l'étape de réseau 'O-Net' [23].

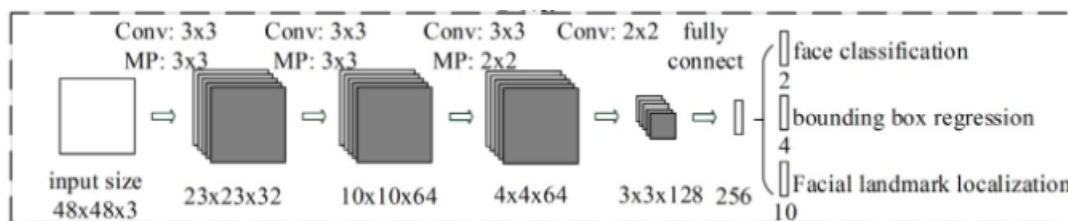


FIGURE 3.7 – O-Net

5 Réseaux Inception

Le réseau Inception a été une étape importante dans le développement des classificateurs CNN. Avant sa création (jeu de mots), les CNN les plus populaires empilaient simplement des couches de convolution de plus en plus profondes, dans l'espoir d'obtenir de meilleures performances.

La figure 3.8 le module de démarrage "naïf". Il effectue une convolution sur une entrée, avec 3 tailles de filtres différentes (1x1, 3x3, 5x5). De plus, la mise en commun maximale est également effectuée. Les sorties sont concaténées et envoyées au module de démarrage suivant. La figure 3.9 illustre un architecture naïf plus complexe avec des réduction [24].

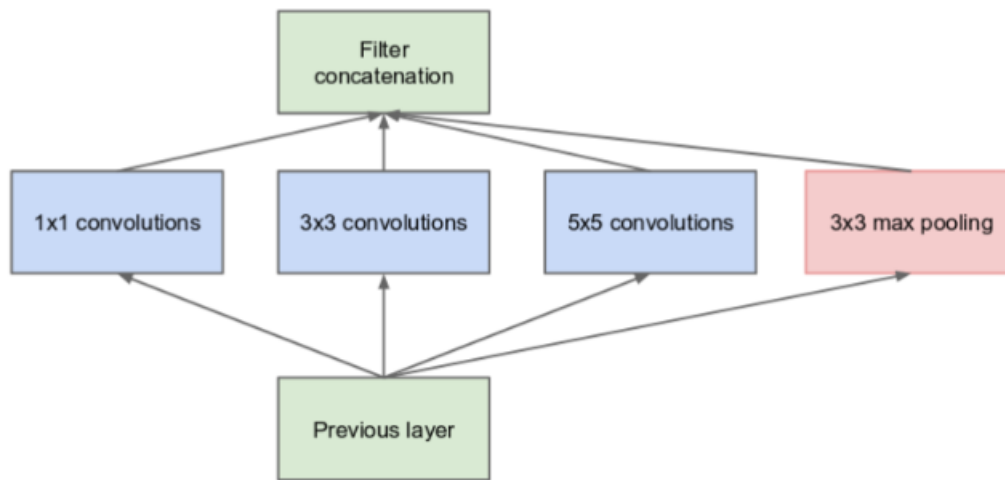


FIGURE 3.8 – Architecture naïf d’Inception

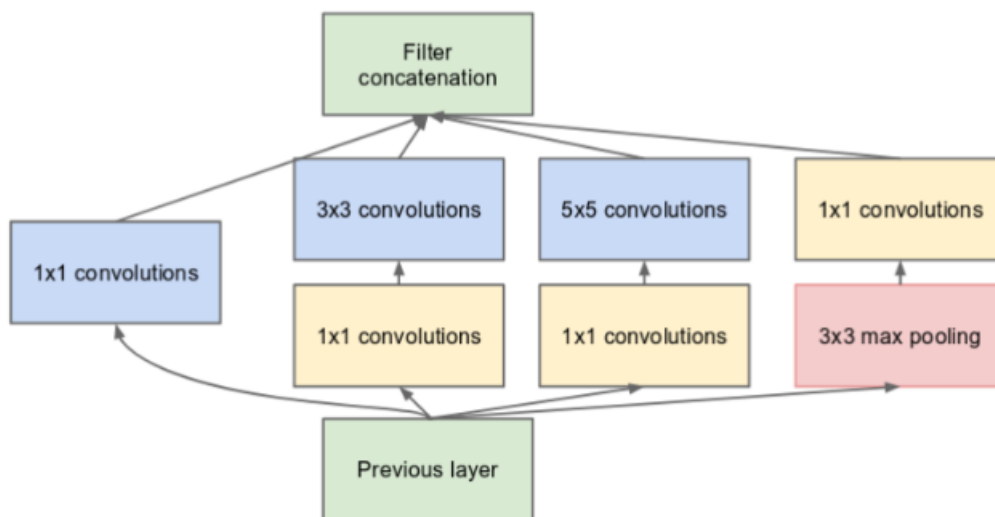


FIGURE 3.9 – Architecture Inception avec réduction

5.1 InceptionV1

En utilisant le module de démarrage à dimension réduite, une architecture de réseau de neurones a été construite. Ceci était populairement connu sous le nom de GoogLeNet (Inception v1) [25]. L’architecture est illustrée dans la figure 3.10



FIGURE 3.10 – Architecture Inception V1

5.2 InceptionV2

L'architecture est illustrée dans la figure 3.14 Factorise la convolution 5x5 en deux opérations de convolution 3x3 pour améliorer la vitesse de calcul. Bien que cela puisse sembler contre-intuitif, une convolution 5x5 est 2,78 fois plus chère qu'une convolution 3x3. Ainsi, l'empilement de deux convolutions 3x3 conduit en fait à une augmentation des performances. Ceci est illustré dans la figure 3.12. De plus, ils factorisent les convolutions de taille de filtre $n \times n$ en une combinaison de convolutions $1 \times n$ et $n \times 1$. Par exemple, une convolution 3x3 équivaut à effectuer d'abord une convolution 1x3, puis à effectuer une convolution 3x1 sur sa sortie. Ils ont trouvé que cette méthode était 33% moins chère que la simple convolution 3x3. Ceci est illustré dans la figure 3.13. Les banques de filtres du module ont été étendues (élargies au lieu d'être plus profondes) pour supprimer le goulot d'étranglement de la représentation. Si le module était plutôt plus profond, il y aurait une réduction excessive des dimensions, et donc une perte d'informations [26]. Ceci est illustré dans la figure 3.14.

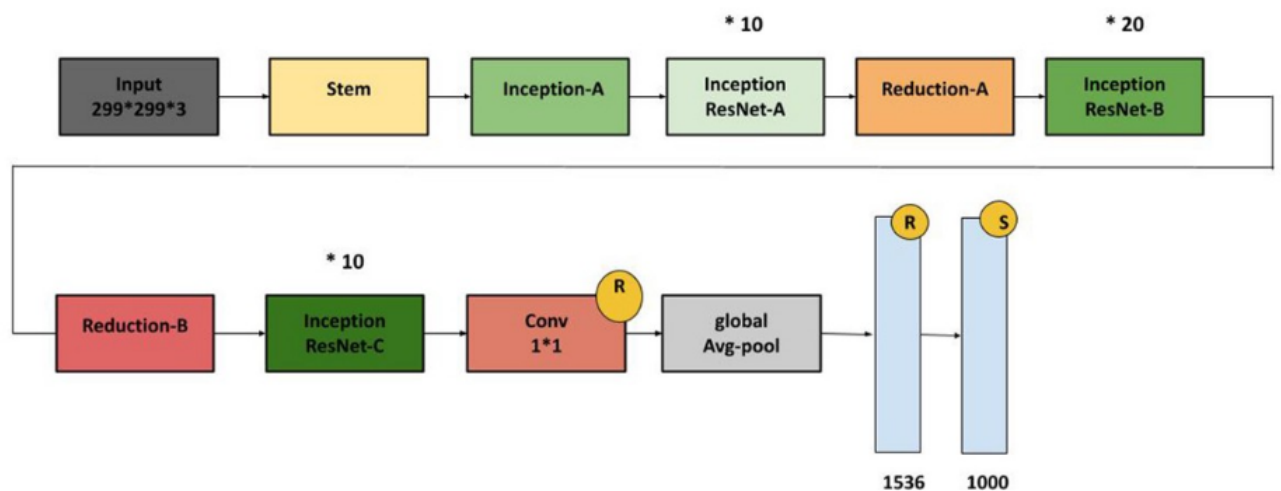


FIGURE 3.11 – Architecture inception v2

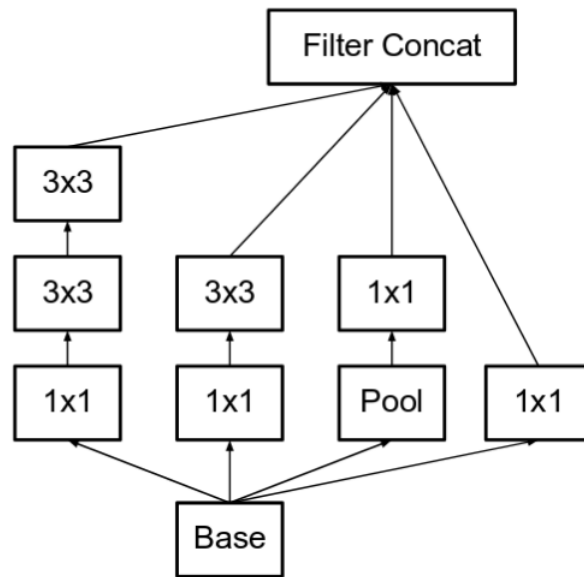


FIGURE 3.12 – Factorisation de convolution 5x5

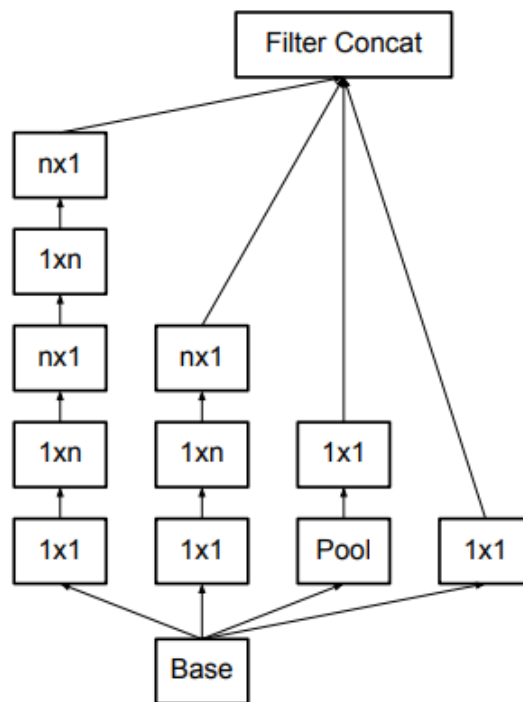


FIGURE 3.13 – Factorisation de convolution 1xn et nx1

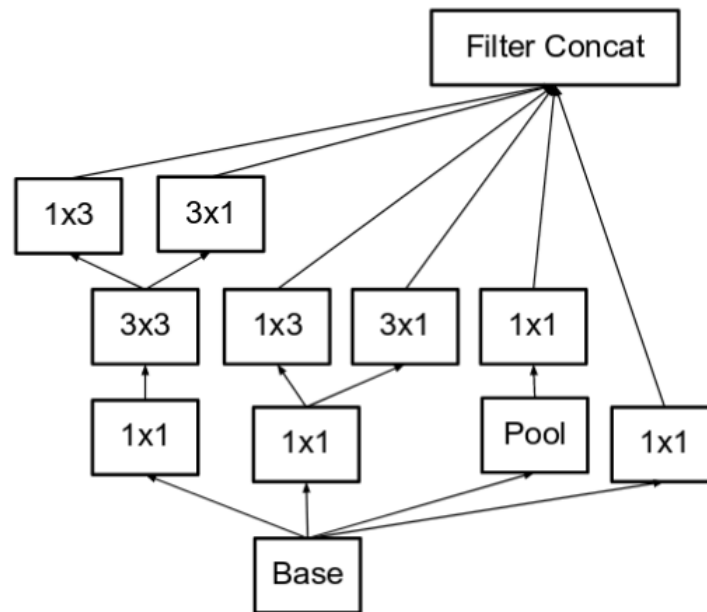


FIGURE 3.14 – Banque de filtres d'inception

5.3 InceptionV3

Inception v3 illustre dans la figure 3.15 a incorporé toutes les mises à niveau de Inception v2, et a également utilisé les éléments suivants [25] :

1. Optimiseur RMSProp.
 2. Convolutions 7x7 factorisées.
 3. BatchNorm dans les classificateurs auxiliaires.
 4. Label Smoothing (Un type de composant de régularisation ajouté à la formule de perte qui empêche le réseau de devenir trop confiant à propos d'une classe. Empêche le surajustement).
- B : c'est un batch normalisation.
 - R : c'est une fonction d'activation ReLU.
 - S : c'est la fonction d'activation.
 - Les bares bleu : c'est les couche Dense.

5.6 Détail de l'architecture du reseau InceptionV3

On va détailler l'architecture d'inception v3 illustrée dans la figure 3.15

Inception A

La figure 3.16 illustre l'architecture du Inception A.

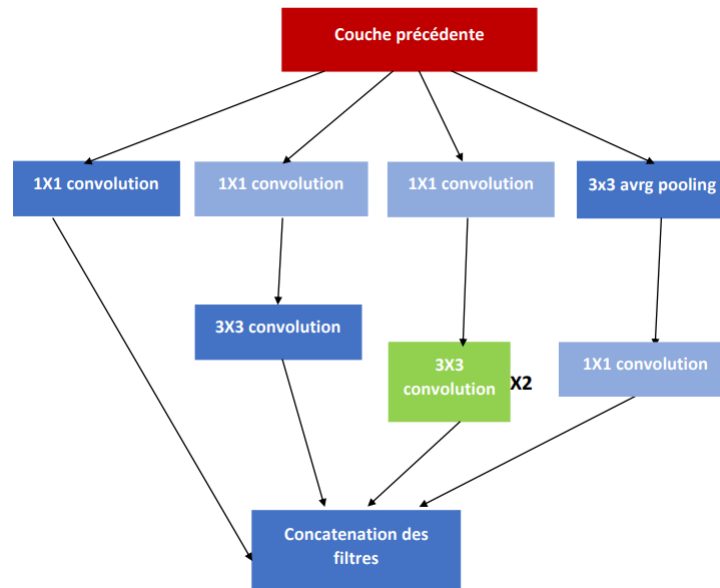


FIGURE 3.16 – Architecture du inception A

Inception B

La figure 3.17 illustre l'architecture du Inception B.

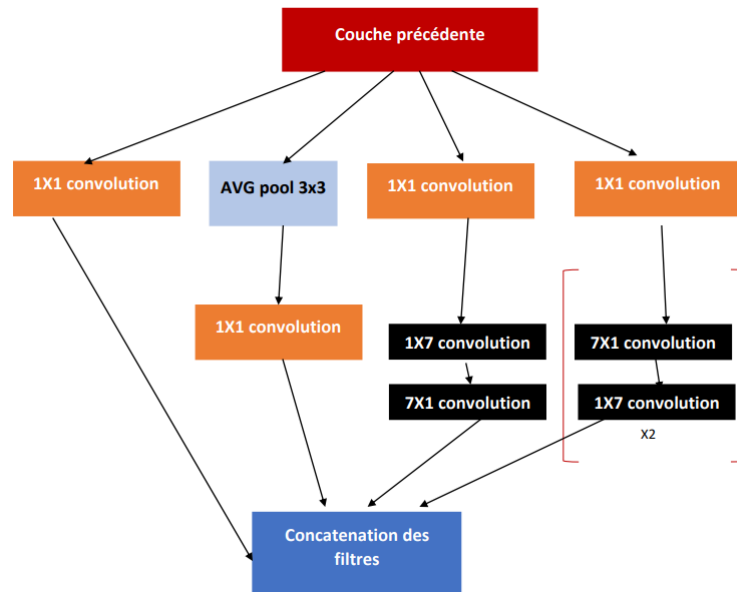


FIGURE 3.17 – Architecture du Inception B

Inception C

La figure 3.18 illustre le détaillé de l'architecture du Inception C.

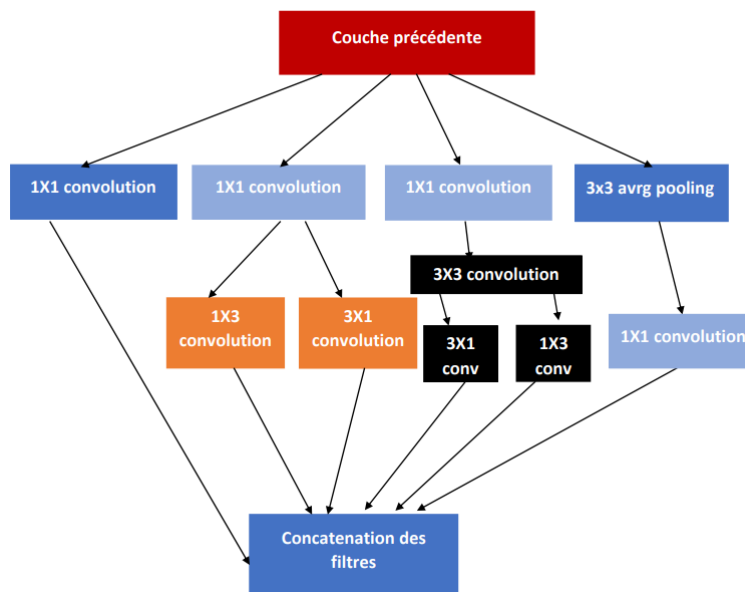


FIGURE 3.18 – Architecture du Inception C

Réduction du Inception A et Inception B

Les réduction de l'architecture Inception A et Inception B sont illustrées par la figure 3.19 et la figure 3.20.

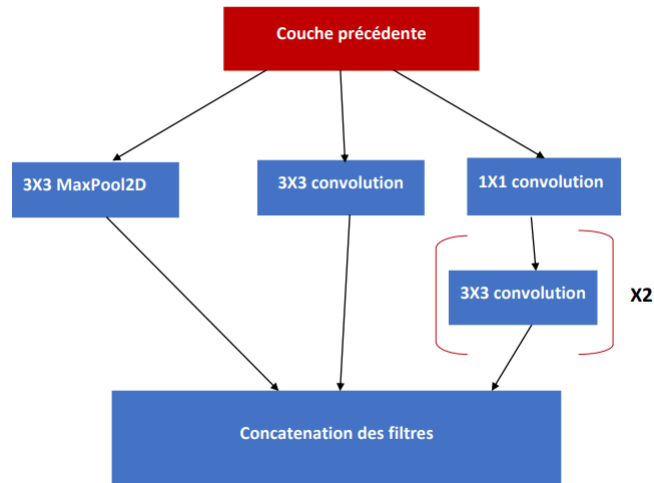


FIGURE 3.19 – Architecture Réduction Inception A

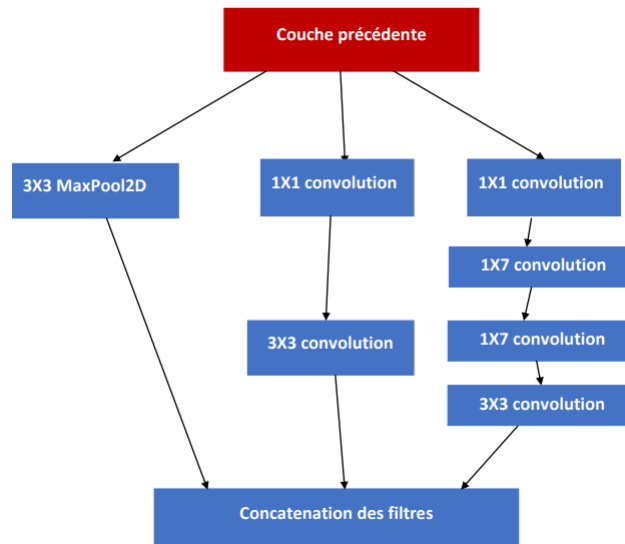


FIGURE 3.20 – Architecture Réduction Inception B

Bloc classificateur auxiliaire

La figure 3.21 illustre le bloc classificateur auxiliaire dont le rôle est d'améliorer les performances globales du réseau.

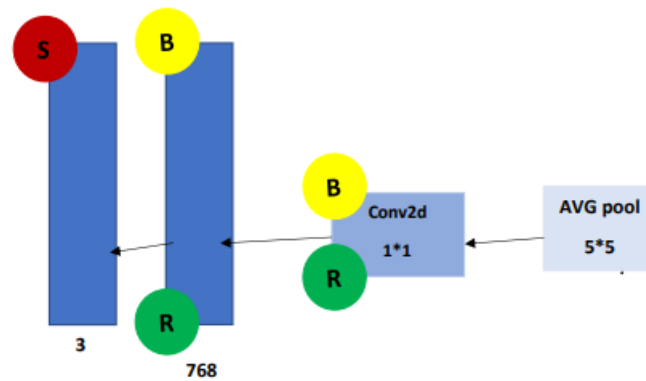


FIGURE 3.21 – Architecture Bloc classificateur auxiliaire

6 Conclusion

La fiabilité des réseaux de neurones convolutifs dépend du jeu de données (base de données) et du réseau d'apprentissage. L'utilisation des réseaux pré-entraînés tel que l'InceptionV3 est très fiable mais nécessite un renforcement de données et de l'apprentissage (Apprentissage par transfert) sur des classes désignées selon l'application.

Chapitre 4

Implémentions et résultats

1 Introduction

Dans cette partie du travail nous allons faire une implémentation pour valider l'architecture du système proposé avec un langage adéquat (dans notre cas python) et une série de tests sur des images réel.

2 Matériel utilisé

Durant la conception de notre application nous avons utilisé le matériel suivant :

1. Raspberry Pi 3.
2. Caméra R pi (caméra Raspberry Pi).
3. Écran d'affichage.
4. station de calcul.
5. PC personnel "DELL".

2.1 Fiches techniques Raspberry Pi

Raspberry Pi (Tarte aux framboises)

Le Raspberry Pi est un petit ordinateur à faible coût qui se connecte à un écran d'ordinateur ou à un téléviseur et fonctionne avec un clavier et une souris ordinaires, comme

illustré dans Figure 4.1. C'est un petit gadget pratique qui se concentre sur l'enseignement des langages de script comme Scratch et Python aux personnes de tous âges. Il peut exécuter toutes les fonctions d'un ordinateur de bureau, telles que la navigation sur Internet et l'affichage de clips en plus grande définition, de feuilles de calcul et de jeux. Il a été utilisé dans plusieurs appareils numériques, notamment des nichoirs à tweets, des machines à musique et des détecteurs, ainsi que des stations météorologiques et des caméras infrarouges, car il est capable d'interagir avec l'environnement extérieur.

Il dispose d'un chipset quadricœur à 1,2 GHz BCM2387 avec un support GPU d'un coprocesseur multimédia dual-core et d'un noyau vidéo et du GPU, qui comprend un coprocesseur multimédia double cœur, y compris un Bluetooth 4,1 (Bluetooth et Bluetooth Classique). Avec Bluetooth Low Energy (BLE) et Wi-Fi BCM43143, le Raspberry Pi 3 offre une mise à niveau vers un nouveau processeur principal et une mise en réseau améliorée. De plus, la gestion de l'alimentation du Raspberry 3 a été améliorée, avec une alimentation améliorée de 3,5 ampères pouvant gérer des périphériques USB externes plus puissants. Les ports USB intégrés du Raspberry Pi 3 offrent une connectivité suffisante pour relier la souris ou tout autre élément au RPi. La plupart des puces du système Raspberry Pi peuvent être overclockées à 800 MHz, et certaines peuvent être overclockées à 1000 MHz. Il est rapporté que le Raspberry Pi 2 peut être overclocké de la même manière, atteignant même 1500 MHz dans les cas extrêmes (sans toutes les fonctions de sécurité et restrictions de surtension). Sur les distributions Linux, vous pouvez utiliser la commande du programme pour exécuter « `sudo raspiconfig` » afin d'effectuer un overclocking au démarrage sans rompre la garantie. Dans certains cas, le Pi désactivera automatiquement l'overclocking lorsque la température de la puce atteint 85 °C pour annuler les paramètres automatisés d'overclocking et d'overclocking (ce qui annulera la garantie). Un refroidisseur peut être utilisé pour éviter la surchauffe du Raspberry Pi [6].

La figure 4.2 Raspberry Pi 3 utilisé dans notre projet.

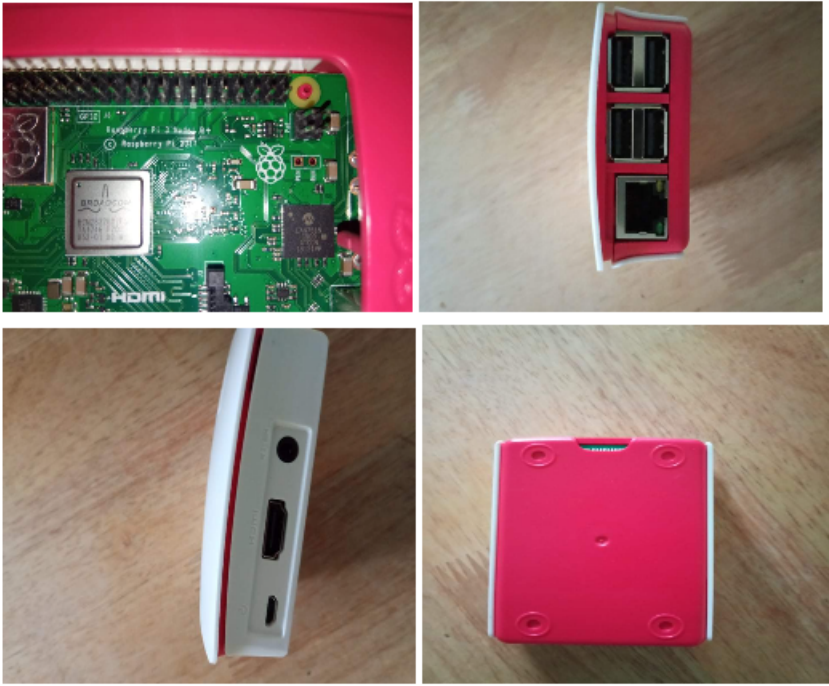


FIGURE 4.1 – Raspberry Pi

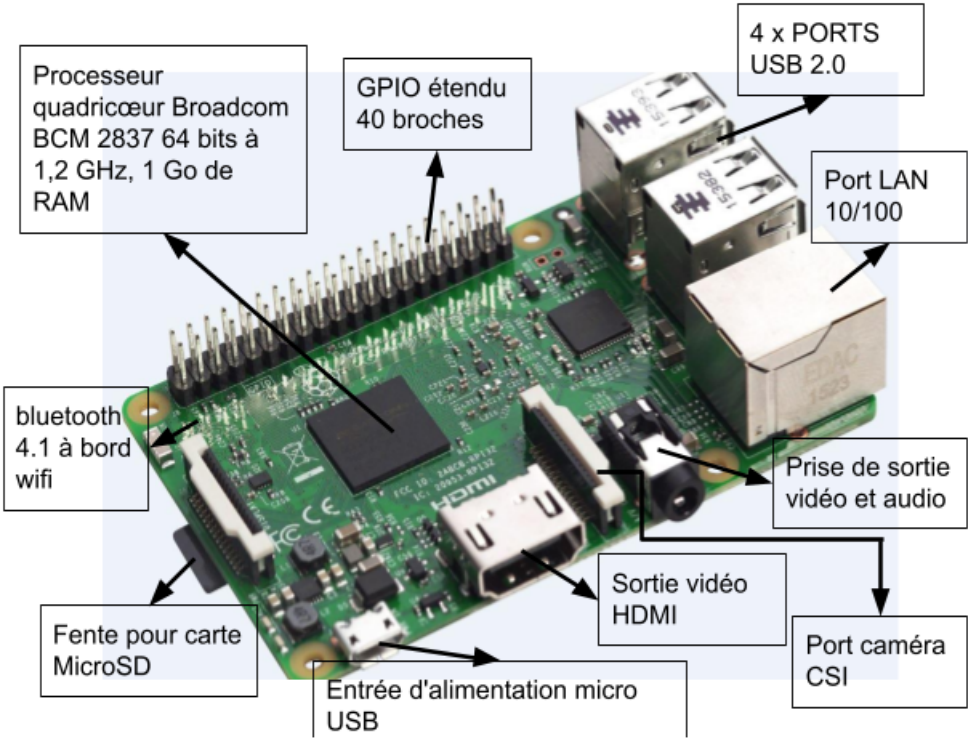


FIGURE 4.2 – Les composants de Raspberry Pi 3

Caméra R pi (caméra Raspberry Pi)

La Raspberry Pi Camera Module V2 est une caméra 8 Mégapixels avec capteur Sony IMX219 pour Raspberry Pi. Facile à installer, elle permet de doter votre système ARM Raspberry Pi d'une caméra 8 Mégapixels. Une solution idéale pour transformer votre Raspberry Pi en Webcam, Caméra IP ou Caméra de vidéosurveillance [6].

Principales caractéristiques :

- Module caméra 8 Mégapixels pour Raspberry Pi.
- Résolution photo : 3280 x 2464 pixels.
- Résolution vidéo : 1080p30, 720p60, 640x480p90

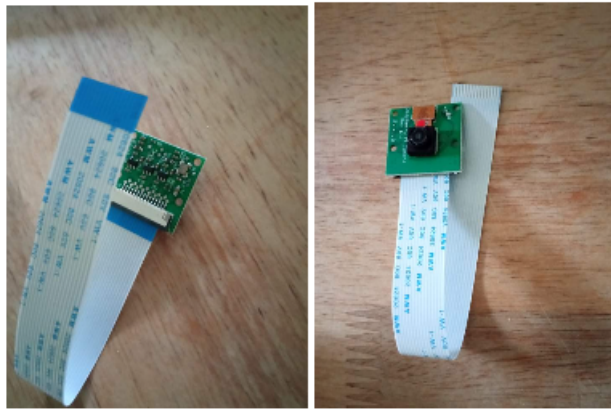


FIGURE 4.3 – Caméra Raspberry Pi

2.2 Mise on marche de Raspberry Pi

Raspberry Pi

Etape 1 : Télécharger la dernière version de Raspbian Téléchargeons la dernière version du fichier Image disque de Raspbian sur le site de la fondation Raspberry Pi : sur le site [w16].

Etape 2 : Etcher Nous flashons le fichier image à l'aide de Etcher sur la carte SD. Etcher (officiellement intitulé balenaEtcher) est un outil qui condense tout le processus de gravure des images du disque dur sur des disques SD ou clé USB externes à un seul endroit. Ce programme s'occupe de traiter l'image, formater le disque dur externe et le fla-

sher avec le bon format pour qu'il s'ouvre automatiquement sur l'appareil qu'il faut [w17].

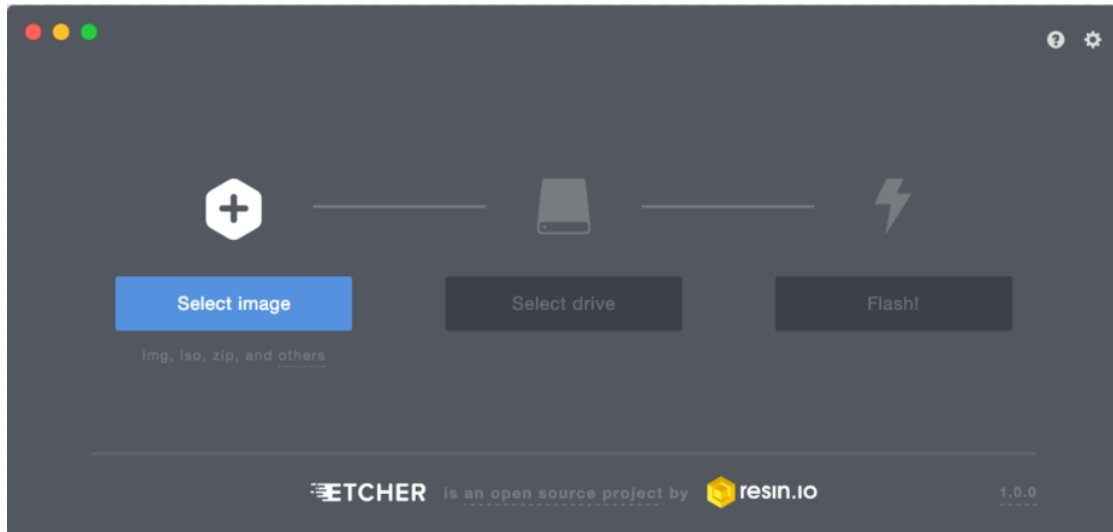


FIGURE 4.4 – L'interface Etcher

Etape 3 : Inséré la carte SD et démarer le Raspberry Pi.

caméra Raspberry Pi

La caméra se connecte au port CSI, situé entre les connecteurs HDMI et jack.

1. Tirez doucement le verrou du connecter (morceau blanc) vers le haut pour le débloquer ;
2. Insérez-y la nappe de la caméra sans forcer, la partie bleue tournée vers la prise jack ;
3. Refermez le verrou, la nappe est maintenant maintenue.

la figure 4.5 montre comment branche la caméra.

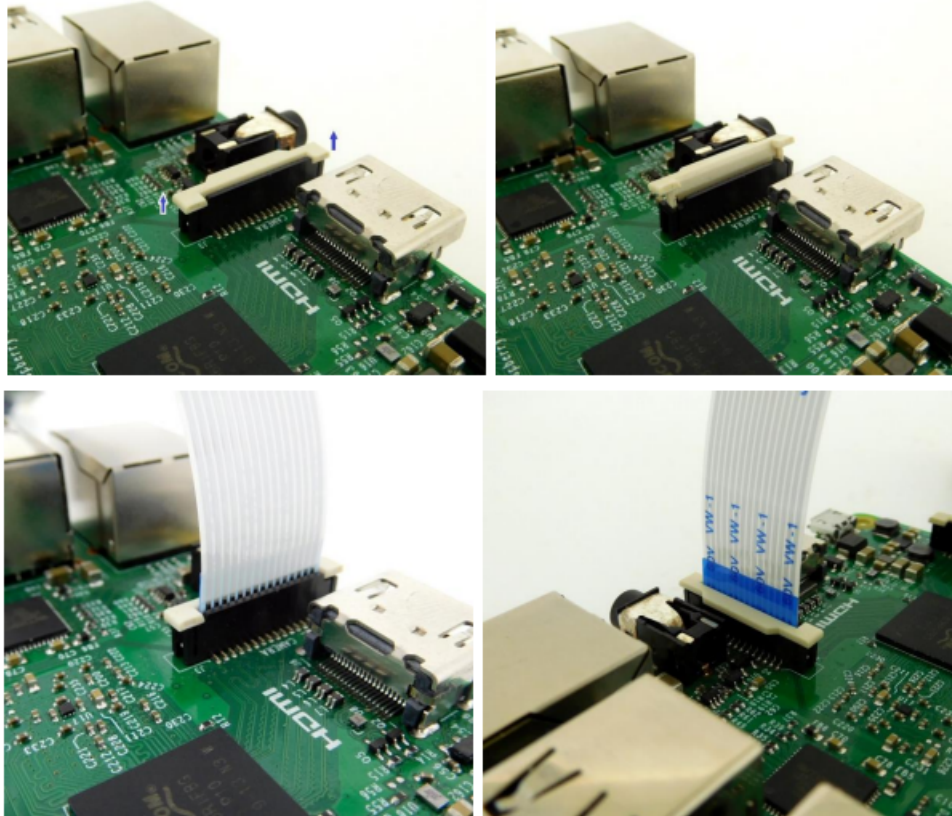


FIGURE 4.5 – Brancher la caméra

Configurer la caméra Raspberry Pi

Le système Raspbian n'est pas configuré pour utiliser la caméra d'origine, vous devez activer l'option dans les configurations :

la figure 4.6 illustre le détail de la configuration de la caméra Raspberry Pi.

Allez dans le menu démarrer > Préférences > Configuration du Raspberry Pi. Ensuite dans la fenêtre de configuration, sélectionnez l'onglet Interfaces et cochez la case Activé en face de Caméra.

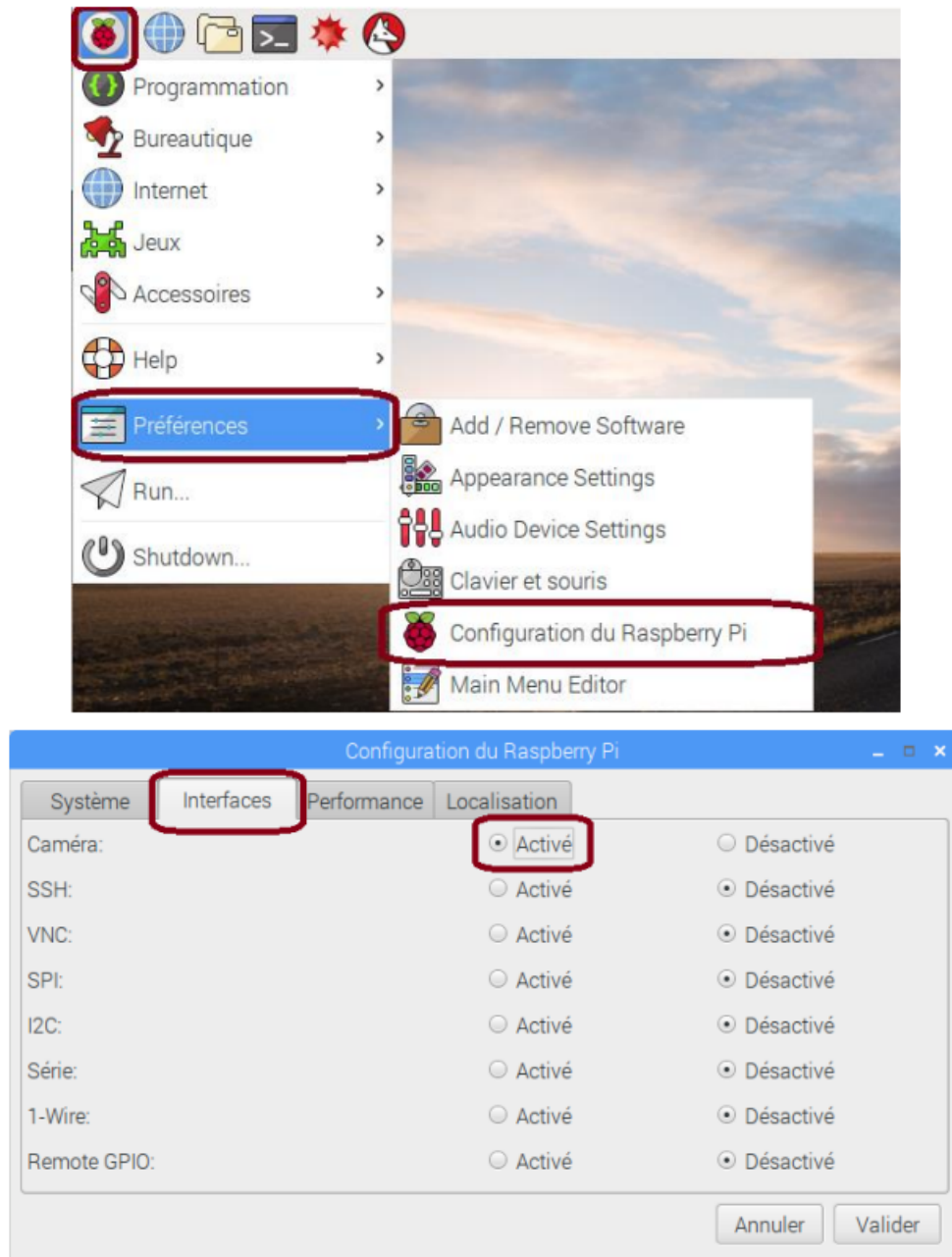


FIGURE 4.6 – Configuration de la caméra

2.3 Station de calcul

Spécifications de l'appareil

- Nom de l'appareil : DESKTOP-074k3R3
- Processeur : Interl(R) Core(TM) i7-7700k CPU @ 4.20GHz 4.20GHz
- Type du système : Système d'exploitation 64 bits, processeur x64.

Spécifications de Windows

- Edition : Windows 10 Professionnel Education
- Version : 21H2
- Installé : 201-07-09
- Build du système d'exploitation : 19044.1706
- Expérience : Windows Feature Experience Pack 120.2212.4170.0



FIGURE 4.7 – Station de calcul du laboratoire LAIG

Les figure 4.8 suivantes montre le matériel utilisé pour notre application.



FIGURE 4.8 – Matériels utilisé dans notre application

3 Logiciel utilisé

3.1 Python

Python est un langage de programmation de haut niveau à usage général et largement utilisé. Il a été créé par ‘Guido van Rossum’ en 1991 et développé par ‘The Python Software Foundation’. Python est un langage de programmation interprété, multi-paradigme et multi-plateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d’un typage dynamique fort, d’une gestion automatique de la mémoire par ramasse-miettes et d’un système de gestion d’exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl. Il existe deux versions majeures de Python : Python 2 et Python 3. Les deux sont assez différents [27].

3.2 OpenCV-Python

‘OpenCV-Python’ est une bibliothèque de liaisons Python conçue pour résoudre les problèmes de vision par ordinateur. Comparé à des langages comme « C / C ++ », ‘Python’ est plus lent. Cela dit, Python peut être facilement étendu avec « C / C ++ », ce qui nous permet d’écrire du code intensif en calcul en « C / C ++ » et de créer des wrappers Python qui peuvent être utilisés comme modules ‘Python’. Cela nous donne deux avantages : premièrement, le code est aussi rapide que le code « C / C ++ » original (puisque c’est le code C ++ réel travaillant en arrière-plan) et deuxièmement, il est plus facile de coder en Python qu’en « C/C++ ». ‘OpenCV-Python’ est un ‘Python wrapper’ pour l’implémentation ‘OpenCV C ++’ d’origine [28].

3.3 Numpy

‘NumPy’ est une bibliothèque python « Numerical Python » utilisée pour travailler avec des tableaux. Il a également des fonctions pour travailler dans le domaine de l’algèbre linéaire, de la transformée de Fourier et des matrices. ‘NumPy’ a été créé en 2005 par ‘Travis Oliphant’. C’est un projet open source et nous pouvons l’utiliser librement [29].

3.4 Pandas

'Pandas' est une bibliothèque écrite pour le langage de programmation Python permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles [w18].

3.5 Matplotlib

Matplotlib est la bibliothèque de visualisation ou de traçage de base du langage de programmation Python. Matplotlib est un outil puissant pour exécuter une variété de tâches. Il est capable de créer différents types de rapports de visualisation tels que des tracés linéaires, des nuages de points, des histogrammes, des diagrammes à barres, des diagrammes circulaires, des diagrammes en boîte et bien d'autres tracés différents [w19].

3.6 TensorFlow

TensorFlow est une bibliothèque de logiciels d'apprentissage automatique open source et gratuite. Il a été créé pour effectuer de gros calculs numériques sans tenir compte de l'apprentissage en profondeur. Ce TensorFlow peut être utilisé pour diverses activités, mais il est principalement axé sur l'inférence et la formation de réseaux de neurones profonds. TensorFlow prend également en charge l'apprentissage automatique traditionnel. TensorFlow de Google est une bibliothèque Python qui permet un calcul numérique rapide. Les modèles d'apprentissage en profondeur sont soit générés directement à l'aide de TensorFlow, qui est également une bibliothèque de base, soit créés pour simplifier le processus à l'aide de bibliothèques wrapper construites au-dessus de TensorFlow. TensorFlow permet la création de graphiques et de structures de flux de données pour déterminer comment les données circulent dans le graphique en recevant des entrées sous la forme d'un tableau de tenseurs multidimensionnel [6].

3.7 Keras

Keras est une API conçue pour les êtres humains, pas pour les machines. Keras suit les meilleures pratiques pour réduire la charge cognitive : il propose des API cohérentes et simples, il minimise le nombre d'actions de l'utilisateur requises pour les cas d'utilisation courants et il fournit des messages d'erreur clairs et exploitables. Il contient également une documentation complète et des guides de développement [30].

3.8 Tkinter

Tkinter (de l'anglais Tool kit interface) est la bibliothèque graphique libre d'origine pour le langage Python, permettant la création d'interfaces graphiques. Elle vient d'une adaptation de la bibliothèque graphique Tk écrite pour Tcl [31].

3.9 Os

Le module OS en python fournit des fonctions d'interaction avec le système d'exploitation. 'OS' fait partie des modules utilitaires standard de Python. Ce module fournit un moyen portable d'utiliser les fonctionnalités dépendant du système d'exploitation. Les modules 'os' et 'os.path' incluent de nombreuses fonctions pour interagir avec le système de fichiers [32].

4 Base d'apprentissage

Pour l'entraînement de notre réseau Nous avons utilisé le jeux de données 'Détection de masque facial-Images 12K'[w20] pour les images avec et sans masque, par-contre pour la port incorrect nous avons telecharger des photos sur internet, pour l'adapter avec notre réseau, nous avons fait quelques modifications sur la forme de cette base, Tableau 4.1 .

classes	Apprentissage 80%	test 20%	total
Visage "avec masque"	4 744	1 186	5 930
Visage "sans masque"	4 717	1 179	5 896
Visage "masque incorrect"	1 051	263	1 314
total	10 512	2 628	13 140

TABLE 4.1 – Base "Détection de masque facial-Images 12K" adaptée pour InceptionV3

La figure 4.9 illustre quelques échantillons de la base de données utilisée.



FIGURE 4.9 – Exemples d'images de l'ensemble de données '12K-images'

5 Apprentissage et Test

5.1 Apprentissage du réseau 'InceptionV3'

Après plusieurs test et changements des caractéristiques du réseaux 'InceptionV3', Nous avons obtenu le meilleur apprentissage avec la configuration suivante : `optimiser= RM-Sprop(learning_rate = 0.0001)`, `loss='categorical_crossentropy'`, `'batch_size = 10'` (La figure 4.10) illustre les résultats d'apprentissage et de test représente par les graphes de précision 'Accuracy', et (La figure 4.11) illustre les résultats d'apprentissage et de test représente par les graphes de perte 'Loss'.

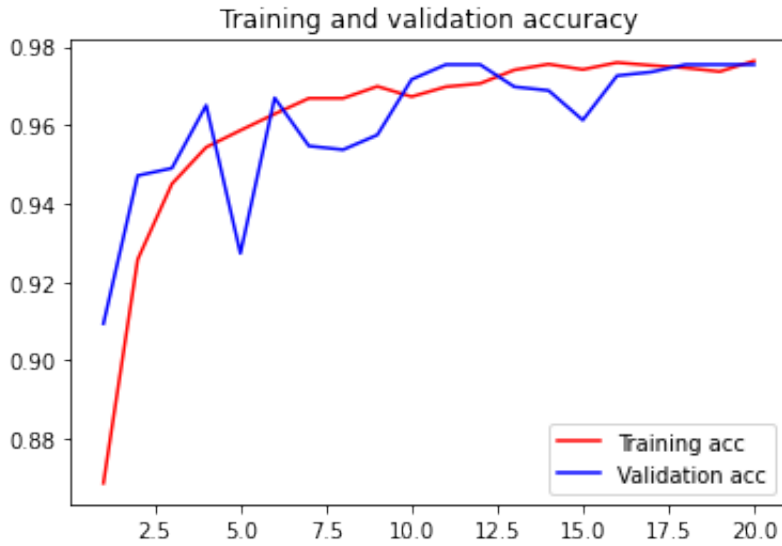


FIGURE 4.10 – le graphe de ‘accuracy’ pour apprentissage de ‘InceptionV3’

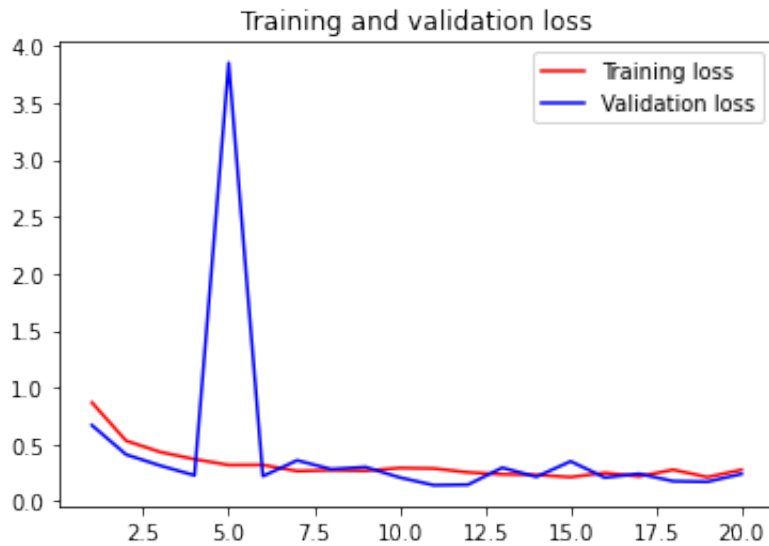


FIGURE 4.11 – le graphe de ‘loss’ pour d’apprentissage de ‘InceptionV3’

Nous remarquons que le réseau a atteint une précision de (97%) durant l’apprentissage et (97%) durant le test. Et une perte de (0.4%) durant l’apprentissage et (0.3%) pour le test. Considérés comme acceptable pour lancer les tests sur des images de scène complète (tableau 4.1)

5.2 Test de réseaux 'InceptionV3'

Nous avons fait une expérience sur 2628 images des trois classes qui n'appartenaient pas à la base de données utilisée dans l'entraînement. La figure 4.12 illustre le résultat de la matrice de confusion :

- C0 : masque incorrect.
- C1 : avec masque.
- C2 : sans masque.

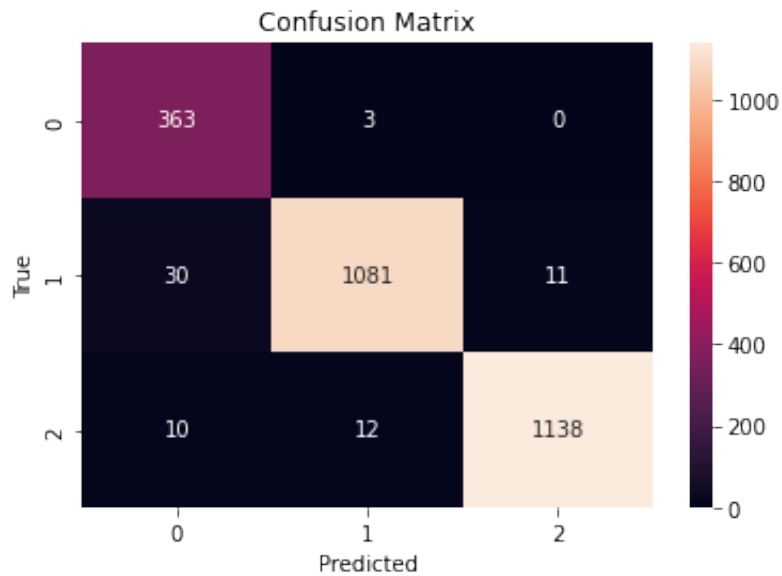


FIGURE 4.12 – Matrice de confusion

6 Interface graphique

La figure 4.13 illustre la page d'accueil de l'interface graphique.

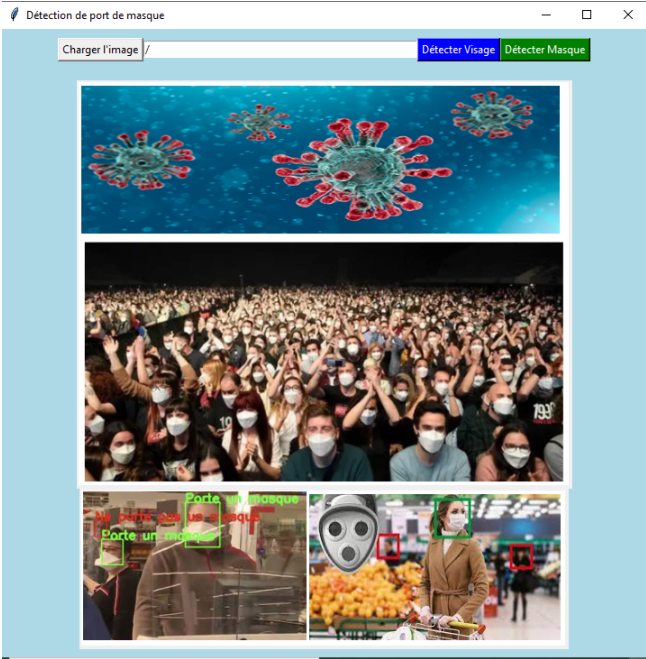


FIGURE 4.13 – L’interface graphique

La figure 4.14 illustre le chargement de l’image dans l’interface.



FIGURE 4.14 – chargement de l’image dans l’interface

La figure 4.15 illustre la détection de visages.

nombre de visage:12



FIGURE 4.15 – Détection de visages

La figure 4.16 illustre la détection de masque



8 visages 2 visages 2 visages

FIGURE 4.16 – Résultat du modèle

7 Quelques tests sur des images dans les salles de cours

Les images suivantes (Figure 4.17, 4.18, 4.19, 4.20, 4.21, 4.22) montrent le test de notre modèle sur des images prises dans des salles de cours de l'université 8 Mai 1945 -Guelma-.

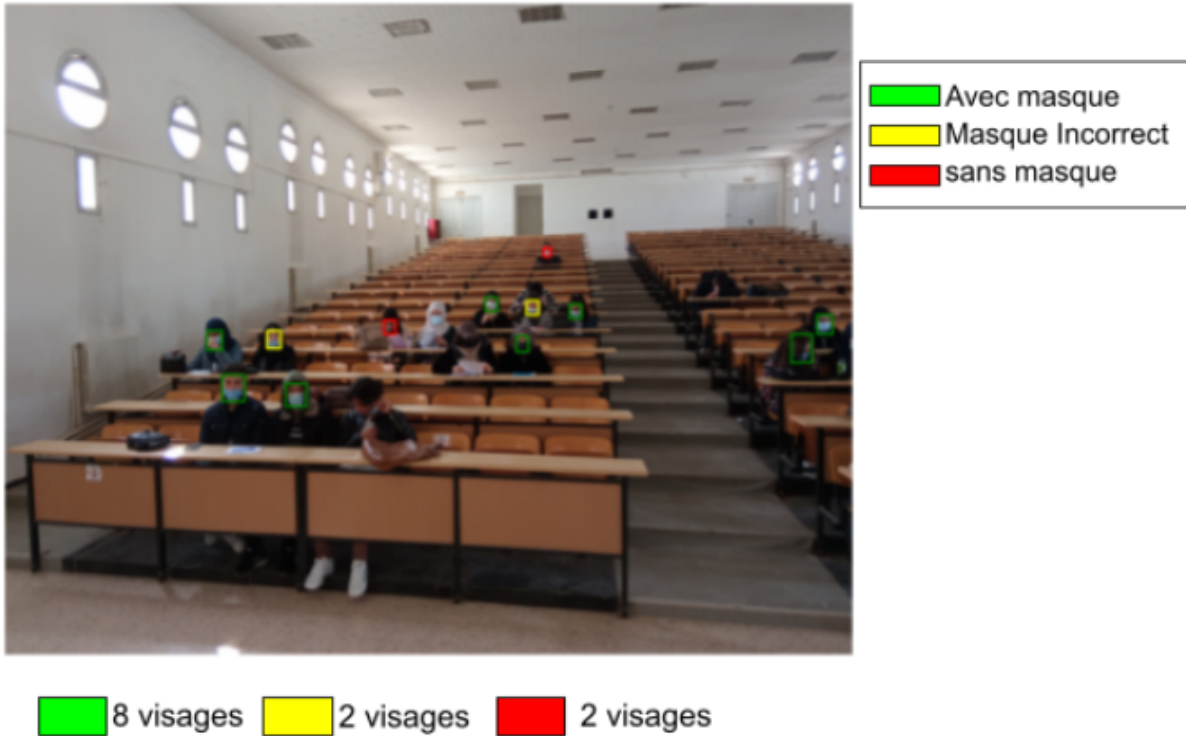
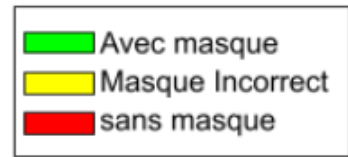
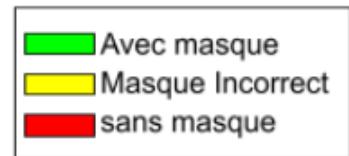
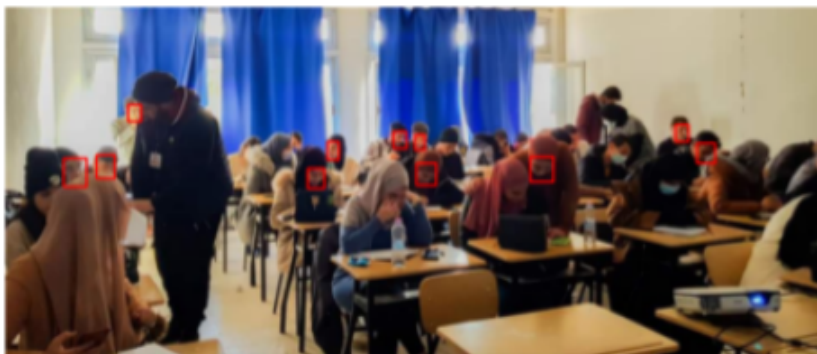


FIGURE 4.17 – Amphi théâtre n°6 de l'université 8 Mai 1945 -Guelma-



13 visages 0 visages 3 visages

FIGURE 4.18 – Amphi théâtre de l'université 8 Mai 1945 -Guelma-



0 visages 0 visages 11 visages

FIGURE 4.19 – Salle de TD de l'université 8 Mai 1945 -Guelma-



FIGURE 4.20 – Salle de TD de l’université 8 Mai 1945 -Guelma-



FIGURE 4.21 – Salle de TP de l’université 8 Mai 1945 -Guelma-



FIGURE 4.22 – Salle de TP de l’université 8 Mai 1945 -Guelma-

Les deux figure (4.23 et 4.24) montrent le test de notre modèle sur des images externe.

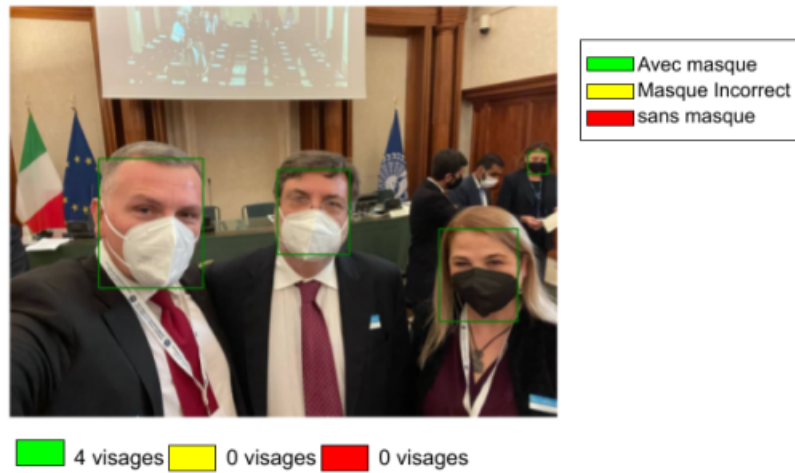


FIGURE 4.23 – salle de conférence externe



FIGURE 4.24 – salle de conférence externe

8 Conclusion

Les résultats de notre système sont très prometteurs dans le domaine la détection de port du masque. Les performances de ce système peuvent être optimisé en améliorant la qualité des données et donner le temps pour l'apprentissage. L'application permet de détecter les visage (avec masque, sans masque et masque incorrect), et présente une base solide pour un développement futur dans le domaine de détection de masque dans des vidéos, en temps réel ou même la reconnaissance des personnes portant du masques.

Conclusion Générale

Face à la transmission de la maladie du coronavirus (COVID-19), l'Organisation mondiale de la santé (OMS) a exigé le port du masque. Avant COVID-19, seules quelques personnes portaient des masques pour se protéger de la pollution de l'air, et les professionnels de la santé lorsqu'ils pratiquaient dans les hôpitaux. Avec la transmission rapide du virus, les scientifiques ont conseillé les masques dans les lieux publics pour empêcher la transmission de la maladie.

Notre projet consiste en la conception d'un système intelligent de détection du port de masque dans les salles de cours basé sur les réseaux à convolution profond (DCNN). Le système mis en œuvre est embarqué sur RaspBerry Pi 3. Il a été construit en affinant le modèle d'apprentissage en profondeur «Inception V3» entraîné sur la base Images 12K contenant plus de 12 000 images de trois classes de personnes : (0) avec des masques mal portés, (1) avec des masques portés correctement et (2) sans masques.

L'entraînement a abouti à un taux d'apprentissage de 97% et de test 97% avec une perte de 0.2% pour l'apprentissage et une perte de 0.2% pour le test.

Les tests ont été effectués sur des images prises dans les salles de cours (Amphithéâtres), les salles de travaux dirigés et les salles de travaux pratiques. Les résultats obtenus sont très prometteurs mais un affinement de l'apprentissage, surtout sur la classe des masques mal portés, permettra de réduire les erreurs de détection et de confusion de cette classe et par conséquent de booster la fiabilité de notre système.

Comme perspectives nous proposons le suivant :

- Extension de la base de la classe "masque incorrect".
- Utiliser inception V4.
- Utiliser Raspberry Pi 4.
- Combiner le réseau Inception V3 avec un autre réseau.

Bibliographie

- [1] Patricia Joly. L'élimination des masques réutilisables est un choix historique discutable. *Le Monde*, (25 mai 2020).
- [2] Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N. Taha, Nour Eldeen M. Khalifa d. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. *Measurement* 167, (2021).
- [3] G.Jignesh Chowdary, Narinder Singh Punn, Sanjay Kumar Sonbhadra, and Sonali Agarwal. Face Mask Detection Using Transfer Learning of InceptionV3, Check for updates.
- [4] Peishu Wu, Han Li, Nianyin Zeng, Fengping Li. FMD-Yolo : An efficient face mask detection method for COVID-19 prevention and control in public. *Image and Vision Computing* 117, (2022).
- [5] Akhil Kumar Arvind Kalia, Kinshuk Verma, Akashdeep Sharma, Manisha Kaushal. Scaling up face masks detection with YOLO on a novel dataset. *Optik - International Journal for Light and Electron Optics* 239, (2021).
- [6] B Varshini, HR Yogesh, Syed Danish Pasha, Maaz Suhail, V Madhumitha, Archana Sasi. IoT-Enabled smart doors for monitoring body temperature and face mask detection. *Global Transitions Proceedings* 2, (2021)
- [7] Shilpa Sethi, Mamta Kathuria a, Trilok Kaushik. Face mask detection using deep learning : An approach to reduce risk of Coronavirus spread. *Journal of Biomedical Informatics* 120, (2021).
- [8] Ahmad Alzu'bi, Firas Albalas, Tawfik AL-Hadhrami, Lojin Bani Younis and Amjad Bashayreh. Masked Face Recognition Using Deep Learning : A Review. *Electronics*, (2021).
- [9] Elliot Mbunge, Sakhile Simelane, Stephen G Fashoto, Boluwaji Akinnuwesi, Andile S

- Metfulaa. Application of deep learning and machine learning models to detect COVID-19 face masks - A review. *Sustainable Operations and Computers* 2, (2021). 235–245.
- [10] Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N. Taha, Nour Eldeen M. Khalifa. Fighting against COVID-19 : A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection. *Sustainable Cities and Society* 65, (2021).
- [11] Sandeep Gupta, S.V.N. Sreenivasu, Kuldeep Chouhan, Anurag Shrivastava, Bharti Sahu, Ravindra Manohar Potdar. Novel Face Mask Detection Technique using Machine Learning to control COVID'19 pandemic. *Materials Today : Proceedings*.
- [12] (Arne Wolfewicz)Un guide du débutant sur la façon dont les machines apprennent.
- [13] Isabelle Bellin et Arnaud Salomon Le deep learning : Une vieille histoire qui nous réserve des surprise. *Data Analytics Post (DAP)*, (22-JUIN-2017).
- [14] Afshine amidi et Shervine Amidi. *Pense-bete de réseaux de neurones convolutionnels, CS20-Deep Learning*.
- [15] Westerlund, M. (2009). The Emergence of Deepfake Technology : A Review. *Tecno-logy Innovation Management Review*, 9 (11), 39-52. <https://doi.org/10.22215/timereview/1282>.
- [16] A. Chaari. Nouvelle Approche D'identification Dans Les Bases de Données Barométriques Basée sur une Classification non Supervisée, thèse de doctorat, Université D'EVRY VAL D'ESSONNE, (soutenu le 06-OCTOBRE-2009).
- [17] Yann LeCun et al. Gradient-Based Learning Applied to Document Recognition. *Actes de l'IEEE*, (Novembre 1998).
- [18] Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton. Classification ImageNet avec des réseaux de neurones à convolution profonde. Université de Toronto, Canada. *NeurIPS*, 2012.
- [19] Karen Simonyan et Andrew Zisserman. Réseaux convolutifs très profonds pour la reconnaissance d'images à grande échelle. Université d'Oxford, Royaume-Uni. prépublication arXiv, (2014).
- [20] Christian Szegedy et al. Inception-v4, Inception-ResNet et l'impact des connexions résiduelles sur l'apprentissage. Actes de la trente et unième conférence AAAI sur l'intelligence artificielle, (2016).

- [21] Christian Szegedy et al. Apprentissage résiduel profond pour la reconnaissance d'images. Conférence IEEE sur la vision par ordinateur et la reconnaissance de formes (CVPR), (2016).
- [22] Rosebrock, Adrian. Deep Learning for Computer Vision with Python : starter Bundle. PyImageSearch, (2017).
- [23] Rose Gradilla. Réseaux convolutifs en cascade multitâches (MTCNN) pour la détection des visages et l'alignement des repères faciaux. Rose Gradilla, (28-juil-2020)
- [24] Raja Bharat. Un guide simple des versions du réseau Inception. towards data science, (29-Mai-2018)
- [25] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. CoRR, abs/1312.4400, 2013.
- [26] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi, Google Inc. Rethinking the Inception Architecture for Computer Vision. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence .
- [27] Van Rossum, Guido. "Python programming langage." USENIX annual technical conference. Vol.41.(2007).
- [28] Howse, Joseph. OpenCV computer vision with python. Packt Publishing Ltd, (2013).
- [29] Walt, Stéfan van der et al. "The NymPy array : a structure for efficient numerical computation." Computing in science engineering 13.2 (2011) : 22-30.
- [30] Ketkar, Nikhil. "Introduction to Keras." Deep learning with Python. Apress, Berkeley, CA, (2017). 97-111.
- [31] Grayson, John E. Python and Tkinter programming. Manning Publications CO. Greenwich, (2000).
- [32] Rossum, Guido. "Python reference manual."(1995).

Webographie

- [w1] Le masque à travers l'Histoire| War Heritage Institute. <https://www.warheritage.be/>. Accessed: 2022-02-07.
- [w2] Masque de protection FFP. <https://fr.wikipedia.org/>. Accessed: 2022-02-09.
- [w3] Masque en tissu. <https://fr.wikipedia.org/>. Accessed: 2022-02-09.
- [w4] Masque à Gaz. <https://fr.wikipedia.org/>. Accessed: 2022-02-09.
- [w5] Masque à Oxygène. <https://fr.wikipedia.org/>, Accessed: 2022-02-09.
- [w6] Maladie Covid-19 (nouveau coronavirus)|Institut Pasteur <https://www.pasteur.fr/>. Accessed: 2022-02-09.
- [w7] Une petite histoire du Machine Learning. <https://www.quantmetry.com/blog/une-petite-histoire-du-machine-learning>, Accessed: 2022-03-13.
- [w8] Une petite histoire du Machine Learning. <https://www.coe.int/fr/web/artificial/intelligence/glossary>, Accessed: 2022-03-13.
- [w9] Découvrez le neurone formel-Initiez-vous au Deep Learning-. <https://openclassrooms.com/fr/courses/>, Accessed : 2022-03-15
- [w10] DEEP LEARNING. <https://dataanalyticspost.com/Lexique/deep-learning/>, Accessed : 2022-03-15
- [w11] <https://levity.ai/blog/difference-machine-learning-deep-learning>
- [w12] <https://www.merkleinc.com/blog/dispelling-myths-deep-learning-vs-machine-learning>
- [w13] Quelques fonctions d'activation https://cs23in.stanford.edu/sildes/2017/cs23in_2017_lecture.pdf, Accessed : 2022-03-12

- [w14] <https://projeduc.github.io/introapprentissageautomatique/introductory.html>, Accessed : 2022-03-12
- [w15] <https://home.mis.u-picardie.fr/~furst/docs/Apprentissageartificiel.pdf>, Accessed : 2022-03-12
- [w16] <https://www.raspberrypi.org/downloads/raspbian/>
- [w17] <https://etcher.fr.uptodown.com/windows>
- [w18] Pandas <https://fr.wikipedia.org/wiki/Pandas>, Accessed : 2022-06-01
- [w19] Matplotlib.blog introduction to matplotlib using python for beginners. <https://www.analyticsvidhya.com/>, Accessed:2022-06-01.
- [20] <https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset>