الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

Mohamed Khider University – Biskra
Faculty of Science and Technology
Department: Electrical Engineering
Ref:……

جامعة محمد خيضر بسكرة

جامعة محمد خيضر بسكرة
كلية العلوم والتكنولوجـــيا
قسم: الهندسة الكهربائيـــة
المرجع:.....

Thesis presented to obtain the degree:

Doctorat LMD

Option :

Electronics

Entitled :

Deep Learning pour la Localisation et Détection pour l'Imagerie

Presented by:

MOKEDDEM MOHAMMED LAKHDAR

In front of the jury composed of:

Pr.DEBILOU AbdErrazak	Professor	President	University of Biskra
Pr.BELAHCENE Mebarka	Professor	Thesis director	University of Biskra
Pr.MALAAB Djamel	Professor	Examiner	University of Batna
Pr.CHARIF Fella	MCA	Examiner	University of Ouargla
Pr.OUAMANE AbdElMalik	Professor	Examiner	University of Biskra

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

Mohamed Khider University – Biskra
Faculty of Science and Technology
Department: Electrical Engineering
Ref:……

جامعة محمد خيضر بسكرة

جامعة محمد خيضر بسكرة
كلية العلوم والتكنولوجـــيا
قسم: الهندسة الكهربائيـــة
المرجع:.....

Thesis presented to obtain the degree:

Doctorat LMD

Option :

Electronics

Entitled :

Deep Learning for Localization and Detection for Imaging

Presented by:

MOKEDDEM MOHAMMED LAKHDAR

In front of the jury composed of:

Pr.DEBILOU AbdErrazak	Professor	President	University of Biskra
Pr.BELAHCENE Mebarka	Professor	Thesis director	University of Biskra
Pr.MALAAB Djamel	Professor	Examiner	University of Batna
Pr.CHARIF Fella	MCA	Examiner	University of Ouargla
Pr.OUAMANE AbdElMalik	Professor	Examiner	University of Biskra

Acknowledgments

It will be tough for me to thank everyone because I was able to complete my thesis owing to the assistance of many people.

First and foremost, I'd want to thank my thesis director **Pr. BELAHCENE Mebarka** for all her help. I am grateful to have worked with her since, in addition to her scientific assistance, she has always been available to advise and encourage me during the production of my thesis. Her comments and guidance enabled me to go forward and view my work from a different perspective.

I also thank **Pr. DEBILOU AbdErazak** for his presence, his time and his encouragement.

Pr. BELAHCENE Mebarka did me the honor of being my rapporteur and **Pr. DEBILOU AbdErazak** honoured me by his acceptance to chair the jury of my defence. For all this I thank them.

I also thank **Pr. CHARIF Fella** and **Pr. OUAMANE AbdElMalik and Pr. MALAAB Djamel** for the honor they do me to be in my thesis jury.

It is impossible for me to forget **Pr. BOURENNANE Salah** for his precious help for my bibliographical research. He always went out of his way to help me.

I thank all the people with whom I shared my studies and especially these years of thesis.

I would especially like to thank my family: my father, my mother, my brothers and my fiancee as a whole for their continuous support and understanding when undertaking my research and writing my project. I also thank my close friends.

Publications and conferences

Papers:

[1] Mokeddem Mohammed Lakhdar, Mebarka Belahcene, and Salah Bourennane. "COVID-19 risk reduce based YOLOv4-P6-FaceMask detector and DeepSORT tracker." Multimedia Tools and Applications (2022): 1-25. Doi: https://doi.org/10.1007/s11042-022-14251-7[2]

[2] Mokeddem Mohammed Lakhdar, Mebarka Belahcene, and Salah Bourennane. "Real-Time Social Distance Monitoring and Face Mask Detection Based Social-Scaled-YOLOv4, DeepSORT and DSFD&MobileNetv2 for COVID-19". International Journal Multimedia Tools and Applications, https://doi.org/10.1007/s11042-023-16614-0

International Conference

[3] Mokeddem Mohammed Lakhdar, Mebarka Belahcene, and Salah Bourennane, "Yolov4FaceMask: COVID-19 Mask Detector." 2021 1st International Conference On Cyber Management And Engineering (CyMaEn). IEEE, 2021. https://doi.org/10.1109/CyMaEn50288.2021.9497271

ملخص:

تتناول هذه الأطروحة مشكلة الكشف عن الأشياء وتحديد موقعها استنادًا إلى تقنيات التعلم العميق. يعد الكشف عن الأشياء وتحديد موقعها من المهام الأساسية في مجال الرؤية الحاسوبية، والتي يمكن تطبيقها في القيادة الذاتية والمراقبة والتصوير الطبي والروبوتات. يتضمن الكشف عن الأشياء التعرف على وجود الأشياء في صورة أو مقطع فيديو. على النقيض من ذلك، يتعلق التحديد بتحديد الإحداثيات المكانية الدقيقة لهذه العناصر، والتي يتم تصويرها عادةً بواسطة مربعات حدودية أو أقنعة تجزئة. أدت التطورات الأخيرة في التعلم العميق، وخاصة الشبكات العصبية التلافيفية (CNNS) والهندسة المعمارية القائمة على المحولات إلى تحسين دقة وكفاءة هذه المهام بشكل كبير. أصبحت تقنيات مثل شبكات CNN القائمة على المنطقة (R-CNN) و(YOLO) YOU ONLY LOOK ONCE و SINGLE SHOT MULTIBOX DETECTOR (SSD) من أحدث الطرق للكشف عن الأشياء وتحديد موقعها. تستفيد هذه الأساليب من مجموعات البيانات واسعة النطاق، مثل COCO وVOC PASCAL

في هذا العمل، نقترح نموذجين، الأول هو كاشف قناع وجه جديد عالي الأداء من مرحلة واحدة يعتمد على YOLOV4 يسمى كاشفFACE MASK.4YOLOV والثاني هو كاشف ومتعقب جديد عالي الأداء لقناع الوجه من مرحلتين بكاميرا أحادية وإطار عمل قائم على التعلم العميق لأتمتة مهمة توطين قناع الوجه واكتشافه وتتبعه باستخدام تسلسلات الفيديو. علاوة على ذلك، نقترح مجموعة بيانا ت جديدة للكشف عن قناع الوجه تتكون من 18000 صورة مع أكثر من 30000 مربع محكم وشروح لثلاثة تسميات فئات مختلفة وهي على التوالي: وجه مقنع / مقنع بشكل غير صحيح / بدون ملثمين. نحن نعتمد على نموذج الكشف عن الكائن SCALED-YOU ONLY LOOK ONE (SCALED-YOLOV4)لتدريب كاشفYOLOV4-P6- FACEMASKو التتبع البسيط عبر الإنترنت وفي الوقت الحقيقي باستخدام نهج عميق للارتباط (DEEPSORT) لتتبع الوجوه. نقترح استخدام DEEPSORTلتتبع الوجوه عن طريق تعيين المعرف لحفظ الوجوه مرة واحدة فقط وإنشاء قاعدة بيانات للوجوه غير المقنعة. YOLOV4-P6- FACEMASKهو نموذج ذو دقة عالية يحقق متوسط دقة يبلغ 93٪ ومتوسط استدعاء 92٪ و سرعة في الوقت الفعلي تبلغ 35 إطارًا في الثانية على بطاقة رسومات GPU TESLA-T4 واحدة في مجموعة البيانات المقترحة لدينا. لإثبات أداء النموذج المقترح، قمنا بمقارنة نتائج الكشف والتتبع مع أحدث النماذج الشائعة الأخر ىلاكتشاف قناع الوجه وتتبعه.

الكلمات المفتاحية : *تحديد الصور; تعلم عميق; الكشف; الموقع; الواصفات; تحسين.*

Abstract

This thesis deals with the problem of detection and localization of objects based on deep Learning techniques. Object detection and localization are essential tasks in computer vision, applicable in autonomous driving, surveillance, medical imaging, and robotics. Object detection entails recognizing the existence of objects in an image or video. In contrast, localization pertains to ascertaining the exact spatial coordinates of these items, usually depicted by bounding boxes or segmentation masks. Recent advancements in deep learning, particularly convolutional neural networks (CNNs) and transformer-based architectures have significantly improved the accuracy and efficiency of these tasks. Techniques such as Region-based CNNs (R-CNN), You Only Look Once (YOLO) and Single Shot MultiBox Detector (SSD have become state-of-the-art methods for object detection and localization. These approaches leverage large-scale datasets, such as COCO and Pascal VOC.

In this work, we propose two models, the first one is a new high-performance one-stage face mask detector based on YOLOv4 called the Yolov4FaceMask detector. The second is a novel high-performance two-stage face mask detector and tracker with a monocular camera and a deep learning framework for automating face mask localization, detection, and tracking utilising video sequences. Furthermore, we offer a new face mask detection dataset comprised of 18000 pictures with over 30000 tight bounding boxes and annotations for three alternative class labels: face masked/incorrectly masked/no masked. To train the YOLOv4-P6-FaceMask detector, we used the Scaled-You Only Look Once (Scaled-YOLOv4) object detection model and the Simple Online and Real-time Tracking with a Deep Association Metric (DeepSORT) strategy to track faces. DeepSORT is recommended for tracking faces by ID assignment to save faces just once and construct a database of non-masked faces. On our dataset of face masks, the YOLOv4-P6-FaceMask model achieves 93% mean average precision, 92% mean average recall and a real-time speed of 35 frames per second on one Tesla-T4 graphic card. To show the proposed model's performance, we compare the identification and tracking results with those of other popular state-of-the-art face mask detection and tracking models.

Keywords: Identification; Localization; Deep Learning; Detection; Descriptors; Optimization.

Résumé

Cette thèse traite du problème de détection et de localisation d'objets basé sur des techniques d'apprentissage profond. La détection et la localisation d'objets sont des tâches essentielles en vision par ordinateur, applicables à la conduite autonome, à la surveillance, à l'imagerie médicale et à la robotique. La détection d'objets implique la reconnaissance de l'existence d'objets dans une image ou une vidéo. En revanche, la localisation consiste à déterminer les coordonnées spatiales exactes de ces éléments, généralement représentées par des cadres de délimitation ou des masques de segmentation. Les progrès récents dans l'apprentissage profond, en particulier les réseaux de neurones convolutifs et les architectures basées sur des transformateurs, ont considérablement amélioré la précision et l'efficacité de ces tâches. Des techniques telles que les CNN régionaux, YOLO et SSD sont devenues des méthodes de pointe pour la détection et la localisation d'objets. Ces approches exploitent des ensembles de données à grande échelle, tels que COCO et VOC.

Dans ce travail, nous proposons deux modèles, le premier est un nouveau détecteur de masque facial à une étape haute performance basé sur YOLOv4 appelé le détecteur Yolov4FaceMask. Le second est un nouveau détecteur et traqueur de masque facial à deux étages hautes performances avec une caméra monoculaire et un cadre basé sur l'apprentissage en profondeur pour automatiser la tâche de localisation, de détection et de suivi du masque facial à l'aide de séquences vidéo. De plus, nous proposons un nouvel ensemble de données de détection de masque facial composé de 18 000 images avec plus de 30 000 cadres de délimitation serrés et des annotations pour trois étiquettes de classe différentes, à savoir : visage masqué/incorrectement masqué/non masqué. Nous nous basons sur le modèle de détection d'objets Scaled-YOLOv4 pour former le détecteur YOLOv4-P6-FaceMask et le suivi simple en ligne et en temps réel avec une approche de métrique d'association profonde pour le suivi des visages. Nous suggérons d'utiliser DeepSORT pour suivre les visages par attribution d'ID pour enregistrer les visages une seule fois et créer une base de données de visages non masqués. YOLOv4-P6-FaceMask est un modèle de haute précision qui atteint une précision moyenne moyenne de 93 %, un rappel moyen moyen de 92 % et une vitesse en temps réel de 35 ips sur une seule carte graphique GPU Tesla-T4 sur notre ensemble de données proposé. Pour démontrer les performances du modèle proposé, nous comparons les résultats de détection et de suivi avec d'autres modèles de pointe populaires de détection et de suivi des masques faciaux.

Mots clés: *Identification ;apprentissage profond;Détection;Localisation;escripteurs ; ptimisation.*

TABLE OF CONTENTS

FIGURES LIST

TABLES LIST

ACRONYMES

DL : Deep Learning

TL : Transfer Learning

ML : Machine Learning

YOLO : You Only look Once

CNN: convolutional neural network

Tanh: Hyperbolic tangent.

SSD: Single-Shot Detector

KNN: K-Nearest Neighbors.

SVM: Support Vector Machine.

MLP: Multi-Layer Perceptron.

LSTM: Long Short Term Memory.

GENERAL INTRODUCTION

1. Background

Nowadays, object detection and localization in indoor and outdoor scenes using deep learning are considered among the current challenging research topics in image processing and computer vision. It is an essential technology in several fields, including maintaining the general safety of the population against diseases and epidemics [93]. World Health Organization (WHO) reports noted that the disease COVID-19 2019 has infected more than 58 million people worldwide and caused more than 1.4 million deaths (April 9, 2021). With this COVID-19 coronavirus outbreak, many countries, or we can say all nations, have been forced to put in place new social distancing and face mask wearing rules. Governments have forced hospitals and different organizations to use new infection interference measures to prevent the spread of COVID-19 as its transmission rate increases [146]. However, the rate of transmission could vary depending on government measures and policies. As COVID-19 spreads through airdrops and close contact, governments have started using new rules requiring individuals to prevent people from sitting next to each other and wearing face masks to reduce the rate. of transmission and propagation. New variants of the coronavirus have taken hold after the relaxation of many countries in respecting safety rules (India, Nigeria, United Kingdom, Brazil…), which has prompted the WHO to recommend the use of equipment personal protective equipment (PPE) between people and in medical care. The coronavirus (COVID-19) spreads rapidly in close contact and in crowded environments. The spread of COVID-19 has affected people's lives and disrupted the economy. It has been classified as an important economic and public health problem. Countries need guidance and monitoring of people in crowded environments and incredibly crowded public spaces to ensure laws on wearing face masks are enforced. This could be used through CCTV systems and deep learning (DL) models. However, most of the mask detection apps and current research on mask detection models aim to solve the problem of masked and unmasked face, but ignore the incorrect wearing of face mask.

In this work, we are particularly interested in the localization and detection of faces as well as their follow-up in order to ensure the tracking of contaminated and at-risk people.

2. Problematic

Among the most common uses of images is identification in biometrics which is a field of artificial vision and which has experienced growing interest in recent years. Several interesting approaches have been developed in the fields of detection, localization and tracking.

The effectiveness of localization and identification techniques in imaging is today very strongly linked to strong constraints imposed on the user, a current research path is therefore turning towards the management of situations where data acquisition is less constrained. Finally, the use of classical methods is often limited in terms of performance or difficulties of use, which is why it seems interesting to evaluate the contribution of deep learning and artificial intelligence models in this field. context.

3. Positionnement

Our study, conducted at the Identification, Command, Control and Communication Laboratory (LI3C) of Mohamed KHIDER BISKRA University, is part of one of the issues studied by the RB_IAIM team " Biometric Recognition & Identification of Anomalies on Medical Imaging This work contributes to the work carried out in the broader sense by the team on research linked to two parallel axes:

a. Biometric identification/authentication of individuals;
b. Recognition of kinship by biometrics;
c. Identification of abnormalities in medical images;
d. Indoor/Outdoor location;
e. Optimization;
f. Artificial Intelligence-Machine Learning.

4. Objectifs

The main objective of the thesis is to contribute to a work to pursue research directed at the same time towards the technique and the uses.

The overall objective of the work is therefore to carry out the localization and detection of faces, that is to say: find where the object is and draw a bounding box around it, and then follow up (tracking) of these faces based on automatic image learning making our approach more flexible, fast and efficient. Afterwards unmasked and incorrectly masked faces are recorded to create an individual risk data set. This dataset is intended for identification/authentication and other applications.

The study and research relating to this thesis focuses on three axes, with a view to achieving a robust, efficient and effective localization and recognition system:

• The first part concerns the study and design of detection methods;

• The second part concerns the study of localization methods. Along this work the computation time is taken into account;

• On the other hand, artificial intelligence has proven to be most effective in computer vision tasks due to its convolution-based architecture. Since the advent of deep learning, face recognition technologies have steadily increased and had a substantial increase in accuracy. This motivated our research to use a location system based on Deep Learning.

The validation of the results is carried out on various indoor and outdoor data and sequences in controlled and uncontrolled environments.

5. Contributions

In this thesis work, the first contribution is the development of structures of the detection model YoloV4FaceMask and the proposal of a new database of masked faces.

Another contribution is the development of a biometric identification system, essentially based on another detection model under the name YOLOv4-P6-FaceMask and the DeepSORT tracker.

As well as their application on still images and sequences in controlled and uncontrolled environments to see if these approaches retain their performance and optimality.

The third contribution is the development and elaboration of a feature optimization structure based on the Deep Learning and Transfer Learning structure. For this, structures based on transfer learning is used based on convolutional neural networks (CNN) in order to extract the most relevant facial features.

6. Organisation of the Thesis

The thesis manuscript is structured around four chapters: after having introduced localization and facial recognition as well as the techniques used, chapter 2 is dedicated to the state of the art of scientific research in the field where the methods and techniques on Machine Learning and Deep Learning as well as the use of machine learning and classification are all reviewed. chapter 3 present deep detection and localization. Then, chapter 4 presents the design and implementation of the first proposed approach and its validation on the proposed databases. Chapter 5 is devoted to the second proposed approach and use of Deep Learning and all the work carried out on Google Colab. In all the chapters the presented results are discussed and improved along the work for better performance rates and also a good test time. Finally, a general conclusion and perspectives close the thesis.

Chapter 1

Generalities about localization and detection

Introduction:

Throughout history, humans have consistently required the ability to determine the location of objects and position themselves within their surroundings. In order to fulfil this need, many methodologies have been employed. In the early stages of human civilization, individuals relied on stones or mountains as navigational aids. The distinctive features of the terrain acted as a navigational reference point for him to navigate across the forest and the deserts. The fundamental principle for any localization is the "reference". All subsequent localization systems are built on this concept in order to determine the position of an object or a person. This chapter aims to review fundamental ideas and terminology in facial biometrics. It will begin by providing a generic definition of biometrics and then explain the rationale behind selecting face recognition as a specific focus. We will then address the issues of face recognition, face detection and localization [1].

1.1 What is biometrics?

There are two types of biometric systems: identification systems (recognition) and verification systems (authentication). see *Figure 1.1*.

Identification is a sort of application for which the system must answer the question: Who am I? The system must locate the identity of a person among those in a database containing persons previously enrolled and return the identity corresponding to the person appearing before the system, or the "unknown" identity if this person does not is not part of the base. This is a [**1 to n**] comparison where n is the number of persons in the database, commonly known as the gallery. Among the potential uses of a system in identification mode, we find the search for hazardous persons or the limited entry to a building of a corporation to its only workers for example.

In authentication, the system must answer the question: Am I the person I claim to be? The use case involves a person identifying themselves to the system, and the system then having to confirm that the person is who they say they are. Authentication-related applications include access to secure data, computing resources, and secure transactions. A biometric identification system's standard performance is measured using the following variables:

1. ***Rank-one Recognition Rate:*** It measures the percentage of entries that are correctly identified.

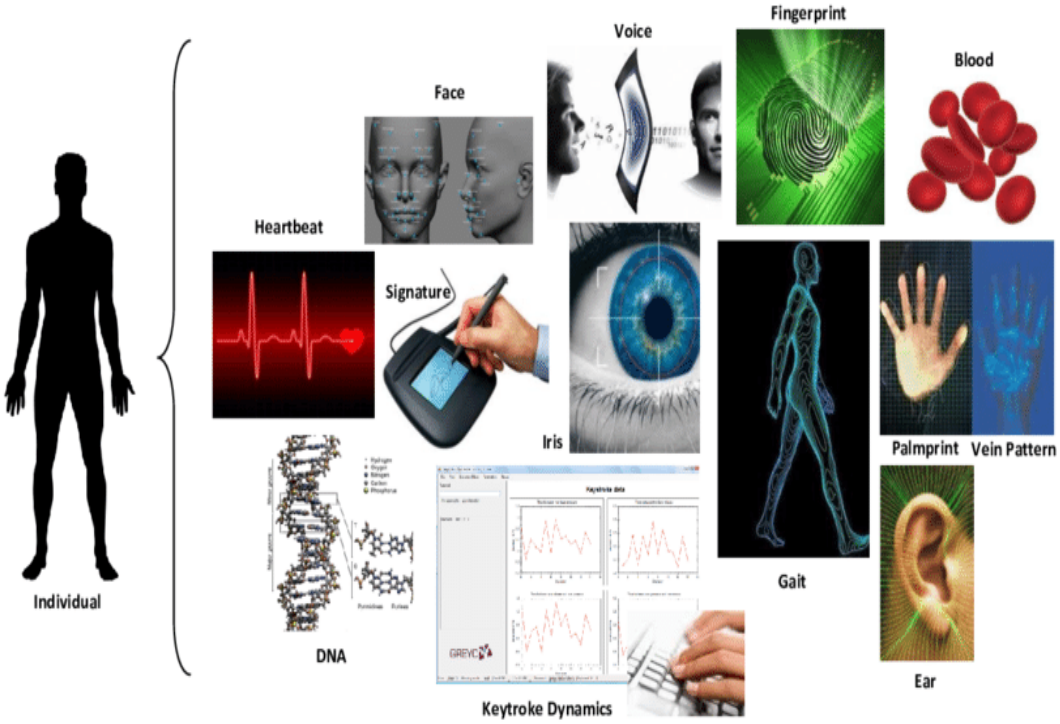


Figure 1. 1: *Some biometric modalities [2]*

2. ***Cumulative-Match-Characteristic (C-M-C):*** The CMC curve gives the percentage of people recognized according to a variable called the rank [3]. When a system selects the closest image as the outcome of the recognition, we say that it recognizes at rank 1. When a system selects the image that most closely resembles the input image out of two, for example, we say that the system recognizes at rank 2. The higher the rank, the greater the matching recognition rate is associated with a low level of security, we may thus conclude.

The following parameters are used for standard performance measurement of a biometric system with a verification scenario.

a. ***False-Reject-Rate (T-F-R) or False-Reject-Rate (F-R-R):*** This rate reflects the proportion of applicants who should be accepted but are turned down by the system.
b. ***False-Acceptance-Rate (T-F-A) or False-Accept-Rate (F-A-R):*** this rate represents the percentage of people who are not supposed to be recognized but who are still accepted by the system.
c. ***Equal-Error-Rate (T-E-E) or Equal-Error-Rate (E-E-R):*** This rate serves as a standard performance measuring point and is derived from the first two criteria. The intersection of TFR and TFA, or the ideal middle ground between false rejections and false accepts, is at this location.

d. ***Receiver-Operating-Characteristic (R-O-C):*** The R-O-C arc is a visual demonstration of the trade-off between TFAs and TFR related to a variable threshold.

As highlighted in the "Handbook of Biometrics" [4], A fast developing area, biometrics has several uses, including securing computer access and gaining entry into a country. For more information, one can refer to the manuals [4, 5].

1.1.1 Why face recognition?

Who is most qualified for giving this information is an evident worry given the demand to identify people [6]? The fingerprint, iris, face, voice, and signature among all the available biometric traits have received the greatest interest. Particularly, the methodologies for face, iris, and fingerprint identification have gradually acquired acceptance as standard biometric recognition technologies. Although iris and fingerprint identification technologies can yield reliable results in some specialised commercial applications, they have the apparent limitations described below [7]:

- **Physically intrusive:** While iris needs the user to position the eye in relation to the sensor, fingerprint demands the user's participation to make physical contact with the sensor surface. Furthermore, these kind of cooperative analysis techniques also require the user to pause for a second to “declare” themselves [6].
- **Socially intrusive:** people cannot recognize other people using this type of data, these types of identification have no place in normal human interactions and social structures [6]. To build a store that recognizes its best customers, or an information kiosk that remembers you, or a home that knows the people who live there, video face recognition and voice recognition have a natural place in these next generation intelligent environments [4]. In particular, they must be:
    - ***Natural and non-intrusive:*** they are discreet (able to recognize from a distance) and generally passive (do not require the generation of special electromagnetic lighting). They should not restrict the user's movement and should be low power and inexpensive [6].
    - ***Biological perception:*** This is perhaps the most important. However, People who can recognise others by their voice and face are consequently likely to feel at ease using face and voice recognition technologies [6].

As a strong proof of the ICAO organization (International-Civil-Aviation-Organization), Hietmeyer [8] pointed out that biometric identification can enable fast and secure processing of air passengers.

To select a single biometric feature for use in computer-aided identity confirmation, he suggested evaluating the compatibility of six biometric traits: face, fingerprint, hand geometry, voice, eyes as well signature on the basis of an MRTD (Machine-Readable-Travel-Documents) system. The compatibility score involves: enrollment factors, data renewal, machine-assisted identity verification requirements, redundancy, public perception, storage requirements and performance. As shown in *figure 1.2*, the face recognition system scored the highest compatibility and is becoming the biometric most likely to be selected for international use.

1.1.2 2DP Face Detection and Localization Difficulties

Many facial properties and the conditions in which they were photographed make automatic processing difficult. In the context of recognition, the main underlying problem is the intra-class variance, i.e. the variability that the face of the same person can take on because of differences in luminosity, pose. . . This intra-class variation can be greater than the inter-class variance, i.e. the variability that the faces of different people take on. In many systems, this intra-class variation is considered noise (unwanted information) making the goal of recognition more difficult.

The extraction of discriminating characteristics is indeed made more complicated and the overall performance of the systems is reduced [7]. We detail here the main difficulties encountered by a 2D automatic facial recognition system in real conditions.

1. **Lighting:** The appearance of a face can vary significantly depending on the lighting. Global (or ambient) illumination and local illumination both have an impact on this. While local lighting produces shadows and highlights in a non-linear way, global illumination impacts the entire face uniformly (or almost so). Figure 1.2 shows an illustration of a face with a moving light source. Numerous solutions have been put up to address these brightness issues. It is possible to obtain implicit modelling of brightness while making a face model. Another strategy that is extensively discussed in the literature is the extraction of characteristics that are unaffected by variations in brightness. Finally, take notice that a few methods address the issue of brightness prior to recognition by performing a pre-processing phase whose primary goal is frequently to repair artefacts brought on by fluctuations in luminosity.

Figure 1. 2 : *faces (examples) of the same person undergoing a change in brightness*

(Images is collected from the internet).

2. **Pose :** The pose of a face defines the rotation that a face may have undergone during the capture. Pose variations can be of two types depending on the type of rotation: in-plane rotation where the axis of rotation is the camera axis, and out-of-plane rotation otherwise. *Figure 1.3* shows an example of a face undergoing out-of-plane rotation. Pose variations greatly affect automatic face recognition systems, which is why many of them are limited to frontal poses or too specific poses requiring prior estimation. In the case of a rotation in the plane, the appearance of the face is not deformed and a good estimate of the angle of rotation can be enough to recalibrate the image by simple reverse rotation and thus obtain a frontal pose (forehead top of image, chin down). The case of out-of-plane rotation is often much more complex, unless the faces used for enrollment and recognition have the same pose.

Figure 1. 3 : *Example of a face of the same person undergoing out-of-plane pose variations (Image collected from internet).*

3. **Facial-expressions:** A face looks very different when facial emotions are present. (*figure 1.4*). When this happens, mouths and even the eyes might experience severe deformations, which can prevent a face recognition system based on, say, areas of interest from working properly. (these can thus undergo significant translations). The mouth is generally the facial element that varies the most, but the appearance of the eyebrows, for example, can be greatly modified [9].

Figure 1. 4 : *Intra-class variability due to the presence of facial expressions(image collected from Internet).*

4. **Occlusions:** As seen in figure 1.5, partial occlusions regularly happen in practical applications. They can be brought on by long hair, eyeglasses, sunglasses, any other object (scarf, etc.), concealing hands, or even by another person. In other cases, such as when rotating out of plane, one aspect of the face may obscure another.

Figure 1. 5 *facial-occlusion (examples of image collected from internet).*

1.2 What is Object-Recognition?

One of numerous related computer vision tasks that are generally referred to as "object-recognition" is the identification of objects in digital images. Image classification is the process of determining the class of a particular item inside an image. Object-localization is the process of identifying one or more objects in a picture and drawing a bounding box around their extent. Object detection, which finds and classifies one or more objects in a picture, combines these two tasks. (*figure 1.6*).

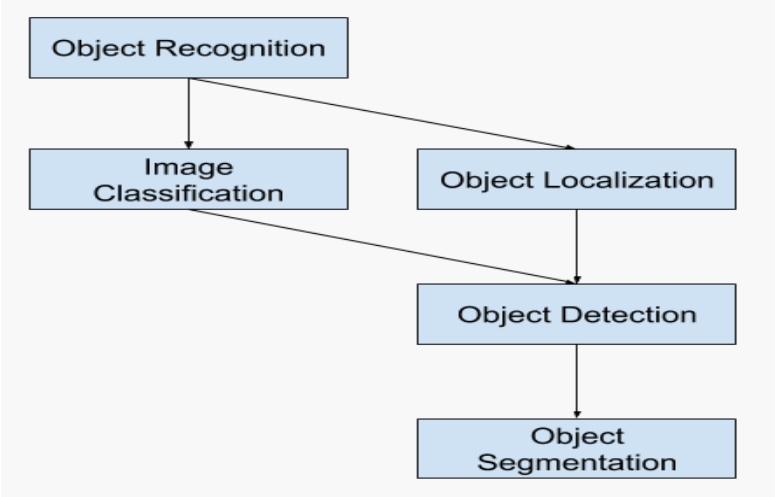


Figure 1. 6 : *Tasks of Object Recognition*

1.3 Object Localization and detection-based Machine-Learning

- **Definitions**

Robotics, machine-learning, and natural language processing are just a few of the many subfields that fall under the broad umbrella of artificial-intelligence (AI) (see figure 1.7). It is strongly related to computer science as well. AI makes it possible to build intelligent-computers that can act and think like people and make decisions on their own. The main goal of AI research is to create computers that can reason, learn, and solve problems like humans do. AI applications can be used to process and make decisions on a variety of tasks, including but not limited to: Autonomous Vehicles, Object Detection and Localization, Fraud Detection, Predicting Consumer Behavior, Robotics, Speech Recognition, Translation.

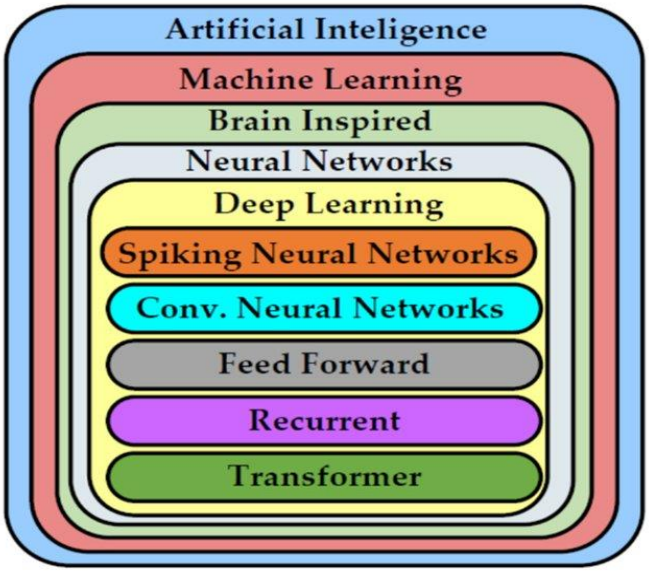


Figure 1. 7 : *AI taxonomy and related fields [10]*

- **Machine-Learning**

A branch of artificial-intelligence called machine-learning (ML) is concerned with creating algorithms that can learn from data and make predictions. Machine learning's fundamental objective is to automatically spot patterns in data and use those patterns to infer future events or make judgements without having to be explicitly programmed to do so. supervised, unsupervised, semi-supervised, and reinforcement learning are common categories for ML algorithms. Based on the learning-algorithm, they are divided.

- ***Supervised-learning:***

The instances in the training dataset are labelled, which is a feature of supervised learning. Typically, in classification issues, the labels are class labels. Induce models that may be used to categorise further unlabeled data by creating a function that maps inputs to desired outputs as the end aim of this form of learning.

- **Unsupervised-learning:**

Clustering and feature reduction are the key applications for unsupervised learning. The objective of this sort of learning is to uncover hidden patterns in the data using unlabelled inputs [11].

- **semi-supervised-learning**

Learning that is semi-supervised combines supervised and unsupervised learning and makes use of both labelled and unlabelled inputs [11].

- ***Reinforcement learning:***

In reinforcement learning, the algorithm selects an output for each data observation before receiving input from the environment. It is frequently used to issues involving sequential decision-making [12].

Artificial Neural Networks

(*Figure 1.8*). The technique attributed to the human brain is based on the fact that the network acquires knowledge from its surroundings through a learning process, and that the power between the connections of neurons is known as synaptic weight, which is where the gained knowledge is stored. The number of neurons and the type of activation function utilised determine the accuracy of an ANN. There is still no rule governing the number of layers that an ANN must have in order to work optimally or for the activation function; the only requirement is that the ANN have at least two layers [11].

An ANN's neurons are organised in multiple degrees of parallel organisation. These levels are classified as follows:

- **Input-layer:** This is the level number one of an ANN. At this level the data enters the ANN and the number of variables is equal to the neurons.
- **Hidden-layer:** The number of hidden layers may differ, and the number of neurons increases as the hidden levels increase.
- **Output-layer:** This is the final level of an ANN. At this level, the results are the outputs, and the number of neurons is equal to the possible output variables.

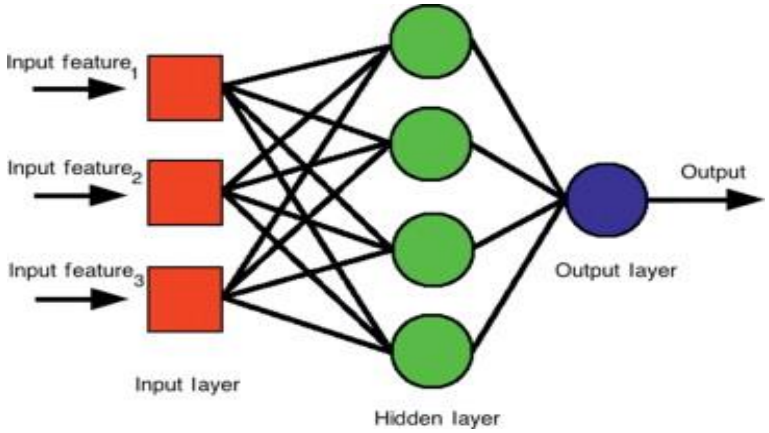


Figure 1. 8 : *Schematic representation of a multilevel ANN [13]*

1.4 Localization and Detection

Object localization and detection are very large and important fields in research, because current research aims to create systems that approach human skills in perception and object tracking and recognition. The importance of the localization and detection comes from the fact that the good result in these phases gives a good result of the recognition and also these phases is not easy to process because of several problems such as the size of the object (*Figure 1.9*) : the light, the shape, the wide variety of objects, the real-time response speed, the complexity of the backgrounds… etc.

One of the most intriguing real-world uses for object localization and detection is in traffic monitoring or for vehicles with automatic or partially autonomous driving assistance, as well as in the localization and detection of people who are not wearing masks during the "COVID-19" incident and in businesses to distinguish between well-made and poorly-made products (for example: company that produces parts so the system will check the quality)

Deep learning has been embraced and incorporated by researchers and businesses for various computer vision use cases as a result of the evaluation of artificial intelligence technologies. One such use is object localization. By enclosing an object in a bounding box, object localization techniques locate the object in an image and determine its location. Object localization is one of the image recognition tasks along with image classification and object detection. Though object detection and object localization are sometimes used interchangeably, they are not the same. Similarly, image classification and image localization are also two distinct concepts.

1.4.1 Classification task

The process of classifying an image is assigning it to one of a set of labels or land cover categories. The objective of this project is to extract data from photos and classify it. (e.g., masked face/non masked face) or a probability (e.g., there is an 85% chance that this is a person).

- **Input:** An image with a single object, such as a photograph.
- **Output:** A class label (one or more integers that are mapped to class labels).

1.4.2 Localization task

Locate the presence of objects in an image and indicate their location (**x** and **y** coordinates) with a bounding box (draw bounding boxes).

- **Input:** an image
- **Output:** "**x**", "**y**", height, and width numbers around an object of interest

Figure 1. 9 : *Example of localization [14]*

1.4.3 Detection-task

Object-detection is a difficult problem that involves picture localization and classification. An object detection method would produce bounding boxes around all things of interest in a picture and assign them a class. (*figure 1.10*).

- **Input:** an image
- **Output:** "**x**", "**y**", height, and width numbers around all object of interest along with class(es).

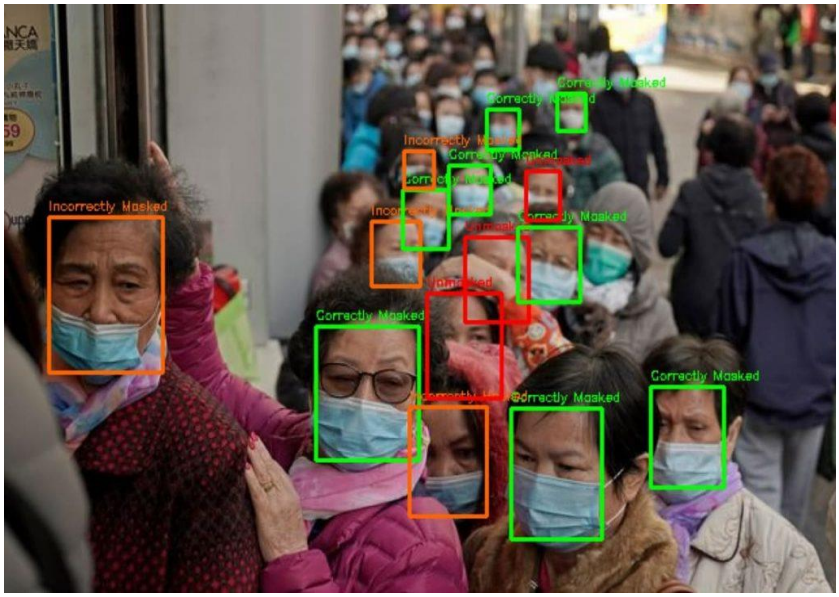


Figure 1. 10 : *Example of Object Detection (face mask detection)*

Conclusion

In this chapter, we have revisited and thoroughly explored the definitions of the four primary applications of biometrics: identification (also referred to as recognition), verification, detection, and localization. Each of these applications plays a critical role in various real-world scenarios, from security systems to user authentication. Additionally, we have revisited and discussed in detail the CNN , detection and localisation . CNN detection and localization refer to the processes by which Convolutional Neural Networks (CNNs) identify and precisely locate objects within an image or video. Detection involves recognizing the presence of objects and classifying them, while localization focuses on determining their exact spatial coordinates, often represented by bounding boxes or segmentation masks. Together, these tasks enable CNNs to not only identify what objects are present in a scene but also pinpoint where they are located, making them essential for applications such as autonomous driving, surveillance, and medical imaging.

Chapter 2 State-Of-The-Art

Introduction

The Viola Jones detector, which was utilised for real-time detection, marked the beginning of the growth of object detectors. Traditionally, object detection algorithms dealt with spatial structures using a structured classifier and hand-crafted features to extract pertinent information from photos.

These conventional methods, however, are unable to handle the many changes in object look and shape and effectively leverage the enormously large data volume. These algorithms have several limitations, even though they are unsupervised and do not require historical data for training. This is especially true when dealing with complex situations like the lighting effect, occlusion effect, and clutter impact. Deep Learning methods are the foundation of the modern era of object detection. In the case of computer vision, we can distinguish traditional machine learning from Deep Learning by saying that Machine Learning extracts hand-crafted features from images and performs classification, whereas Deep Learning techniques extract the features and classify them in a single step.

In this part of thesis, we review most of the recent work on the location and detection of people based on Deep Learning. First we will start with an introduction to the topic of object localization and detection itself and its key metrics.

2.1. Recent Research of Object Localization and Detection

The classification of images consists in systematically distributing images according to classes established beforehand, classifying an image makes it correspond to a class, thus marking its relationship with other images. In general, recognizing an image is an easy task for a human over the course of his existence, he has acquired knowledge that allows him to adapt to the variations resulting from different acquisition conditions. For example, it is relatively easy for him to recognize an object in several orientations partially hidden by another from near or far and according to various illuminations.

However, technological progress in terms of image acquisition (microscopes, cameras, sensors) and storage generate databases rich in information and multiply the fields of application, it then becomes difficult for humans to analyse the large number of images, the time required, the repetitive nature of the task and the concentration required are problematic. However, this is not necessarily easy for a computer program for which an image is a set of numerical values. The objective of image classification is to develop a system capable of automatically assigning a class to an image. Thus, this system makes it possible to carry out an expertise task which can prove costly to acquire for a human

being due in particular to physical constraints such as concentration, fatigue or the time required by a large volume of image data.

Automatic image categorization has a wide range of uses, from document analysis to medicine and the military. Thus, there are applications in the medical field such as cell and tumour recognition, handwriting recognition for checks and postal codes, in the urban domain such as road sign recognition, pedestrian recognition, vehicle detection, building recognition to aid in localization, and in biometrics such as face, fingerprint, and iris recognition.

All of these applications have one thing in common: they all require the construction of a processing chain from accessible pictures that consists of numerous steps in order to deliver a choice as output. Each phase of the development of such a classification system, namely the feature extraction and learning phases, necessitates the search for acceptable approaches for best overall performance. In most cases, we have picture data from which we must extract useful information in the form of digital vectors. This extraction process enables us to operate in a digital environment. It is thus necessary to create, during the learning phase, a decision function to determine if a new datum corresponds to one of the classes present.

2.2. Classifiers

ImageNet [15] is an image database organized according to the WordNet hierarchy contains 14197122 annotated images. The dataset has been used in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) since 2010, which is a benchmark in picture classification and object recognition. A collection of manually annotated training photos is included in the publicly available dataset.

A series of test photos is also made available, although the hand annotations are not included. In this part, we report the findings of cutting-edge classifiers trained on the ImageNet dataset.

The results of classification are illustrating in table1.1

- **LeNet[16]:** LeNet is the first applications that were successful of convolutional networks. This model developed by Yann-Le-Cun in the 1990. Of these, the best known is the LeNet architecture used to read postal codes, digits, etc.

Table 2.1 *Deep Learning classifiers results on ImageNet dataset*

Model Name	Accuracy
AlexNet	63.3
ZFNet	64
VGG-19	74.5
Inception V3	78.8
EfficientNet-B1	79.1
ResNeXt-101	80.9
EfficientNet-B3	81.6
NASNet-A	82.7
PNASNet-5	82.9
EfficientNet-B7	84.3
FixResNeXt-101	86.4
VAN-B6	87.8
Mixer-H/14	87.94
VITG/14	90.45
Model soups(BASIC-L)	90.98
COCA	91

- **AlexNet[17]**: The first work that popularized convolutional-networks in computer-vision was AlexNet, developed by Alex-Krizhevsky, Ilya-Sutskever and Geoff-Hinton. AlexNet was sended to the ImageNet-ILSVRC-challenge [18] in 2012 and clearly outperformed its competitors (*Figure 2. 1*). The network of this model had a very similar or the same architecture of LeNet model, but was deeper, larger, and had convolutional-layers stacked on top of each other (previously it was common to have only one convolutional-layer always immediately followed by a pooling-layer).

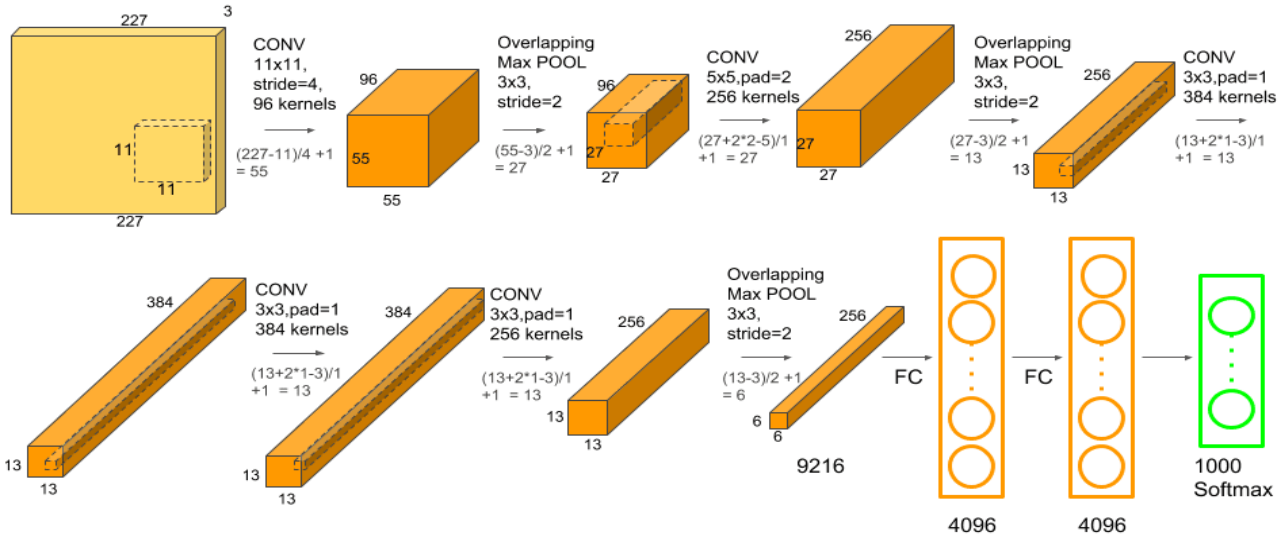


Figure 2. 2: *AlexNet Network architecture [17]*

- **ZFnet [19]:**

The winner of ILSVRC challenge 2013 was a convolutional-network by Matthew-Zeiler and Rob-Fergus. It became ZFNet (short-for-Zeiler-and-Fergus-Net). It was just an improvement of AlexNet model by adjusting the hyper-parameters of the architecture, in particular enlarging the size of the convolutional-layers and minimizing the kernel-size on the first-layer. (*Figure 2. 2*)

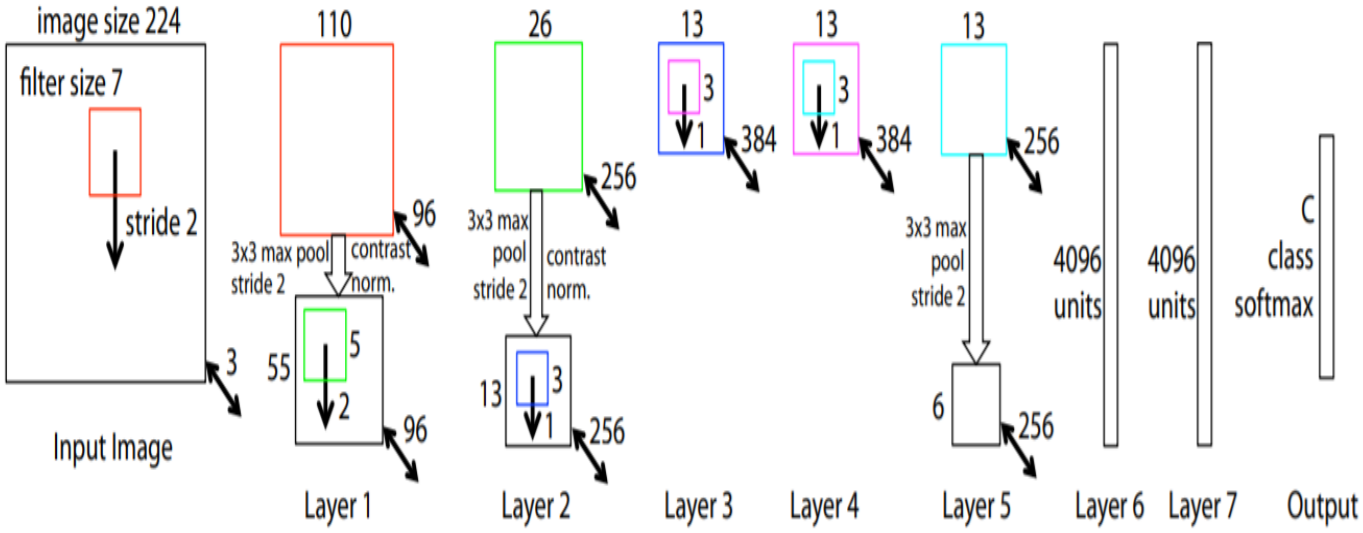


Figure 2. 2: *ZFNet Network architecture [19]*

- **GoogLeNet [20]:** The winner of ILSVRC challenge 2014 was a convolutional-network by Szegedy and other authors from Google. The main contribution of this work was the development

of an inception-module which significantly minimised the number of parameters in the network compared to AlexNet (4M, compared to AlexNet with 60M). Also, this module removes several parameters by using global AVG-pooling-instead of PMC at the network's end. There are numerous versions of GoogLeNet, including Inception-v4. [21].

- **ResNet [22]:** Residual network developed by Kaiming He et al. was the winner of ILSVRC 2015. It features connection jumps and strong use of batch normalization. It also uses global AVG pooling instead of PMC at the end. (*Figure 2. 3*)

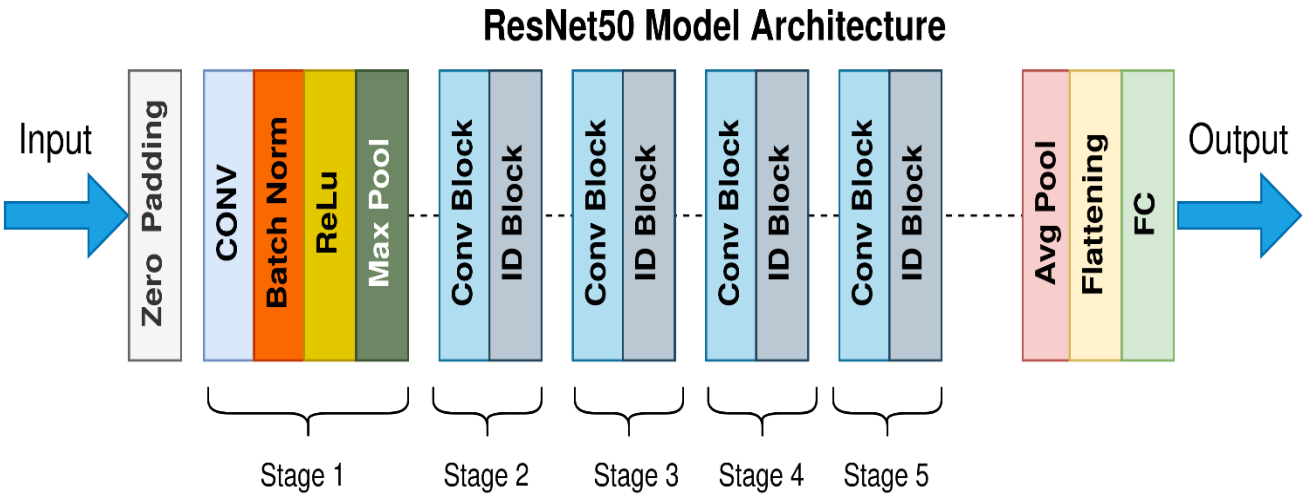


Figure 2. 3: *ResNet Network architecture [22]*

- **ResNeXtModel [23]:** The conventional approach to enhancing the accuracy of a model involves increasing its depth or width. However, such augmentation results in escalated model complexity and parameter count, while gain margins diminish rapidly. In response, Xie et al. introduced the ResNeXtModel architecture, which is simpler and more efficient than its predecessors. The ResNeXtModel was inspired by the stacking of comparable blocks in VGG_Net/ResNetModel and the "split transform-merge" behavior of the Inception module. Essentially, it is a ResNet-model with each ResNet block replaced by a ResNeXt-model that is similar to the inception-model. The intricate, customized transformation modules from the Inception model are substituted with topologically identical modules in the ResNeXtModel blocks, rendering the network more scalable and generalizable. The authors. also emphasize that the cardinality (topological paths in the ResNeXtModel block) can be regarded as a third dimension, alongside depth and width, to enhance model accuracy. The ResNeXtModel is elegant and more concise, achieving higher accuracy while having significantly fewer hyper-

parameters than similar depth ResNetModel architectures. It was also the first runner-up in the ILSVRC2016Challenge.

- **CSPNet-model** [24]**:** Existing neural-networks have shown incredible results in achieving high-accuracy in computer-vision tasks; however they rely on excessive computational resources. Wang and other authors. believe that heavy inference computations can be reduced by cutting-down the duplicate-gradient-information in the network. They proposed Cross-Stage-Partial-Network (CSPNet), which creates different paths for the gradient flow within the network. CSPNet separates feature maps at the base layer into two-parts. One part is passed through the partial convolution-network block (e.g., Dense and Transition block in DenseNet or Res(X) block in ResNeXt) while the other part is combined with its outputs at a later stage. This minimises the amount of parameters, improves processing unit utilisation, and reduces memory footprint. It is simple to implement and generic enough to be used on various architectures. like ResNet, ResNeXt, DenseNet, Scaled-YOLOv4 etc. Using CSPNet on these networks lowered calculations by 10% to 20% while maintaining or improving accuracy. This strategy dramatically reduces memory costs and processing bottlenecks. It is employed in various cutting-edge detector types, as well as for mobile and edge devices.

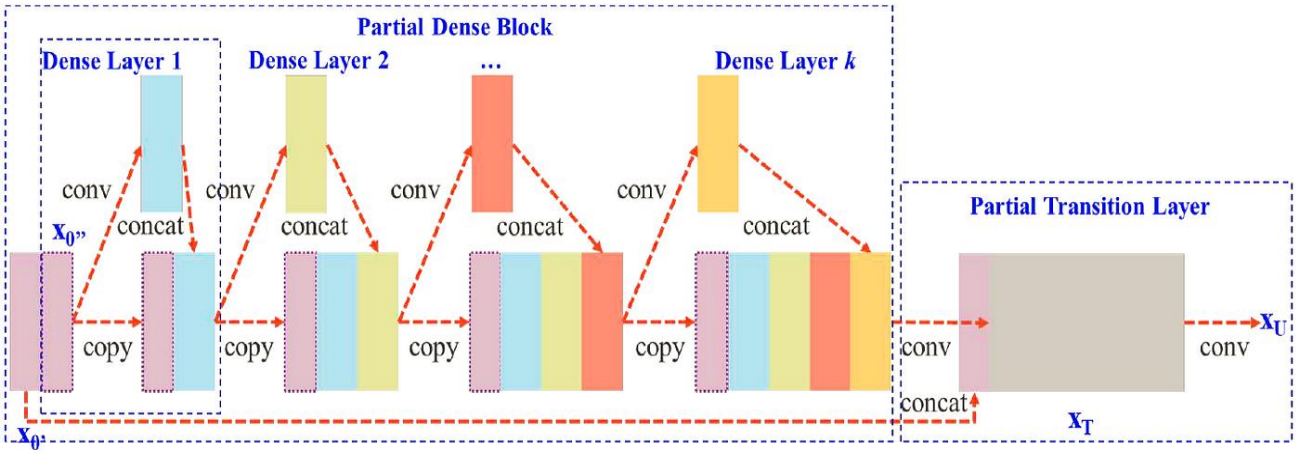


Figure 2. 4: *Exemple of CSPNet [24]*

- **EfficientNet** [25]: Tan and other authors. thoroughly investigated network-scalability and its implications on model-performance. They summarised how network-factors such as depth, breadth, and resolution affect accuracy. Any parameter that is scaled separately incurs a cost. Increased network-depth can aid in the capture of richer and more complex characteristics, but they are challenging to train owing to the vanishing gradient-problem. Similarly, increasing

network width makes it simpler to capture fine-grained information but makes capturing high-level characteristics more challenging. Gains from increased picture resolution, such as depth and breadth, become saturating as the model scales. In the paper [25], The authors recommended using a compound coefficient that can scale all three dimensions evenly. EfficientNet-model is a simple and efficient architecture. It beat prior models in terms of accuracy and speed despite being much smaller. It has the ability to usher in a new era in the field of efficient networks by giving a massive boost in efficiency.

2.3. Detectors

Detectors are classified into two types: one_stage detectors and two_stage detectors. A two-stage detector is a network that includes a separate module for generating area proposals. During the first step, these models attempt to find an arbitrary number of object suggestions in a picture, and then classify and localise them in the second. Because these systems have two distinct processes, they take longer to create suggestions, have more intricate design, and lack global context. which models are well-known:

- **Recurrent-Convolutional-Neural-Network (R-CNN)** [26] The first publication in the R-CNN family, r-cnn, illustrated how CNNs may be utilised to significantly increase detection performance. R-CNN employs a class-agnostic region proposal module in conjunction with CNNs to transform detection into a classification and localization challenge.
- **Fast-RCNN**[27] is One of the key drawbacks of R-CNN/SPPNet was the requirement to train different systems individually.
- **Faster-RCNN**[28], which starts with Region-Proposal-Network (RPN) to produce regions of interest, followed by categorization and bounding box regression.
- **FPN[29],** which Use of image pyramid to obtain feature-pyramid (or featurized-image-pyramids) at multiple levels is a common method to increase detection of small-objects.
- **R-FCN**: Region-based Fully-Convolutional-Network (R-FCN) [30] that shared almost all calculations within the network, unlike previous two stage detectors which applied resource intensive techniques on each proposal. They advocated against using fully connected layers in favour of convolutional-layers.

The second type of detector, known as one-stage detectors, classifies and localises semantic items in a single shot utilising dense sampling. Among the most well-known models are:

You-Only-Look-Once(YOLO) [31] is a model made-up of a single neural-network that can be trained end-to-end using back-propagation. It combines the preceding techniques' two processes, object detection and localisation, into a single model. YOLOv2 [32]. An upgrade on the YOLO [31], the YOLO9000 model could predict 9000 item types in real time and offered a simple balance between speed and accuracy. DarkNet-19 was used to replace GoogLeNet's backbone architecture. YOLOv3 [33]. The authors rebuilt the feature extractor network with a bigger Darknet-53 network in order to make "incremental improvements" over prior YOLO versions (YOLO, YOLOv2). YOLOv4 [34]. included a number of intriguing concepts to develop a quick and easy to train object detector that could function in existing production systems. **Scaled-YOLOv4 Scaling: Cross-Stage-Partial (CSP) [35]. Single-Shot-Multibox-Detector (SSD) [36],** was the first single stage detector that matched accuracy of contemporary two stage detectors like Faster R-CNN [44], while maintaining real time speed. SSD was built on VGG-16 [17], with additional auxiliary structures to improve performance. **RetinaNet [37]** Given the difference between the accuracies of single and two stage detectors,

Table 2.2 *Comparison of the speed and accuracy of detectors on the MS-COCO dataset (from coco dataset website)*

Model-Name	**Back_bone**	**M-ap**	**FPS**
Faster-RCNN	-	22.0	3.0
SSD	VGG- 16	27.0	10.0
YOLO v2	-	22.0	-
YOLO v3	Darknet53	33.0	31.0
YOLO v4	CSP-Darknet53	43.50	62.0
YOLO v4-CSP	CSPDarknet53s	47.50	97.0
Efficient DetD1	EfficientNetB1	40.50	74.0
Efficient DetD0	EfficientNetB0	34.60	97.0
A-S-F-F	Darknet -53	42.40	46.0
YOLO v3SPP	Darknet -53	42.90	73.0
YOLO v4P5	CSP -P5	51.80	43.0
YOLO v4P6	CSP-P6	54.5	32.0
YOLO v4-P7	CSP-P7	55.5	17.0

Lin et al. suggested that the reason single stage detectors lag is the “extreme foregroundbackground class imbalance” [67]. They proposed a reshaped cross entropy loss, called Focal loss as the means to remedy the imbalance. **EfficientDet [38].** develops towards the notion of scalable detector with improved accuracy and efficiency. It adds efficient multi-scale features, as well as BiFPN and model scaling. BiFPN is a bi-directional feature pyramid network with learnable weights for connecting input features at various sizes.

One_stage detectors use dense sampling to categorise and localise semantic items in a single shot. To localise objects, they employ predetermined boxes/keypoints of varying scale and aspect ratio. It outperforms two_stage detectors in terms of real-time performance and design simplicity. YOLO is the most popular one_stage model.; *table 2.2* shows the comparison between members of the detection models and their performance. The evaluation of illustrated models was usually based on COCO(Microsoft-Common-Objects-in-Context)dataset [39].

2.4. Recent face detectors

The topic of face recognition (FR) has been a prominent and widely discussed subject in the field of computer vision. With the advent of deep learning techniques and the availability of large-scale datasets, deep face recognition has made significant strides and is now extensively employed in various real-world applications. The initial step in the face recognition process is face detection, which involves identifying all the faces present in an input image and providing their bounding box coordinates along with a confidence score. To provide a comprehensive classification of deep face detection methods, the author of [40] has categorized them into seven distinct groups.

2.4.1. Multi-stage methods

A multi-level detector creates many suggestions and then refines them by one or more further levels using a coarse-to-fine technique or a proposal-to-refinement strategy [40]. The initial step proposes a certain scale of potential bounding boxes using a sliding window, while subsequent phases eliminate false positives and refine the remaining boxes. In this case, the cascaded architecture [41, 42, 43] is certainly an effective solution for coarse-to-fine face recognition. Face recognition can be viewed as a specific goal of general object recognition. Therefore, many works [44, 45, 46] inherit the remarkable achievements of general object detectors. For example, Faster-RCNN, a classic and effective detection framework, uses a region proposal network to generate region proposals with a series of dense anchor boxes in the first stage, and then refines the proposals in the second stage. Based on the proposal-to-refine scheme, a lot of work has been devoted to improving the modeling of the refinement stage [47,

48, 49] and the proposal stage [50, 51], and great progress has been made in accurate face recognition. Besides modeling, training multi-class detectors is another interesting topic. To address the poor optimization of MultiStage detectors, a general training strategy [52] is developed for CascadeCNN [41] and FasterRCNN to achieve end-to-end optimization and better-performance.

2.4.2. One- stage method

The one-step approach is capable of performing candidate classification and bounding box regression directly from feature maps, without relying on the proposal stage. This approach is derived from a general-purpose object detector known as the Single Shot Multi-Box Detector (SSD). While maintaining the same structure as SSD, the one-step approach achieves faster processing speeds compared to multi-stage methods, while still maintaining comparable accuracy. Several studies (53, 54, 55) have developed deep face detectors based on SSD that are robust to different scales of faces.

In terms of the backbone architecture, many face detectors utilize the Feature Pyramid Network (FPN) (29), which consists of a top-down architecture with skip connections. FPN merges high-level and low-level features to enhance detection. The high-level feature maps provide rich semantic information, while the low-level layers contribute more local information. The fusion of these features combines the advantages of both sides and significantly improves the detection of objects with a wide range of scales. Consequently, many single-stage face detectors (56, 57, 58, 54) have been developed to leverage the benefits of FPN. These methods not only address the scale issue in face detection using FPN but also attempt to overcome the inherent limitations of the original FPN, such as the conflict of receptive fields.

Despite the high efficiency of single-stage methods, their detection accuracy is lower than that of two-stage methods. This is partially due to the imbalance problem between positive and negative samples caused by the dense anchors. The proposal-to-refine scheme is able to alleviate this issue. As a result, RefineDet (59) introduces an anchor refinement mechanism to improve the accuracy of single-stage methods.

The network employs a module to effectively eliminate a considerable number of negative instances. SRN [56] proposes a selective two-step classification and regression technique, which draws inspiration from the RefineDet-model. The two-step classification is executed at low-level layers to restrict the search space of the classifier, while the two-step regression is carried out at high-level layers to achieve precise localization. Numerous subsequent studies [87, 77, 81] have enhanced SRN by employing various successful strategies, such as data augmentation during training, improved

feature extraction and training supervision, anchor assignment and matching strategies, multi-scale testing strategies, and others. Although most of the aforementioned approaches necessitate the use of preset anchors for face identification, certain single-stage representative detectors, such as DenseBox-model [70], UnitBox-model [71], and CenterFace-model [79], can recognize faces without them. In the subsequent subsection, we will present these detectors as anchor-free models.

Table 2.3: *The categorization of deep face detection methods.[40]*

Category	Description	Method
Multi_stage	Detectors produce candidate boxes initially, then refine the candidates in one or more phases.	Faceness [60], HyperFace [61], STN [44], ConvNet-3D [62], SAFD [51], CMSRCNN [63], Wan et al. [64], Jiang et al. [48], DeepIR [49], Grid loss [65], Face R-CNN [47], Face R-FCN [66], ZCC [67], FDNet [46], FA-RPN [50], Cascaded CNN [41], MTCNN [68], Qin et al. [52].
Single_stage	Face categorization and bounding box regression are accomplished simultaneously by detectors using feature maps.	DDFD [69], DenseBox [70], UnitBox [71], HR [72], Faceboxes [55], SSH ,S3FD [73], DCFPN [53], FAN [74], FANet , RSA [75], S2AP [76], PyramidBox [54], DF2S2 , SFace , DSFD [58], RefineFace [77], SRN [56], PyramidBox++ [78], CenterFace [79], VIM-FD [80] , ISRN [81], AInnoFace , ASFD [82], RetinaFace [57], HAMBox[83] .
Anchor_based	Detectors place a number of dense anchors on feature maps before doing classification and regression on these anchors.	Wan et al. [64], Face Faster RCNN [48], RSA , Face R-CNN [47], FDNet [46], DeepIR [49], SAFD [51], SSH , S3FD [73], DCFPN [53], Face.boxes [55], FAN [74], FANet , Pyramid Box [54], ZCC [67], S2AP [76], DF2S2 ,SFace , Retina Face [57], DSFD [58], Refine Face [77], SRN [56], VIMFD [80],
Anchor_free	Face detectors locate faces without the need of predefined anchors.	Dense Box [70], Unit Box [71], Center Face [79]
Multi-Tasklearning	Detectors learn classification and bounding BoxRegression with additional objectives (such as LandmarkLocalization) in a same framework.	STN [44], ConvNet3D [62], HyperFace [61], MTCNN [68], Face R-CNN [47], RetinaFace [57], DF2S 2, PyramidBox++ [78], CenterFace [79], PCN , FLDet [84].

CPUReal-Time	Detectors can run on a single CPU core in real-time for VGAresolution images.	Cascade CNN [40], STN [44], MTCNN [68], DCFPN [53], Face.boxes [55],
ProblemOriented	Detectors are designed to address particular face detection issues, such as small faces, obstructed faces, rotated and hazy faces.	HR [72], SSH , S3FD [73], Bai et al. [9], PyramidBox [54], GridLoss [65], FAN [74], LLE.CNNs [85], PCN , GroupSampling [86]

2.4.3. Anchor-based and anchor-free methods.

Because of their lengthy development history and good performance, most modern face detectors are anchor-based, as shown in Table 2.3. In general, we pre-select anchors on the feature maps, do one or more classification and bounding box regression on these anchors, and then output the acceptable anchors as the recognition result. Anchor mapping and matching algorithms are thus critical for detection accuracy. Most anchor-based approaches, for example, scale compensation [88, 73], maximum output background label [73], predicted maximum overlap score [67], group-by-scale sampling [86], and so on, rely on algorithms in this direction. However, variables (such as size, increment, ratio, and number of anchors) must be carefully tweaked for each particular dataset, limiting generalizability. Furthermore, dense anchors increase the computational overhead and pose the issue of positive/negative anchor imbalance.

Anchor-free algorithms [89, 90] are gaining popularity in general object detection. In the field of face detection, some pioneering studies have arisen in recent years. The functions DenseBox-model [70] and UnitBox-model [71] attempt to forecast the pixel-wise bounding box on face. Face detection is seen by CenterFace-model [79] as a generalised job of keypoint estimation, which predicts the facial centre point and the size of the bounding-box in a feature-map. In summary, anchor-free detectors eliminate present anchors and improve generalisation capacity. In terms of detection accuracy, more research is needed to improve resilience to false positives and training process stability. see *table 2.3*

2.5. Recent face mask detectors

In this part, the existing algorithms for face mask detection are reviewed with their features, we based on survey papers focus on deep-learning approaches for face-mask detection, which can be found in the reference [91].

- **S. Ge et al. [92]** created an algorithm with the goal of detecting masked faces, where the term "mask" is used to describe both actual facemasks and any occlusion on the face. Faces that are obscured by various objects, such as scarves, hair, hands, or the niqab, are seen as masks that hide the face. In Section IV, the many forms of masks are discussed. To identify occluded faces, a comprehensive dataset known as MAFA or Masked Faces was introduced in their study . More than 35,000 masking face photos may be found in MAFA, ensuring that at least a portion of the face is hidden. Additionally, faces with diverse angles and orientations are included in the dataset. They listed six characteristics of MAFA. The authors divided this proposed model into three-parts: (a) a proposal module, (b) an embedded module, and (c) a verification module. The first one combines two CNN and extracts the features from face images. The second one is dedicated to finding the missing facial landmarks that occurred by occlusion. In this sub-phase, the Locally-Linear-Embedded (LLE) algorithm is utilised. The last module is the performing of classification and regression tasks using unified CNN to determine if it is a face or not and scales the position of missing facial cues. the authors compared the results of their model with six other face detectors and achieved the best performance. The average-precision of model was 74.6%.
- **Inamdar, Madhura, and Ninad Mehendale** in [93] proposed a model named Facemasknet,to detect if a person is wearing a facemask correctly or not, which summed up to a three class classification: no mask, improperly worn mask, and with a mask. The model was trained using a bespoke dataset of 35 photos. The faces in those 35 photographs included both masked and unmasked ones. The input data were pre-processed and scaled in the required value prior to training. The input image or live streams underwent pre-processing before being sent through the Facemasknet model, which first identified the face and then retrieved the Region of Interest (RoI). They claimed that their pieces contain two detectors. First, the face is detected. The RoI is extracted, and the Facemasknet model is then applied to those cropped images or live streams for classification. The green and yellow bounding boxes prominently refer to the face and facemask in an image, respectively. It contains very small and region-biased data. This model reported an accuracy of 98.6%. [91]

- **Khandelwal, Prateek** et al. [94] presented a model and implemented it in a real-world application that determines whether or not a mask is applied in an image. Two steps were used to divide the labour. One included finding faces in the image, while the other involved classifying masks. For face detection, they utilised a MobileNetV2 model built on the CNN architecture. This model is not able to detect faces smaller than a particular pixel count. After faces were found, the data was cleaned up and faces were labelled using semi-supervised learning before being fed into the mask detection stage. The model was constructed with MobileNetV2. The photos were scaled according to their needs before being fed into the network. Additionally, they employed an augmentation technique to add variation to the data. The authors took a validation set of 840 images combined with a mask and no mask among 4,225 annotated images. This model achieved high-performance and was already implemented, but the model had two major drawbacks. First, classification or detection of partially overlapped faces cannot be done using this method. Secondly, this model cannot detect faces where the height of the camera exceeds 10 feet. Their model achieved an Area Under Region of Convergence (AUROC) of 97.6%.

- **Agarwal et al**. In [95], a novel framework for face mask image recognition challenges is developed in response to the present COVID-19 pandemic. Classification tasks have been successfully used with CNN-SVM hybrid models. The suggested method produces high accuracy results when compared to previous comparable efforts. In comparison, it is discovered that the suggested work outperforms the other approaches. CNN is used to extract features from pictures, while SVM is used for classification. As a result, the authors summarise the evolution and propose a successful research framework based mostly on deep learning approaches based on CNN models.

Conclusion

In this chapter, we have exposed different methods of classification and detection of object, face and face mask. Detection techniques vary between one_stage and two_stage methods. In the next section, we present in detail detection techniques based on machine-learning systems and especially deep-learning. The evolution of object, face, and face mask detection has shifted from traditional feature-based methods to deep learning-based approaches due to their superior accuracy and robustness. Key insights include: Deep Learning Dominance: Deep learning methods, particularly CNNs, have become the standard for these tasks due to their ability to learn complex features and generalize well. Real-Time Performance: Models like YOLO, SSD, and MobileNet are preferred for real-time applications due to their speed and efficiency. Transfer Learning: Pre-trained models fine-tuned for specific tasks (e.g., face mask detection). Applications: These technologies are extensively utilized in security, healthcare (e.g., mask detection during pandemics), and retail (e.g., customer analytics). The selection of approach is contingent upon the particular application, computing limitations, and required precision. Deep learning methodologies represent the future; nonetheless, conventional techniques retain specific applications where simplicity and interpretability are paramount.

Chapter 3

Deep Learning for Detection/Localization

Introduction

In order to get machine learning (ML) one step closer to its ultimate objective of artificial intelligence (figure 3.1), deep learning was established. It relates to algorithms that were influenced by the anatomy and operation of the brain. To model complex interactions between data, they can learn many levels of representation. With the addition of more layers to the network, Deep Learning, which is based on the concept of artificial neural networks, is designed to manage massive volumes of data. With little to no human input, a deep learning model may extract features from raw data by applying numerous layers of processing that include both linear and nonlinear transformations, and it can then gradually learn about those features as it progresses through each layer [97, 98].

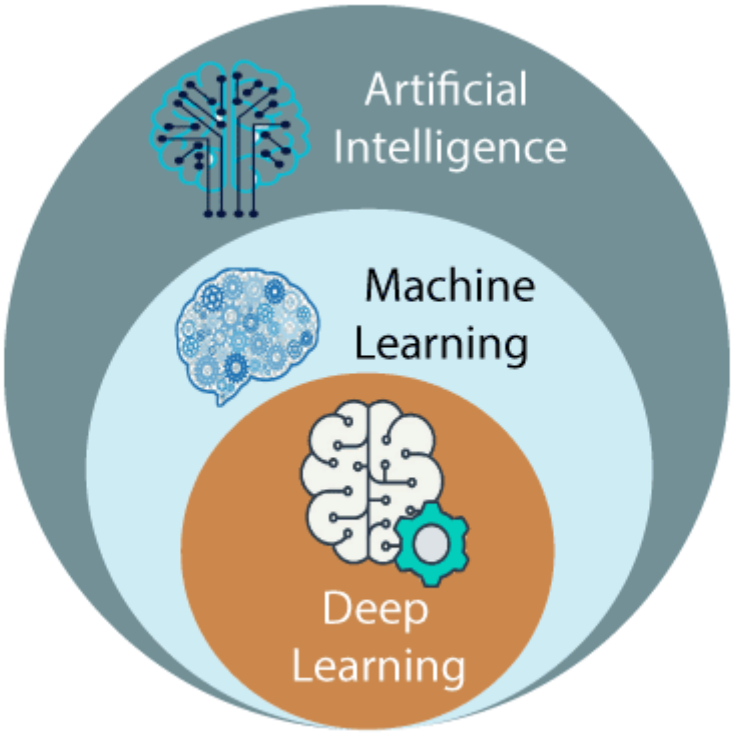


Figure 3. 1: *Artificial intelligence, machine learning, and DL relationships [96]*

Deep learning has evolved over the past five years from being a specialized topic in which only a few academics were interested to being the one that researchers appreciate most. Top magazines including Science [99], Nature [100], and Nature Methods [101], to mention a few, are now publishing research on deep learning. Deep learning has influenced the GO [102], taught humans how to drive, detected cancer [103], diagnosed autism [104], and even helped a person become an artist [105].

Dechter (1986) [106] and Aizenberg et al. were the first to use the term "Deep Learning" to machine learning (ML) and artificial neural networks, respectively. (2000) [107].

3.1 Why Deep Learning?

The ML algorithms that were discussed in the first section are effective for a wide range of issues. However, they were unable to resolve certain significant AI issues like object and speech recognition.

The failure of conventional algorithms to complete such an AI assignment was one of the driving forces behind the creation of deep learning.

But it wasn't until more data became accessible, particularly as a result of Big Data and linked items, and until processing power increased that we were able to fully grasp the promise of Deep Learning.

Deep Learning scales well; the more data supplied, the higher the performance of a Deep Learning algorithm. This is one of the key distinctions between Deep Learning and regular ML algorithms. Deep Learning models have no such restrictions (theoretically) and have even surpassed human performance in fields like image processing, in contrast to many traditional ML methods that have an upper constraint on the amount of data they can receive frequently referred to as the "performance plateau". (*figure 3. 2*).

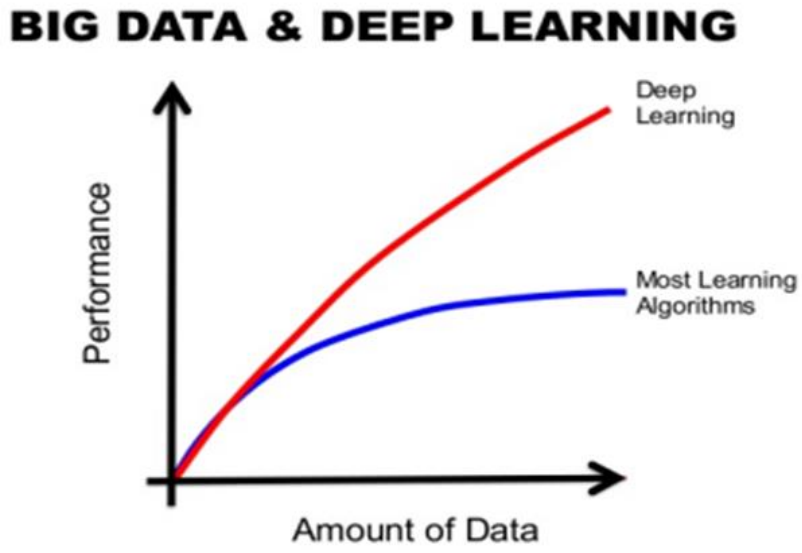


Figure 3. 2: *Performance difference between DL and most ML algorithms as a function of the amount of data [108]*

The feature extraction stage is another distinction between Deep Learning algorithms and conventional ML methods. Traditional machine learning algorithms require a subject matter expert to undertake the arduous and time-consuming task of feature extraction manually, whereas deep learning methods execute this task automatically. (*figure 3. 3*).

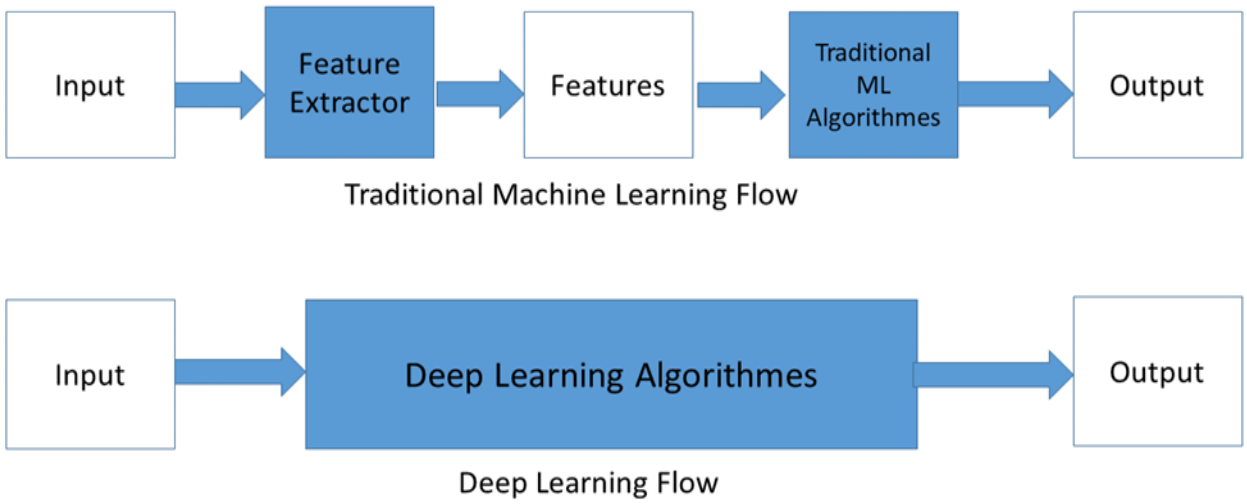


Figure 3. 3: *The classical ML process compared to that of DL*

3.2 Convolutional Neural Networks

A sort of customized neural network for data processing called a convolutional neural network (CNN) has a grid-like architecture (figure 3.4). Examples include data of the time series type, which resembles a 1D grid by being sampled at regular intervals, and data of the picture type, which resembles a 2D grid of pixels. Convolutional networks have achieved great success in real-world settings. The term "convolutional neural network" denotes the use of the convolutional mathematical operation by the network. A unique linear operation is convolution. Convolutional networks are just neural networks that, in at least one of their layers, employ convolution rather than matrix multiplication.

They are widely used in natural language processing [110], recommendation systems [109], and picture and video recognition [110].

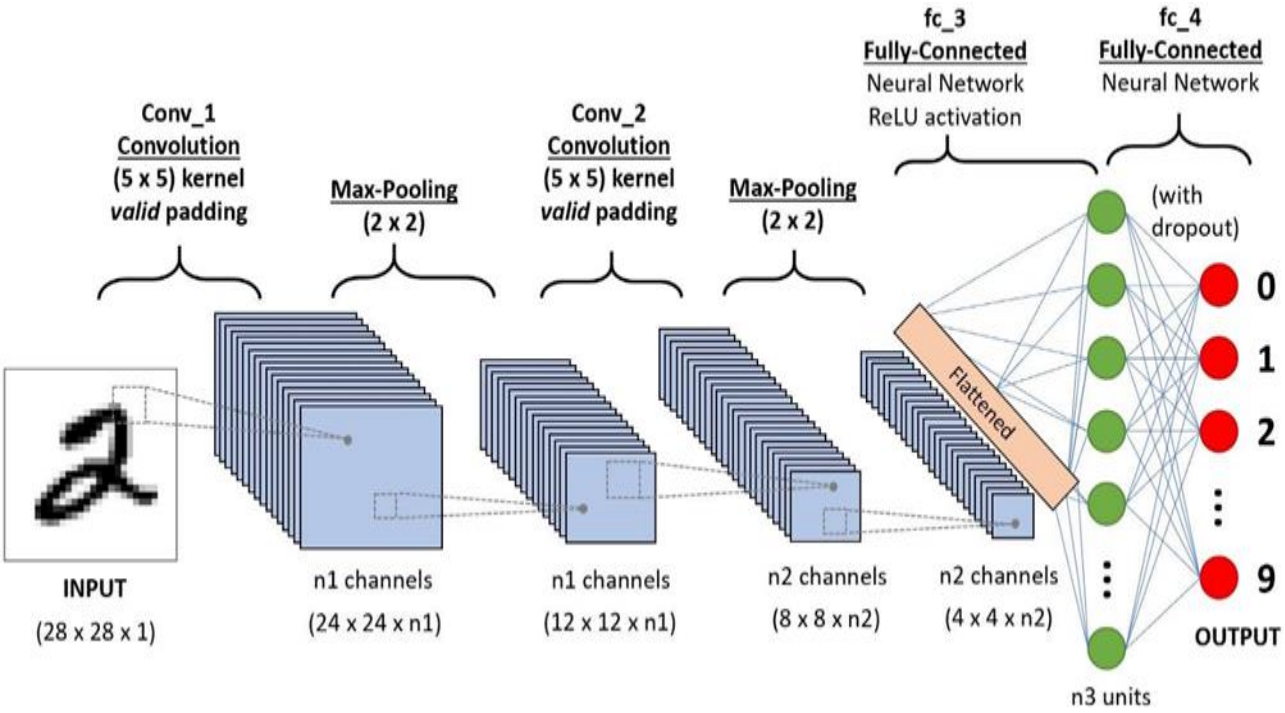


Figure 3. 4: *Convolutional Neural Network architecture [113]*

3.2.1 Convolution Operation

Convolution is an operation on two real argument functions in its most basic form. We begin with illustrations of two potential functions to help you comprehend the rationale behind convolution. Let's say we are using a laser sensor to locate a spaceship. The output of our laser sensor, x(t), represents the position of the spaceship at time t. Because x and t are real numbers, we can always obtain a different reading from the laser sensor.

Let's say our laser sensor is a little loud now. We want to aggregate a number of readings to get a less noisy estimate of the spacecraft's location. We want these measures to be a weighted average and to give greater weight to recent observations since, of course, more recent measurements are more pertinent. This may be accomplished using the weighting function w(a), where an is the measurement's age.

This weighted average process produces a new function that gives a smooth estimate of the spacecraft's location when applied at each instant:

$$s(t) = \int x(a)w(t - a)da \tag{3.1}$$

Convolution is the name of this operation. Typically, an asterisk is used to indicate the convolution operation:

$$s(t) = (x * w)t \tag{3.2}$$

The concept of a laser sensor in our case being able to produce measurements at every moment in time is unfeasible. Typically, time is discretized (digitalized) when working with data on a computer, and our sensor will output data at regular intervals. It is more practical to suppose that our laser gives a measurement once per second for the sake of our example.

Therefore, the time index t can only accept integer values. Now that x and w are assumed to be integers, we may define the discrete convolution as follows:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t - a) \quad (3.3)$$

The first argument of a convolution, in this case the x function, is frequently referred to as the input and the second argument, in this case the w function, as the kernel in the context of convolutional networks. The result is occasionally referred to as a feature map.

3.2.2 Convolutional Layer

The three key concepts of sparse interactions, parameter sharing, and equivariant representations are the foundation of convolution and may be used to enhance ML systems.

minimal interactions Traditional neural networks multiply a matrix of parameters by a different parameter for each input unit and output unit to describe how they interact with one another. This indicates that, unlike convolutional neural networks, each output unit interacts with each input unit. Making the kernel smaller than the input achieves this. When processing an image, for instance, the input picture may have thousands or millions of pixels, yet we can identify minute details like edges using kernels that only take up a few tens or hundreds of pixels. This allows us to keep fewer parameters, which lowers the model's memory needs and increases the model's effectiveness. Additionally, it means that fewer operations are needed to calculate the outcome. These efficiency gains are typically extremely large (figure 3. 6).

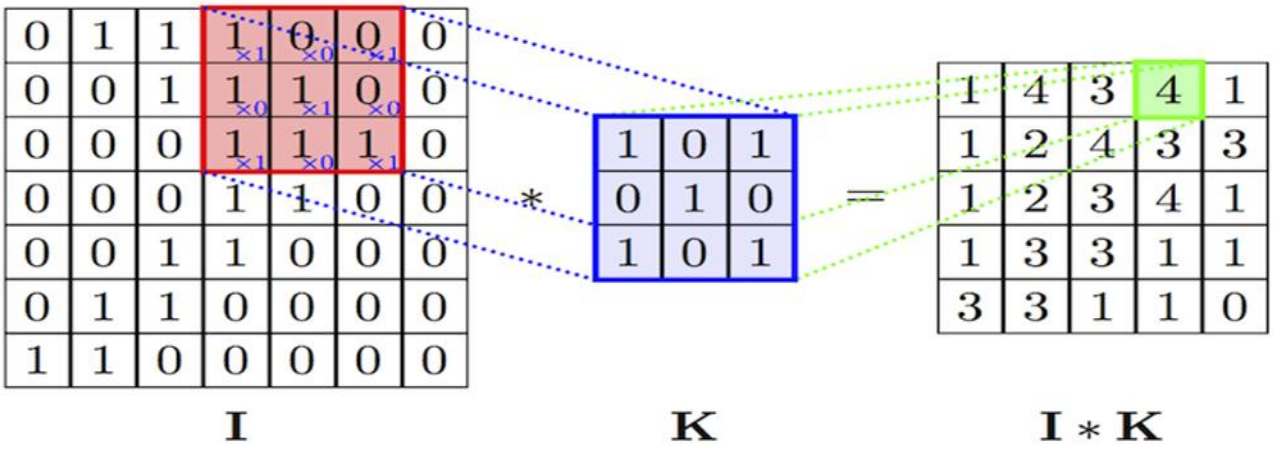


Figure 3. 5: *Description of convolution operation [111].*

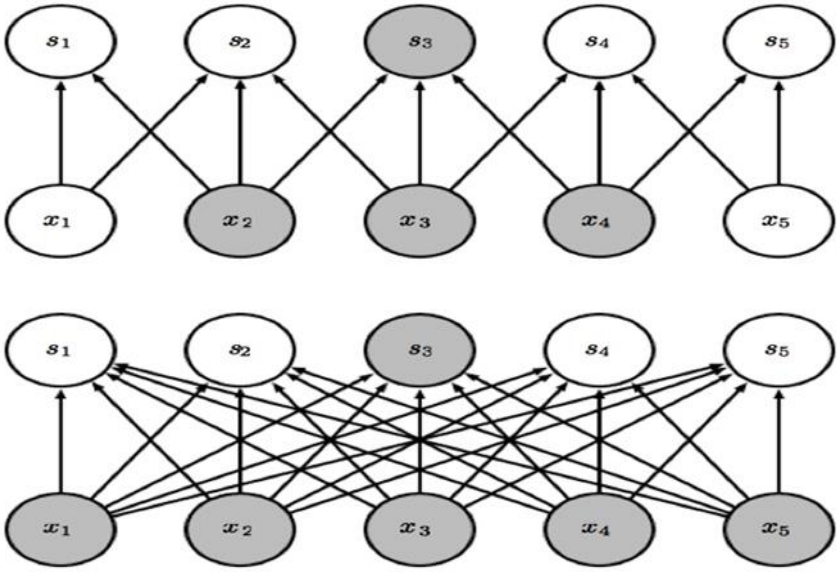


Figure 3. 6: *Description of Sparse interactions (connectivity) [112].*

We draw attention to the impacted output units and the input unit x_3. Only three outputs are impacted by x when s is created using convolution with a kernel of width 3 (top). (Bottom) X_3 can reach all outputs when s is created through matrix multiplication.

3.2.3 Pooling layer

A convolutional network with an unusual architecture has three distinct kinds of layers. Prior to using the pooling function, a convolutional layer is used to create a collection of linear activations, which are then passed through a nonlinear activation layer such the Rectified Linear Unit (ReLu).

- It permits reducing the size of the representations progressively in order to lessen the number of parameters and computations in the network and, consequently, control overfitting;It allows invariance to small translations;
- Useful when one prefers to know if a characteristic is present rather than the region of its presence;
- Several types of pooling differ (MAX pooling (very popular), AVG pooling, ...). (*Figure 3.7*)

3.2.4 Perceptron

We add a perceptron or an MLP at the end of the network after extracting the characteristics of the inputs. The extracted characteristics are sent into the perceptron, which creates a vector with N dimensions and N being the number of classes, or each element being the likelihood of belonging to a class. When the classes are purely mutual, the softmax function is used to determine each probability:

$$softmax(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (4)$$

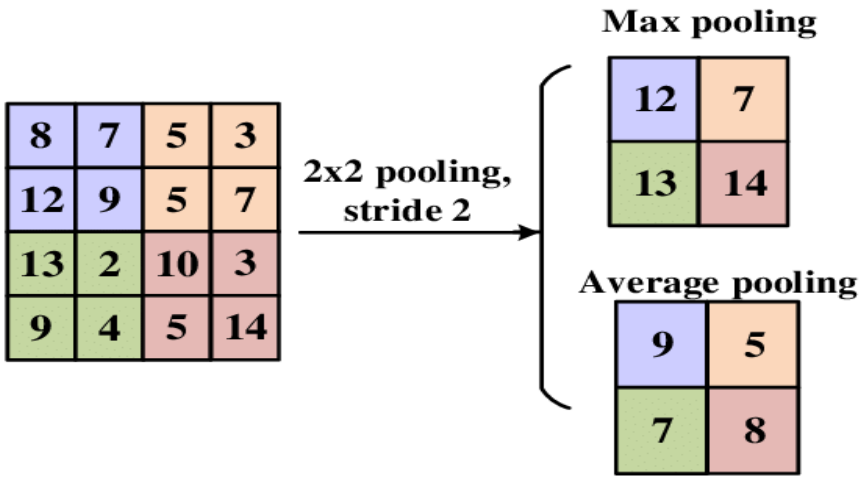


Figure 3. 7: *Pooling operation example*

3.2.5 Activation functions

In ANNs, activation functions are employed to transform input signals into output signals, which are then provided as input to the following layer. It has a significant impact on ANN prediction accuracy, thus attention must be used while choosing it. An ANN behaves as a Linear Regression 15 Model if it lacks an activation function, resulting in a linear function as the output signal. As a result, the network's performance is constrained. The BinaryStepFunction, Linear, Sigmoid, Tanh, ReLU, LeakyReLU, Parametrized ReLU, ExponentialLinearUnit, Swish, and SoftMax are some of the most significant activation functions.

3.2.5.1 Binary step function

The binary step function (Figure 3. 8) is a threshold-based activation function, meaning that once a given threshold is reached, activation occurs, and below that point, deactivation occurs. The threshold is zero on the graph up top.

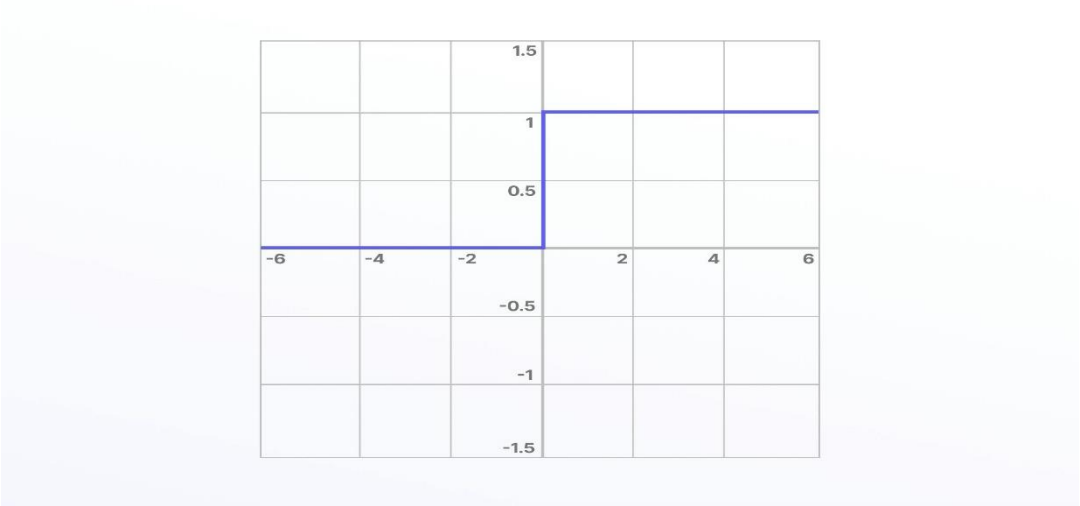


Figure 3. 8: *Binary step function [114]*

3.2.5.2 Linear Activation Function

The activation is proportionate to the input in the case of the linear activation function (Figure 3. 9), sometimes referred to as "no activation" or the "identity function" (multiplied by 1.0). The function just throws out the value it was given, doing nothing to the weighted sum of the input.

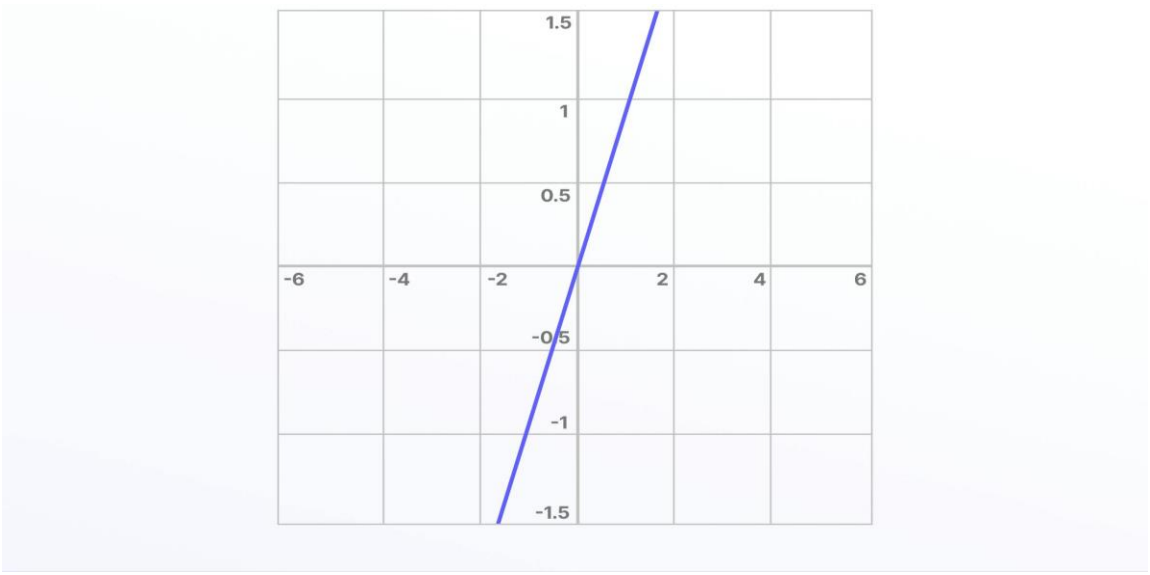


Figure 3. 9: *Linear Activation Function [114]*

3.2.5.3 Non-Linear Activation Functions

A linear regression model is all that the previous linear activation function is. Due of its limited computational power, the model is unable to construct complex mappings between the network's inputs and outputs.

3.2.5.4 Sigmoid / Logistic Activation Function

Any real value may be used as an input for this function, Which it outputs values that range from 0 to 1. As demonstrated below, the output value will be closer to 1.0 the larger the input (more positive) and closer to 0.0 the smaller the input (more negative).

3.2.5.5 Tanh Function (Hyperbolic Tangent)

with a -1 to 1 range of output fluctuation, the HyperbolicTangent function (Figure 3.10) is very similar to the sigmoid / logistic activation function and even has the same S-shape. Tanh's output value approaches 1.0 when the input is greatest (more positive), whereas it approaches -1.0 when the input is smallest (more negative).

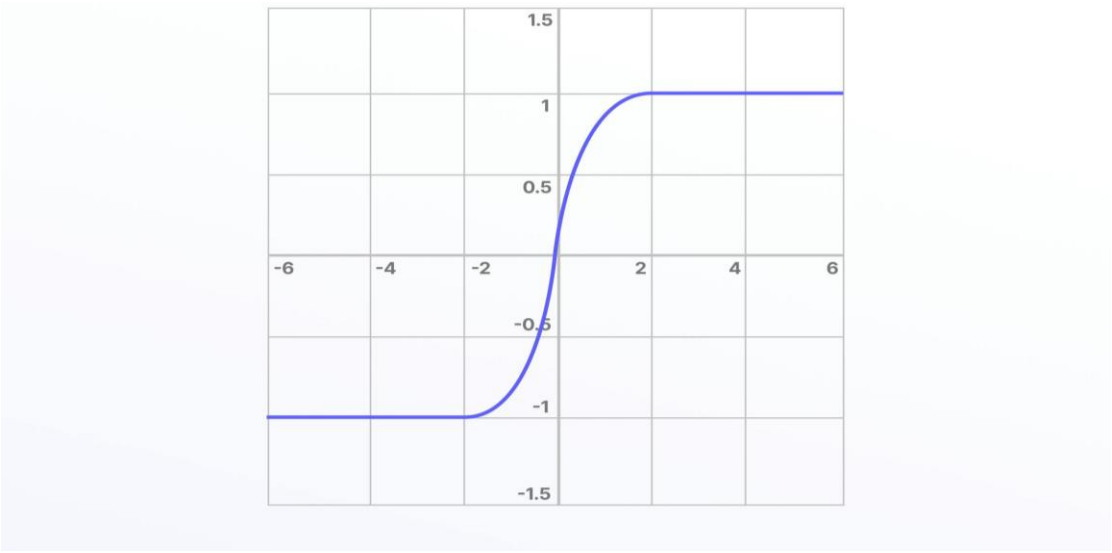


Figure 3. 10: *Hyperbolic Tangent function [114]*

3.2.5.6 ReLU Function

RectifiedLinearUnit is referred to ReLU (Figure 3. 11). ReLU has a derivative function and enables for backpropagation while still being computationally efficient, while giving the impression of being a linear function.

The fundamental issue here is that not all of the neurons are activated simultaneously by the ReLU function. Only if the result of the linear transformation is less than 0 will the neurons become inactive.

$$f(x) = max\ (0, x) \tag{3.5}$$

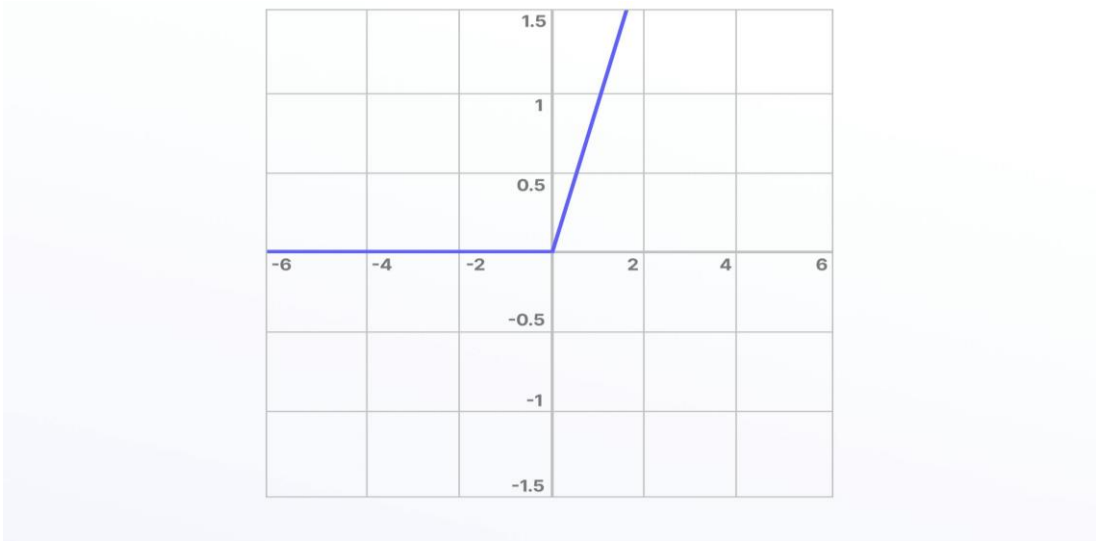


Figure 3. 11: *ReLU Function [114]*

3.2.5.7 Leaky ReLU Function

Leaky ReLU (Figure 3.12), which has a little upward slope in the negative area, is an improved variant of the ReLU function for solving the Dying ReLU problem.

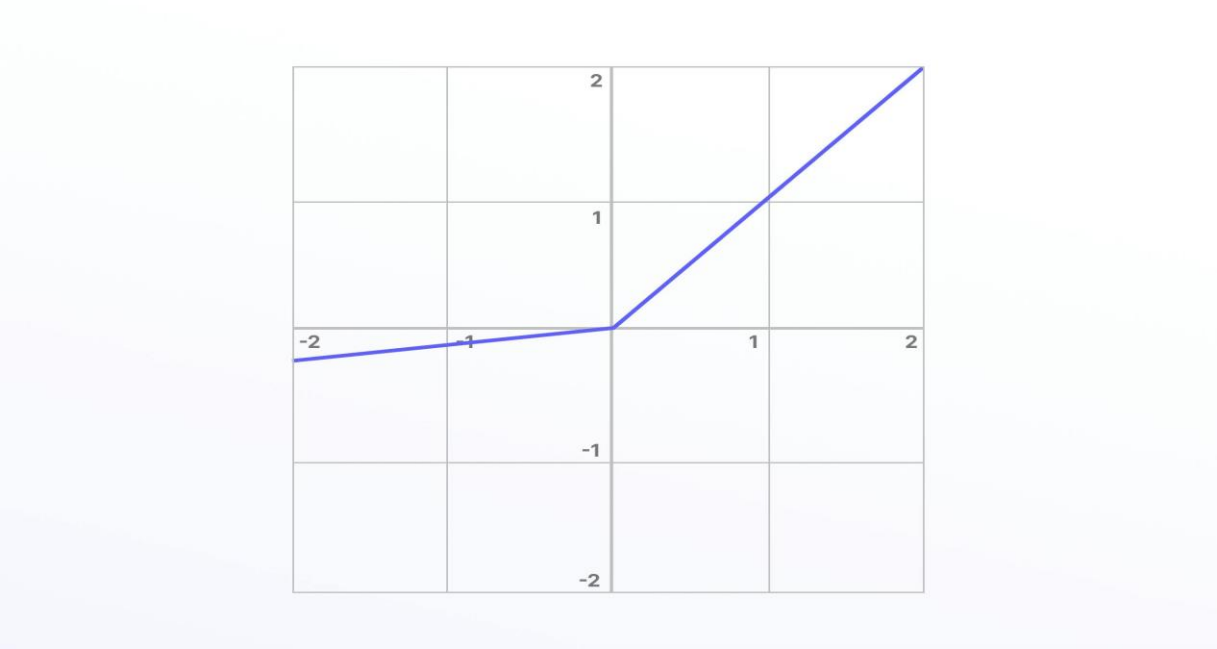


Figure 3. 12: *Leaky ReLU Function [114]*

3.2.5.8 Mish

"Mish: A Self Regularized Non-Monotonic Neural Activation Function" (*Figure 3. 13*) is the new deep learning activation function that shows improvement over ReLU (+ 1.671%) on final accuracy.

$$f(x) = x*\tanh(\text{softplus}(x)) \tag{3.6}$$

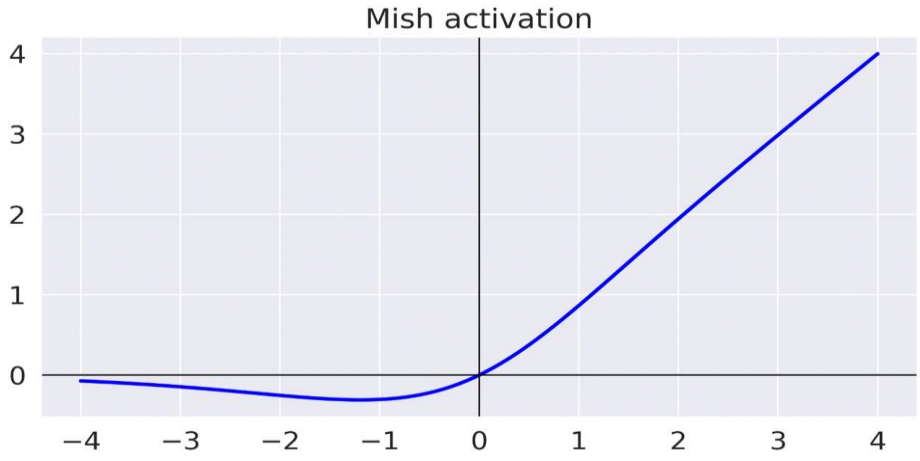


Figure 3. 13: *Mish activation function [114]*

3.2.6 Performance Metrics

The performance measures are helpful in capturing the model's performance. Prior to introducing more complex measurements, we identify four fundamental variables:

- **TruePositive (TP):** The model correctly predicts that an item belongs to a class.
- **TrueNegative (TN):** The model correctly predicts that an item does not belong to a class.
- **FalsePositive (FP):** The model incorrectly predicts that an item belongs to a class.
- **FalseNegative (FN):** The model incorrectly predicts that an item does not belong to a class.

The number of all inaccurate predictions divided by the whole dataset number yields the error rate (ERR). Error rates range from 0.0 to 1.0, with 1.0 being the worst.

$$ERR = \frac{FP + FN}{TP + TN + FN + FP} \tag{3.7}$$

A model's accuracy is a statistical metric for how well it identifies or omits a circumstance. In other words, accuracy is the proportion of true outcomes (including TP and TN) to all instances studied. TN elements are substantially more numerous and predominate in object recognition difficulties, despite the fact that it is a common metric. It's outlined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.8}$$

Precision, also known as Positive Predictive Value, is the ratio of real positive results to all positive results. It is the proportion of positive records that the model properly detected out of all positive records.

$$Precision = \frac{TP}{TN + FP} \tag{3.9}$$

Sensitivity, also known as recall, measures the proportion of genuine positive predictions to true positives and false negative outcomes. It displays the positive results that the algorithm properly detected out of the real positive results.

$$Recall = \frac{TP}{TN + FN} \tag{3.10}$$

F1 score, also known as F-measure, is a statistic used to quantify incorrect predictions by taking into account recall and accuracy.

$$F1 - score = \frac{2\ x\ precision\ x\ recall}{precision + recall} \tag{3.11}$$

Mean average precision (mAP) is defined as the mean of average precision across all K classes.

$$mAP = \frac{\sum_{i=1}^{K} AP_i}{K} \quad (3.12)$$

3.2.7 LossFunction

A LossFunction compares the target and anticipated output values; it assesses how effectively the neural network predicts the training data. When training, we try to minimise the difference between anticipated and goal outputs.

3.2.7.1 Types of LossFunctions

There are two forms of LossFunctions in supervised learning, which correspond to the two types of neural networks: regression and classification loss functions. [115].

Regression LossFunctions — used in regression tasks; given an input value, the model predicts a corresponding output value (rather than pre-selected labels); Ex. MeanSquaredError, Mean Absolute Error [115].

Classification LossFunctions — utilised in classification tasks; given an input, the neural network produces a vector of probabilities of the input belonging to various pre-set categories — can then select the category with the highest probability of belonging;

Ex. Binary Cross-Entropy, Categorical Cross-Entropy:

- ***MeanSquaredError (MSE)***

MSE is a common loss function that calculates the average of the squared differences between the target and anticipated outputs.

This function has several qualities that make it ideal for computing loss. The difference is squared, thus it makes no difference whether the projected value is higher or lower than the target value; nevertheless, values with a big inaccuracy are penalised. MSE is also a convex function (as seen in the image above) with a clearly defined global minimum, which allows us to use gradient descent optimisation to select the weight values more simply. [116].

- ***Mean-Absolute-Error (MAE)***

The average of the absolute discrepancies between the target and predicted outputs is calculated using MAE.

In some circumstances, this loss function is utilised instead of MSE. As previously stated, MSE is extremely sensitive to outliers, which can have a significant impact on the loss because the distance is

squared. MAE is used to counteract this when the training data contains a significant number of outliers. [117].

- ***Binary Cross-Entropy / Log-Loss***

This is the loss function used in binary classification models, which take in an input and must categorise it into one of two pre-defined categories.Classification neural networks operate by producing a vector of probabilities — the likelihood that the supplied input falls into each of the pre-defined categories — and then picking the category with the greatest probability as the final output. [118].

There are just two possible real values of y in binary categorization— ***0*** or ***1***. Thus, in order to correctly measure the loss between the actual and expected values, the real value must be compared to the predicted value. (***0*** or ***1***) with the probability that the input-aligns with that category (***p(i)*** = probability that the category is ***1***; ***1 — p(i)*** = probability that the category is 0).

- **Categorical Cross-Entropy_Loss**

When the number of classes is higher than two, we use categorical cross-entropy, which works in the same way as binary cross-entropy.Categorical cross-entropy is a subset of binary cross-entropy, where ***M = 2*** — the number of categories is ***2*** [118].

3.3 Deep Localization and Detection

Object localization algorithms consist of two categories: The 1st category is named two stage detectors perform object localization and detection jointly, whose famous models are RCNN (Recurrent-Convolutional-Neural-Network) [26], Fast-RCNN [27], Faster-RCNN [28], which starts with Region-Proposal-Network (RPN) to generate regions of interest, then performs classification and bounding box regression. The second category is called single-stage detectors, whose famous models are You Only Look Once (YOLO) [31], YOLOv2 [32], YOLOV3 [33], YOLOv4 [34], Scaled-YOLOv4 Scaling: CSP [35], Single-Shot-Multibox-Detector (SSD) [36], RetinaNet [37] and EfficientDet [38].

3.3.1 Single Stage Detectors

3.3.1.1 YOLO family combining Detection and Localization

- **YOLOv1:** The You Only Look Once (YOLO) model can be translated as “see only once” is a model that consists of a single neural network that can be trained end-to-end by back-propagation. It merges the two steps of the previous algorithms, object detection and localization, into one model [31]. However, YOLO formulates the object detection problem

differently, as a regression task that spatially separates bounding boxes and associates class probabilities. The network is trained to learn very general representations of objects. It takes the entire image as input and predicts bounding boxes simultaneously for all classes in an image. The image is divided into an S×S grid, then bounding boxes B and confidence scores are predicted for these boxes. If the center of an object is in a cell of the grid, the grid will predict bounding boxes B with a confidence score. To calculate the confidence score, one must calculate Intersection over-Union (IoU), which is the difference between the predicted box and the Ground Truth. In this way, the confidence score which is probability (Object) × IoU can be calculated. The advantage of YOLO over other methods is its speed. It is so fast compared to other methods, which makes it ideal for real-time applications. However, this comes with a trade-off in terms of accuracy. YOLO makes more location errors. The network architecture is shown in figure 3. 14

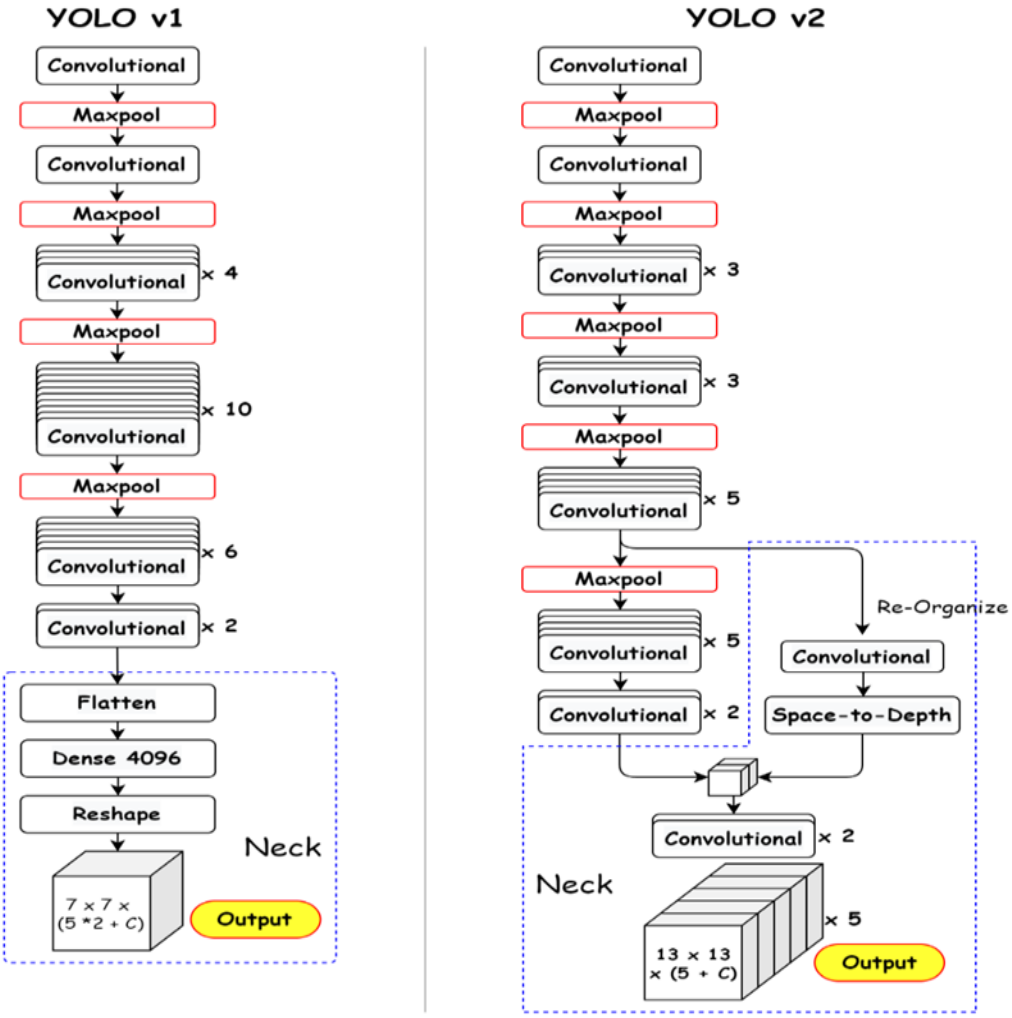


Figure 3. 14: *Architecture of YOLOv1 and YOLOv2 models [32]*

- **YOLOv2 [32]:** The newer version of YOLO, the YOLOv2 version, which overcomes the constraints of YOLOv1, was released at the end of 2016 by Joseph Redmon and Ali Farhad. The following are the primary adjustments that make YOLOv2 a superior model in terms of performance:
    - ***BatchNormalization:*** The addition of batch normalisation to all convolutional layers increased mAP by 2%. It also aided in tuning the model, reducing any overfitting.
    - ***High-resolution classifier:*** First, the model is tuned on ImageNet at 448 * 448 resolutions, which gives the model more time to update its filters and raises mAP by 4%.
    - ***AnchorBoxes:*** More BoundingBoxes are employed, and K-means clustering is used to calculate the AnchorBox input dimensions.
    - ***Fine grained features:*** It predicts detections on a 13 × 13 feature map, which is smaller than the one used by YOLOv1. This enhanced tiny item localisation while staying efficient for bigger objects.
    - **Multi scale training:** The YOLOv1 fared poorly at identifying objects with varying image sizes. YOLOv2 selects picture dimensions at random, with the lowest being 320 * 320 and the highest being 608 * 608.

- **YOLOv3 [33]:** The authors published the third version of YOLOv3 in April 2018.The mAP-50 on the COCO dataset rose from 44.0% to 57.9% of YOLOv2. In comparison to RetinaNet, which has 61.1% mAP by default, RetinaNet has an input size of 500x500. When the input size is 416 * 416, the detection speed is around 98 ms/frame, whereas YOLOv3 has 29 ms/frame. The network Architecture of output results of YOLOv3 detection model is shown in figure 3. 15.
    - ***Backbone:*** Darknet-53 is used;
    - ***Neck:*** FPN (Feature Pyramid Network) is used;
    - ***Head:*** YOLO is used.

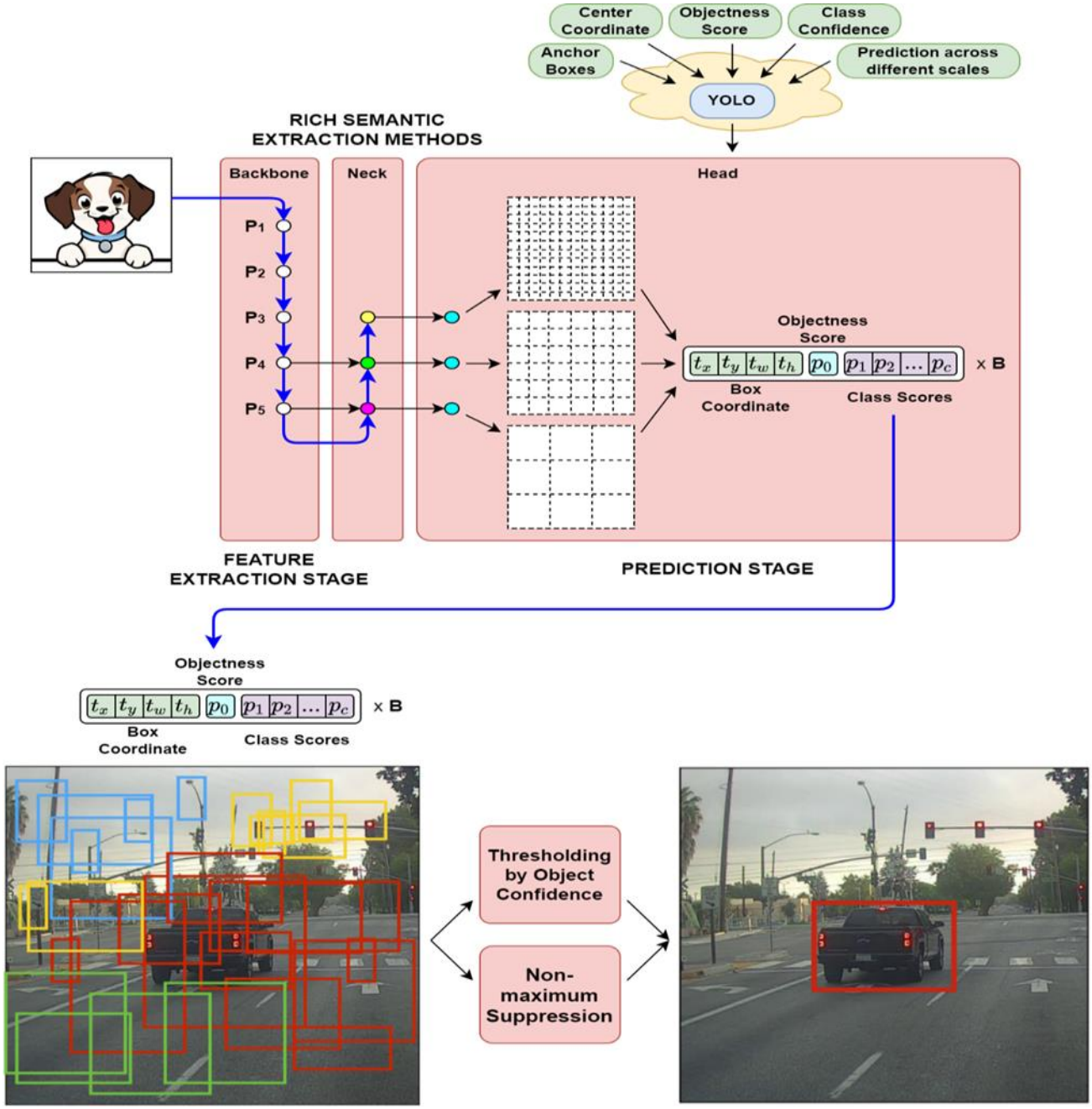


Figure 3. 15: *Architecture of YOLOv3[119]*

- **YOLOv4** (see Figure 3. 16)**:** Yolov4 was published in 2020 and is an upgraded version of the YOLOv3 algorithm, with a 10% increase in mAP and a 12% increase in the number of frames per second.

The authors of YOLOv4 present a series of contributions termed a "bag of freebies" in their article. This is a sequence of measures that may be made to increase the model's performance without raising inference latency. Because they cannot alter the model's inference time, the majority of them make improvements to the training pipeline's data management and data augmentation. These strategies increase and scale up the training set, exposing the model to previously unforeseen circumstances.

Another advancement is the use of "bag of specials" approaches, which alter network design and occasionally raise the cost of the output process [34].

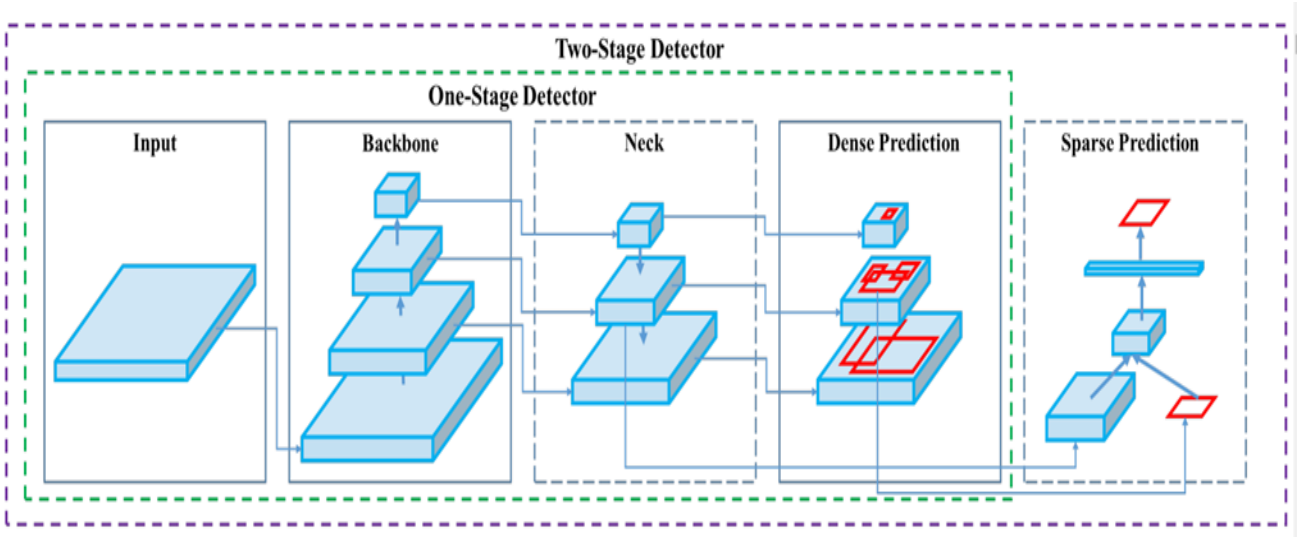


Figure 3. 16: *Architecture of YOLOv4 [34]*

- **YOLOv5:** YOLOv5 (Figure 3. 17) was distributed just on GitHub in 2020, with no accompanying study. It differs from all previous versions in that it is a PyTorch implementation rather than a fork from the original Darknet. The most significant enhancements are mosaic data augmentation and auto-learning bounding box anchoring. YOLOv5 is significantly faster and lighter than YOLOv4 [34], although its accuracy is comparable to the YOLOv4 benchmark. YOLOv5 runs training data through a data loader, which augments data online, with each training batch. Scaling, colour space changes, and mosaic augmentation are the three types of augmentations performed by the data loader. The most unique of them is mosaic data augmentation, which merges four photos into four random ratio tiles. The PyTorch framework lets you to reduce the floating point accuracy in training and inference from 32 bits to 16 bits. This drastically reduces the YOLOv5 model's inference time.

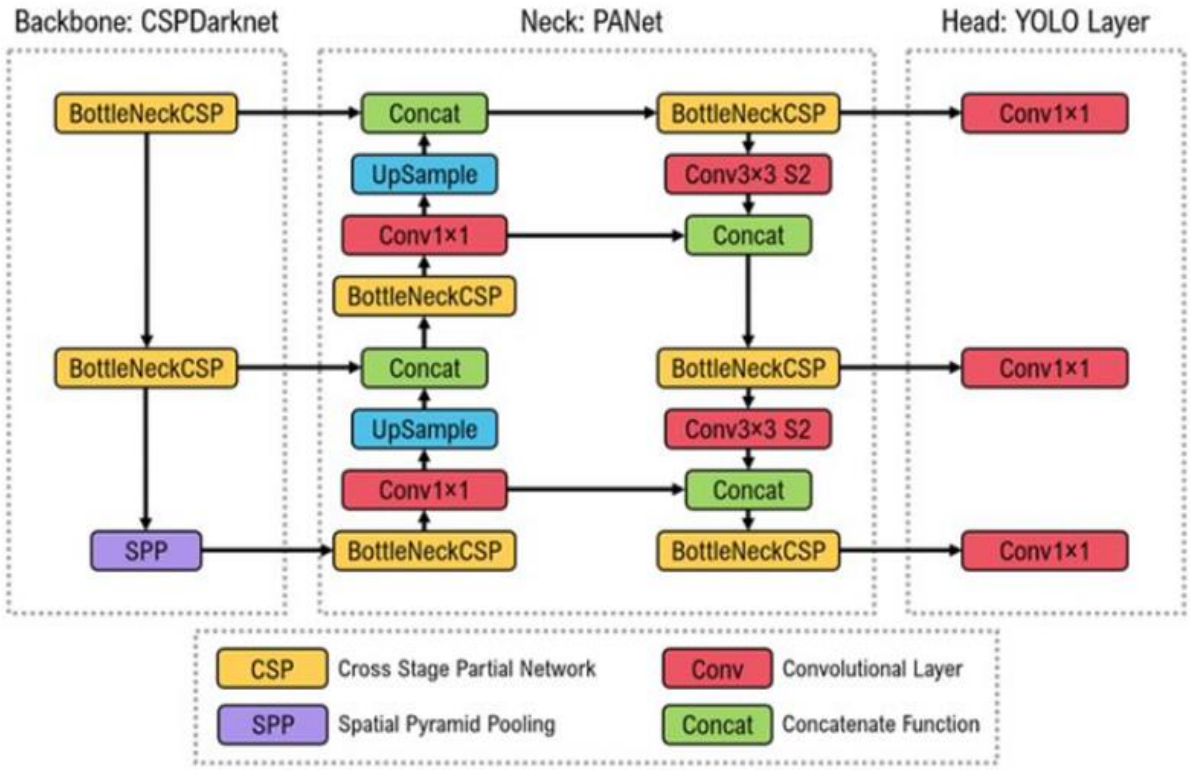


Figure 3. 17: *The architecture of the YOLOv5 model [120]*

The YOLOv5 architecture is divided into three parts: The backbone is CSPDarknet, the neck is PANet, and the head is YOLO Layer. The data is first supplied into CSPDarknet for feature extraction before being loaded into PANet for feature fusion. Finally, the YOLO Layer returns the object detection results (class, score, position, and size).

- **Scaled-YOLOv4:** Scaled-YOLOv4 is a YOLOv4-based target detection model. The depth of layers and the number of stages in the network's backbone and neck are scaled in Scaled YOLOv4 to improve model performance. Some layers in the YOLOv4 scaled neural network are also created with a CSP architecture, which minimises the amount of processing resources necessary to train the network. Since its recent publication by the Google Research/Brain team, the EfficientDet family of models has become the favoured object detection models. The new contribution of the EfficientDet [35] research was to consider how to scale object detection algorithms up and down strategically.

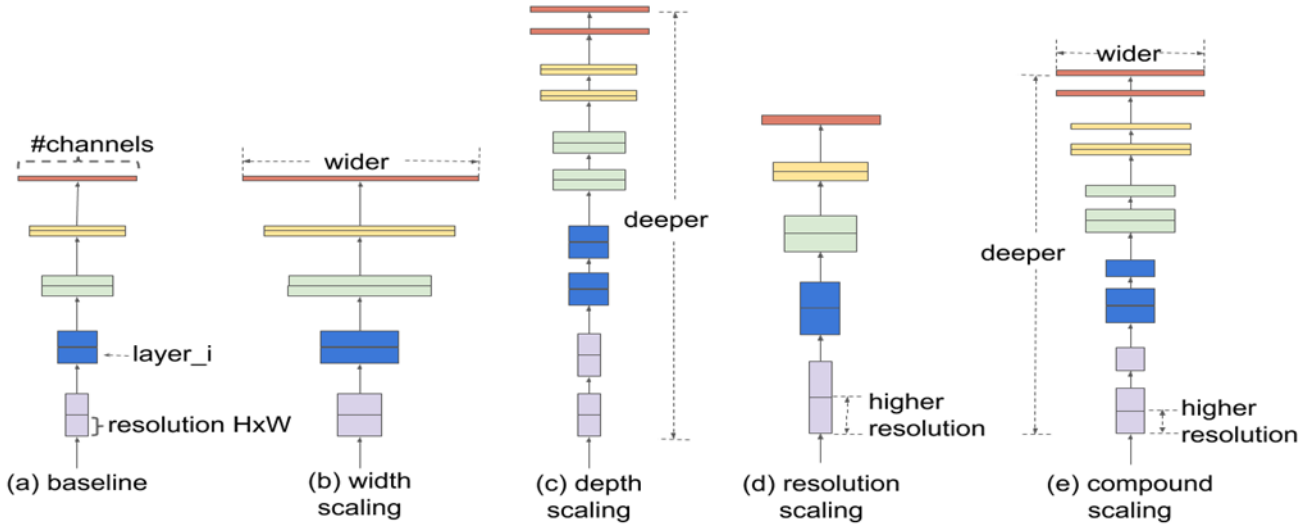


Figure 3. 18: *Model scaling example [35]*

increasing object detection models involves taking big picture input resolutions, increasing the breadth of convolutional network layers, scaling the depth of convolutional layers, and then scaling everything together (Figure 3.18). The inventors of EfficientDet utilised a search to discover the ideal scaling threshold from EfficientDet-D0 to EfficientDet-D1[38], and then used this setting to linearly scale to the well-known EfficientDet-D7 [38]. Overall, the authors of Scaled-YOLOv4 [35] strike a compromise between certain scaling principles - picture size, number of slices, and number of channels - when developing their model and optimising model performance and inference speed (Figure 3.19). They are considering using some CSP based CNN backbones, ResNet, ResNeXt and traditional Darknet backbones in their network. In the Scaled-YOLOv4 [35] paper, the authors often write that they "CSPized" a certain part of the network. CSPize means applying the concepts outlined in the Cross-Stage Partial Networks.

To recognise large objects in large pictures, the scientists discovered that increasing the depth and number of stages in the CNN backbone and neck is critical. (Increasing the width appears to have no effect.) This enables them to scale up the input size and number of stages before dynamically adjusting width and depth to meet real-time inference performance requirements. In addition to these scaling parameters, the authors modify the design of their model.

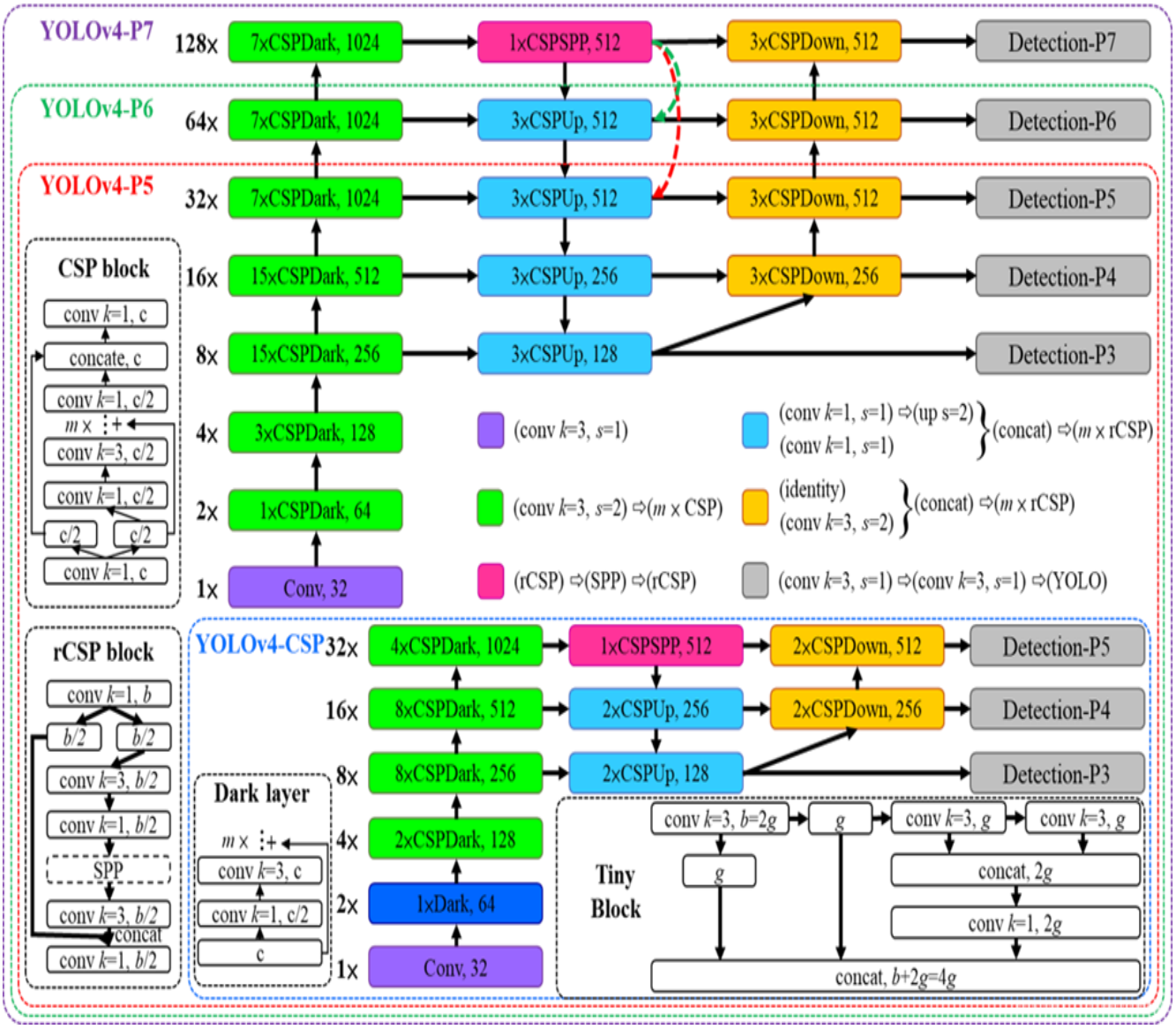


Figure 3. 19: *Architecture of YOLOv4-large (Scalled YOLOv4), including YOLOv4-P5, YOLOv4-P6, and YOLOv4-P7[35]. The dashed arrow means.*

3.3.1.2 Swin Transformer

Transformers [121] have had a profound impact on the field of Natural Language Processing (NLP) since its inception. Its application in language models such as Bidirectional Encoder Representation from Transformers (BERT) [122], Generative Pre-trained Transformer (GPT) [123], Text-To-Text Transfer Transformer (T5) [124] pushes the state forward development. State-of-the-art transformers [121] use attention models to build dependencies between sequence elements and can handle longer contexts than other sequence architectures. The success of Transformers in NLP has spurred interest in their application in computer vision. Although CNNs have been the mainstay of viewing progress, they also have some inherent flaws, such as: B. Lack of importance of global context, fixed weights

after training [125] and other computer vision tasks. It splits the input image into multiple non-overlapping patches and converts them into embeddings. A large number of Swin Transformer blocks are then applied to the patch in 4 stages, with each subsequent stage reducing the number of patches to maintain a hierarchical representation. The Swin Transformer block consists of a local Multi-head Self-Attention (MSA) module based on alternately shifted patch windows in consecutive blocks. In local self-awareness, computational complexity scales linearly with image size, while moving windows allow cross-window connections. In [126] also showed how moving windows can improve recognition accuracy with little overhead. Transformers represent a paradigm shift in CNN-based neural networks. While its use in image processing is still in its early stages, its potential to replace convolutions in these tasks is very real. Swin Transformer achieves state-of-the-art on MS COCO dataset, but uses higher parameters than convolutional models.

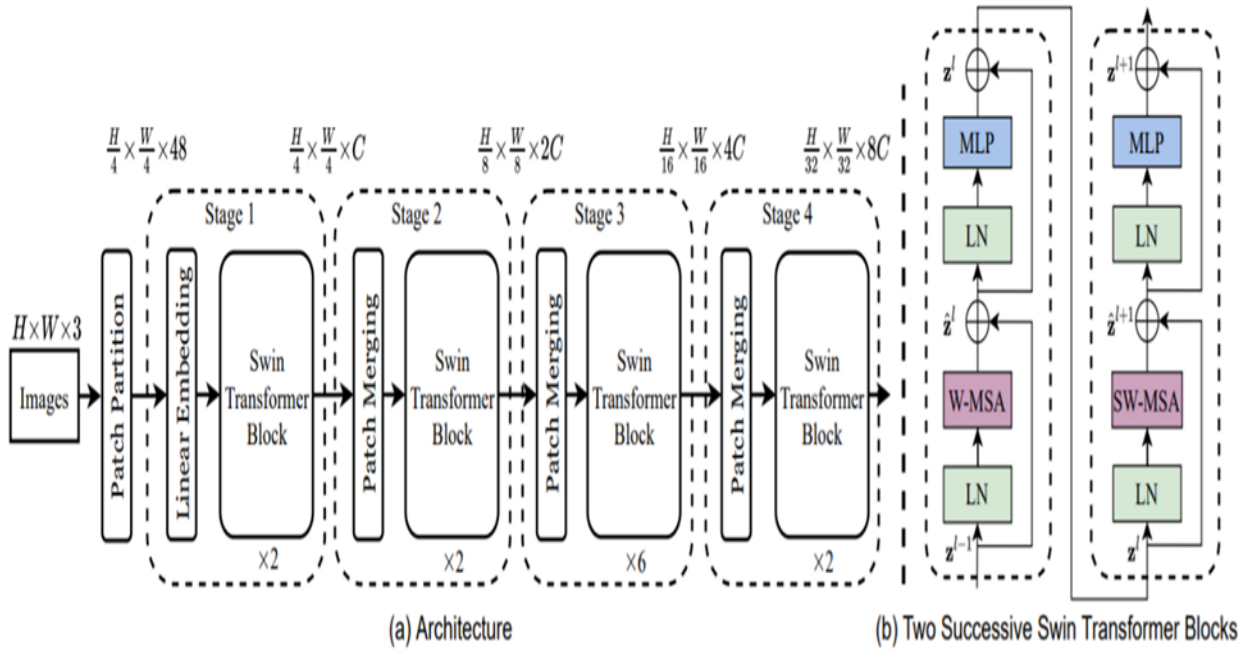


Figure 3. 20: *Swin Transformer model architecture [126]*

3.3.1.3 Knock Detector (SSD)

Single-Shot Detector (SSD) has eliminated the region proposal stage of the object detection pipeline. Thus, the neural network does not need to resample features to produce hypotheses and bounding boxes. Using a convolution filter, SSD predicts object categories and offsets in bounding box locations. Additionally, another convolution filter is used to perform object detection at different scales. The filters are applied to the feature maps of the first part of the neural network. This leads to a faster and

more accurate algorithm than the previous ones. Similar to YOLO and RFCN, SSD offers a model consisting of a single convolutional neural network that can be trained end-to-end. Specifically, SSD is based on a feedback convolutional neural network that produces a set of bounding boxes and scores for the presence of object classes [49]. The first part of the neural network called the base network follows a standard architecture (VGG-16 architecture) and is responsible for feature extraction. The second part of the network produces a set of forecasts.

These predictions include the predicted coordinates of the bounding boxes, including the center, width, and height coordinates of the box. In addition, the network generates a vector of probabilities related to trust for each class of objects. In addition, two other methods are used during practice time. To keep the most relevant areas, a method called non-maximal suppression is used, then the result of this is consumed by the Hard Negative Mining method lists the predicted negative boxes based on the confidence score and selects a subset of them to be used for the calculation of the error. Indeed, many negative boxes are expected during the training and could have a destructive effect on the formation of the network see *figure 3.21*.

The SSD model applies multiple layers to the feature maps generated by the base network to increase the number of relevant bounding boxes.

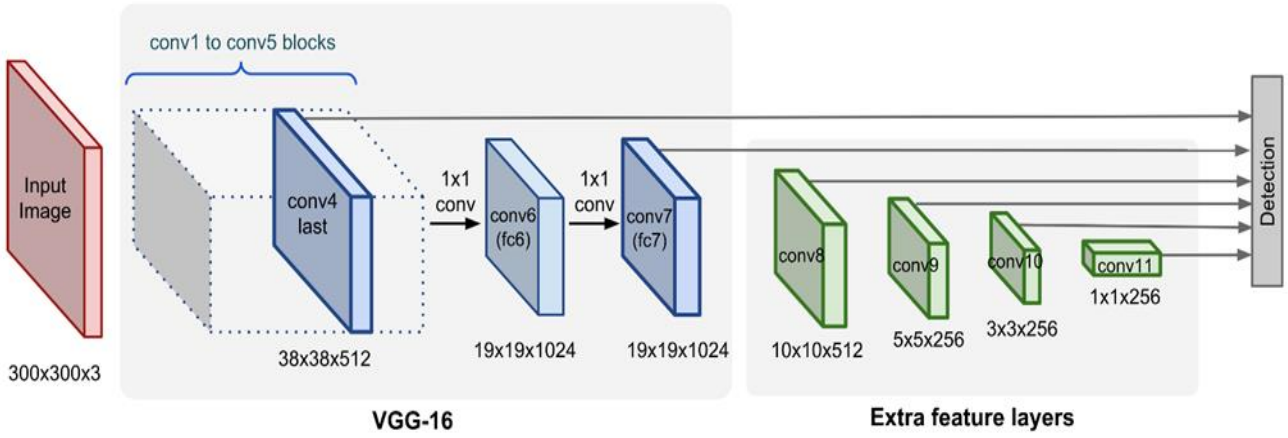


Figure 3. 21: *SSD model architecture [36]*

3.3.2 Two-Stage Detectors

3.3.2.1 Region-Based Convolutional Network (R-CNN)

The region-based convolutional network (R-CNN) is the first work that applied the deep learning method in object detection problems. The main idea [26] is that the algorithm finds all objects in an

image using an exhaustive search algorithm and then ranks the proposed objects using CNN. The search algorithm for locating objects in an image is called selective search.

This search algorithm was designed to locate objects in images. The selective search algorithm is able to deal with a variety of image conditions.

The basis of the selective search algorithm is the hierarchical clustering algorithm. Using bottom-up clustering, the selective search algorithm is able to generate object locations at all scales. The grouping process continues until the entire image becomes a single region. The detected regions are then processed using various color spaces with different invariance properties, different similarity measures, and varying the starting regions. The output of the selective search algorithm is a set of region proposals that may contain an object. The R-CNN model combines selective search and CNN methods to locate and classify objects.

The R-CNN is composed of three modules. The first generates a set of proposal regions using the region of interest search. The second is a CNN to extract a fixed-length 4096-dimensional feature vector from each region. The third module is a set of linear SVM classifiers whose input is the feature vector and its output is the probability of belonging to an object category. The architecture of R-CNN is shown in *figure 3.22*.

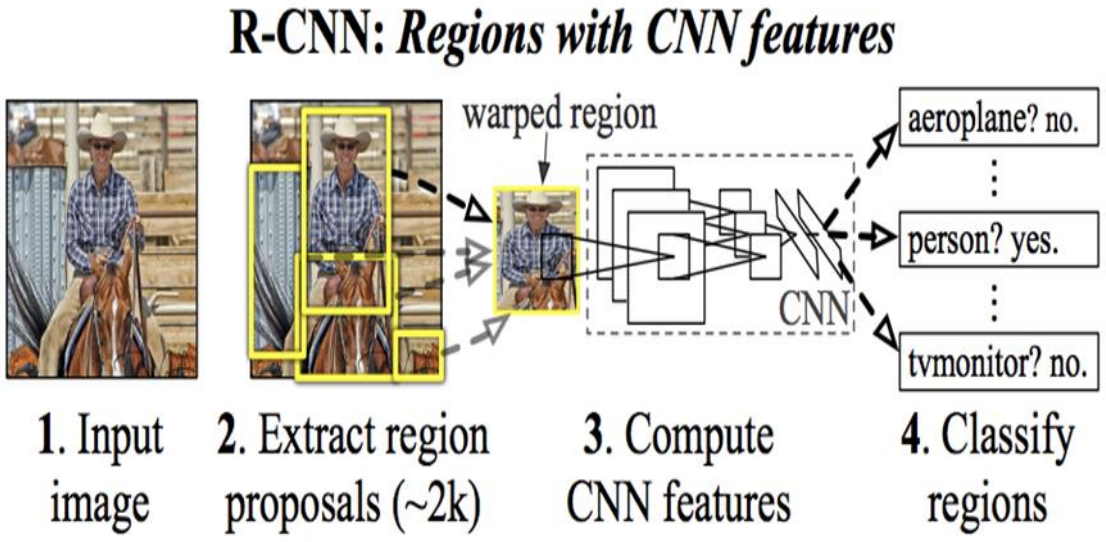


Figure 3. 22: *Principle of Region-Based Convolutional Network (R-CNN) [26]*

3.3.2.2 SPP-Net

The authors of this work, proposed the use of SpatialPyramidPooling (SPP) layer [127] to process image of arbitrary size or aspect ratio. They came to understand that a fixed input was only necessary for the CNN's completely linked portion. SPP-net [128] simply added a pooling layer and repositioned CNN's convolution layers prior to the region proposal module, making the network independent of size/aspect ratio and minimising calculations. The selective search [18] algorithm is utilised to generate candidate windows. Feature maps are obtained by passing the InputImage through the ConvolutionLayers of a ZF-5 [129] network. The candidate windows are then mapped on to the feature maps, which are subsequently converted into fixed length representations by spatial bins of a pyramidal pooling layer. This vector is passed to the fully connected layer and ultimately, to SVM classifiers to predict class and score. Similar to R-CNN [26], SPP-net has as post processing layer to improve localization by bounding box regression. It also uses the same multistage training process, except that the fine tuning is done only on the FullyConnectedLayers. SPP-Net is considerably faster than the R-CNN model with comparable accuracy. It can process images of any shape/aspect ratio and thus, avoid object deformation due to input warping. However, as its architecture is analogous to R-CNN, it shared R-CNN's disadvantages too like multistage training, computationally expensive and training time as well

3.3.2.3 Fast R-CNN

Fast R-CNN [27] is a variant of R-CNN aimed at accelerating object detection. R-CNN suffers from three major drawbacks:

- The first is that the algorithm consists of several steps that are learned and regulated separately;
- The second is training time. It is reported that for 5K frames of VOC 2007, training takes 2.5 GPU-days.

The last problem is at the time of testing, where it is necessary to make applications in real time, however, each image requires processing of 47 seconds.

Fast R-CNN is designed to reduce the amount of computation and memory required for R-CNN by using lossy multitasking to train the entire network in a single pass and update all network layers. Rapid R-CNN takes the entire input image and sends it to the main CNN. Using multiple convolutions and clustering layers, a feature vector is extracted from the input. The feature vector is used by a region-of-interest clustering layer to extract a fixed-length feature vector for each object proposal. The

selective search method is applied to search the RoI regions. Each feature vector is then flattened to feed into fully connected layers which ultimately generate two analogous output layers:

- The first is a one-hot coding vector passed through a softmax layer to indicate the probability of belonging to K object classes for each proposed object;
- The second output is a real-valued vector with four real values for each of the K object classes, which encodes the coordinates of the predicted selection boxes for the detected objects.

Fast R-CNN parts is shown in figure 3.23.

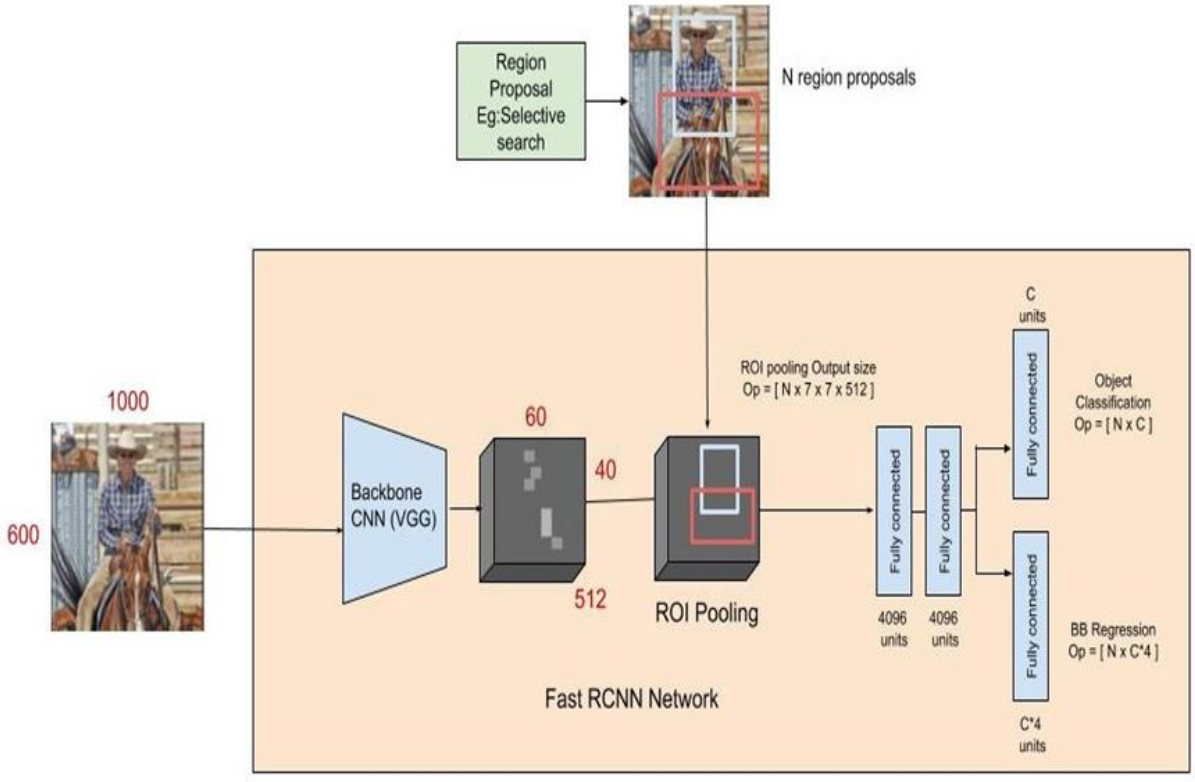


Figure 3. 23: *Architecture of the Fast R-CNN model [27]*

3.3.2.4 Faster R-CNN

Faster RCNN [28] is an object detection architecture presented by Ross Girshick, Shaoqing Ren, Kaiming He and Jian Sun in 2015. Faster R-CNN is designed to replace the selective search algorithm used in previous versions of R- CNN. The problem with selective search is that it is computationally expensive. Although Fast R-CNN introduced new features to reduce training and testing time, selective search remained a bottleneck for R-CNN algorithms.

In faster R-CNN, a new network called Region Proposal Network (RPN) has been introduced to replace the selective search algorithm. This network aims to propose regions that will be used later by the Fast R-CNN network to predict bounding boxes and detect objects. RPN uses a pretrained model on the Image-Net dataset for classification. More specifically, the RPN network, which is a deep convolution network that offers regions, takes an image as input and generates a feature map as output. The feature map is then used by a small network.

The small network takes as input an n × n sliding window on the feature map. The output of the small network is an equivalent output, one regression layer per box and one classification layer per box. On each window location, the small network predicts several region proposals. The number of region proposals is defined by a parameter called K. The K proposed regions determine the number of reference areas applied to all window locations to create region proposals. These boxes have different scales and aspect ratios to capture all possible objects at the current drag position and are called anchors. In this way, there is an anchor K for each sliding window. By using anchors, FasterR-CNN can handle multiple scales and formats. The box classification layer generates a probability vector indicating an objectivity score for each anchor box. The detected anchor boxes are then selected based on the objectivity score. Anchor boxes exceed a predefined threshold and then are routed to Fast R-CNN. It should be noted that FasterR-CNN merges the RPNnetwork with FastR-CNN using a mechanism called "attention mechanism". The RPNnetwork guides the Fast R-CNNnetwork where to look. To share the computation, the convolution functions are shared between RPN and Fast R-CNN. The rest of the algorithm is similar to FastR-CNN. FasterR-CNN is composed of 3 parts as shown in figure 3.24.

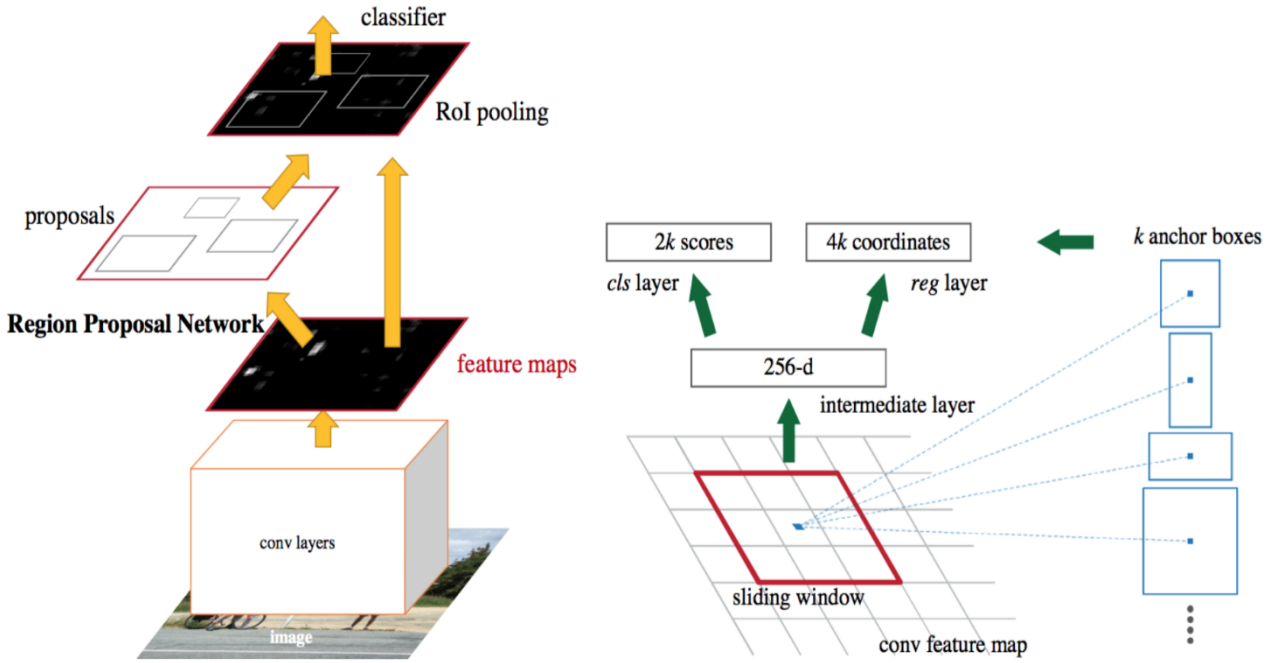


Figure 3. 24: *Architecture of Faster R-CNN [28]*

3.4 Conclusion:

In this chapter3, we have seen what and how Deep Learning differs from traditional ML algorithms. We've seen some major milestones in its evolution and the feats that have been accomplished with it. We have introduced some methods used by the Deep Learning community (classification, localization and detection) and we have explained the principle of each. In the next steps, we pass to the practical, and we illustrate our proposed deep learning models.

Chapter 4

Experiment 1: Yolo4FaceMask: COVID-19 Mask Detector

Introduction

The COVID19 epidemic has compelled numerous nations to enact stricter regulations regarding the use of face masks. To combat the spread of COVID-19, governments have compelled hospitals and other organisations to install additional infection control procedures. The transmission rate of COVID19 is approximately 2.4 [130]. The rate of transmission, however, may vary depending on the measure and strategies used by governments. Governments have begun imposing new restrictions mandating people to wear face masks as COVID-19 spreads via airdrops and close contact. The goal of using face masks is to minimise the rate of transmission and dissemination.

Personal protective equipment (PPE) is recommended by the World-Health-Organisation or WHO for usage between people and in medical treatment. However, most nations' capacity to grow PPE production is quite restricted [130]. COVID19 is now a serious public-health and economic concern due to the virus's negative impacts on people's quality of life, leading to acute respiratory illnesses, death, and financial crises worldwide. According to the WHO, more than six million people have been infected with COVID19 in over 180 countries, with a 3% fatality rate. COVID19 spreads quickly in crowded places and through close touch. In many nations, governments face enormous obstacles and hazards in safeguarding people from coronavirus. Because many nations prohibit people to wear face masks in public, masked face identification is critical for facial applications such as object detection. To fight and win in the battle against the COVID19-pandemic, governments need guidance and oversight on people in especially crowded public spaces to ensure face mask laws are enforced. This might be implemented by integrating monitoring systems with artificial-intelligence models [130].

However, most of the mask detection apps and current research on mask detection models aim to solve the problem of detecting masked and unmasked faces, but ignore the problem of wearing the mask incorrectly. Lack of research will lead to the spread of the virus by people who wear face masks incorrectly.The medical masked face is central to this work to minimize the transmission and spread of COVID19. Our main objective in this first work is: 1) the detection and the localization of masked-faces, 2) incorrectly-masked-faces and 3) unmasked-faces. The result of the proposed mask detector in the image or in the surveillance video is shown in Figure. 4.1. Given an image, a region of masked faces and incorrect masks and unmasked faces on the input image based on Yolov4 will be shown in the output. The work is structured in five parts. The first is devoted to the introduction; the second part lists the most original recent works. We describe our approach in part three and review the design of the approach in part four. The fifth part is dedicated to the results and discussions. We end this chapter with a conclusion and some perspectives.

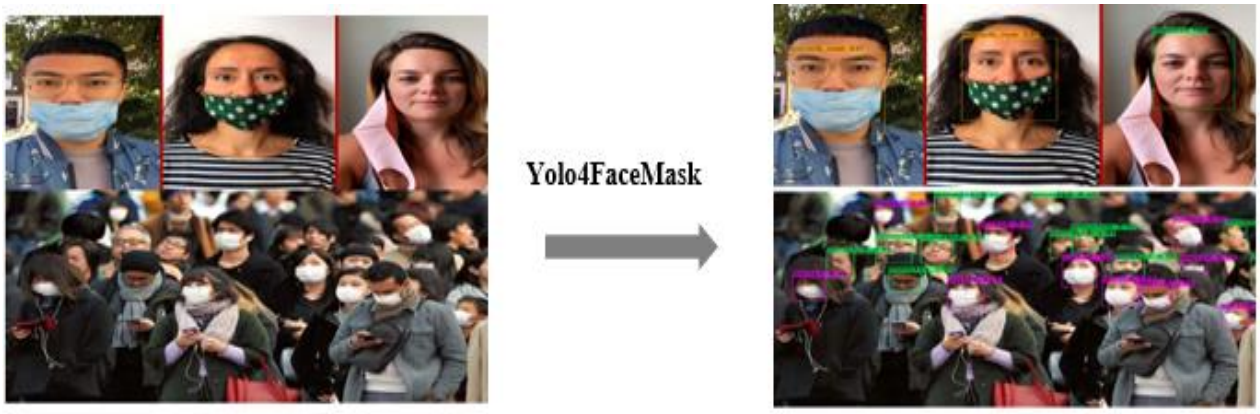


Figure 4. 1 : *The outcome of the proposed masked face detector*

4.1. Related work

4.1.1. Recent DL-based detection and localization methods

A survey of two object detectors, along with datasets, metrics, and fundamentals, can be found in [131]. other investigation [132] focuses on DL approaches for object detection. State-of-the-art object detectors use DL approaches, which are generally divided into two categories. The first is called OneStage detectors, whose famous models are YOLO v2/v3/v4 [31-35], SSD [36], RetinaNet [37] and EfficientDet [38]. The second one is named TwoStage detectors, RCNN (RecurrentConvolutionalNeuralNetwork) [26], FastRCNN [27] and FasterRCNN [28] which starts with region proposals and then performs bounding box classification and regression. These models have generally been evaluated on datasets from PascalVOC [133] and MSCOCO [39]. The accuracy and real-time performance of these approaches are good enough to deploy pre-trained models for face mask detection.

Table 4.1: Object detection and accuracy

Dataset	**VOC12**		**MSCOCO**
Model Name	mAP	FPS	mAP
RCNN	0.53	0.5	-
Fast RCNN	0.68	7	0.19
Faster RCNN	0.70	19	0.22
SSD	0.75	45	0.27
YOLOV2	0.73	67	0.22
YOLOV3	0.75	47	0.33
YOLOV4	0.79	62	0.43

Table 4.2: Speed and Accuracy of YOLOv4 on MSCOCO [58]

ModelName	m-AP	F-P-S
YOLOV4(320)	0.370	-
(416)	0.410	096
(512)	0.430	083
(608)	0.4350	062

4.1.2. Faces Detection

Viola- Many advances in deep learning for discovering and learning things in numerous sectors of application have occurred in recent years. Most activities, in order to verify identification, concentrate on picture reconstruction and face recognition. However, the primary goal of this investigation is to identify persons who do not use masks in public settings in order to limit Covid-19 transmission. [141] developed a method for recognising a face mask using the SRCNet classification network, achieving 98.7% accuracy in categorising photos into three kinds. First and foremost, it is about "wearing the proper face mask." Second, one is "wearing the wrong mask" Many advances in deep learning for discovering and learning things in numerous sectors of application have occurred in recent years. Most activities, in order to verify identification, concentrate on picture reconstruction and face recognition. However, the primary goal of this investigation is to identify persons who do not use masks in public settings in order to limit Covid-19 transmission. [141] developed a method for recognising a face mask using the SRCNet classification network, achieving 98.7% accuracy in categorising photos into three kinds. First and foremost, it is about "wearing the proper face mask." Second, one is "wearing the wrong mask" Jones [57] presented a boost-based cascade architecture with basic but fast Haar features as one of the most renowned early face detectors.

N. Marku and colleagues [134] suggested an object identification approach based on pixel intensity comparisons. The comparison of pixel intensities between distinct nodes in this study results in a quick detection time. M.Belahcene et colleagues employ IPC detection for face verification [135], which combines detection and alignment into a single model. G. Ghiasi et al proposed using a hierarchical deformable component model [136] to identify occluded faces in order to perform face identification and key point localization. In addition to the face detectors mentioned above, CNN-based models have made significant development in recent years. In [137], B. Yang et al introduced a face identification technique that used a feature aggregation approach [138] based on CNN to extract features. S. Yang et al developed a deep learning technique to face identification based on the responses of face components [60]. C. Zhu et al. recently suggested MS-RCNN: region-based contextual multi-scale

CNN for unconstrained face identification [139], which used contextual information. M. Opitz et al suggested the loss of grid: obstructed face detection. S. Luo et al introduced SFA: face detector attention small faces [65], which is a multi-branch framework for detecting tiny faces (accurate detection). Face identification using receptive field-enhanced multi-task cascading convolutional neural networks was suggested by X. Li et al [140].

4.2. recent Methods of COVID-19 face mask-based DL

Many advances in deep learning for discovering and learning things in numerous sectors of application have occurred in recent years. Most activities, in order to verify identification, concentrate on picture reconstruction and face recognition. However, the primary goal of this investigation is to identify persons who do not use masks in public settings in order to limit Covid-19 transmission. [141] developed a method for recognising a face mask using the SRCNet classification network, achieving 98.7% accuracy in categorising photos into three kinds. First and foremost, it is about "wearing the proper face mask." Second, one is "wearing the wrong mask" The work in [142] proposed by Sabbir Ejaz et al applied PCA (Principal Component Analysis) [143] in order to know the masked and unmasked faces. Observed that PCA is effective for face recognition without mask with an accuracy of 96.25%, but its accuracy is decreased to 68.75% in face recognition with mask. G. J. Chowdary [144] proposed a face mask detection model using transfer learning (TL) of InceptionV3, proposed approaches by achieving 99.9% accuracy during training and 100% during testing , but the model is trained and tested on the Simulated Masked Face (SMFD) dataset contained only 1570 images.

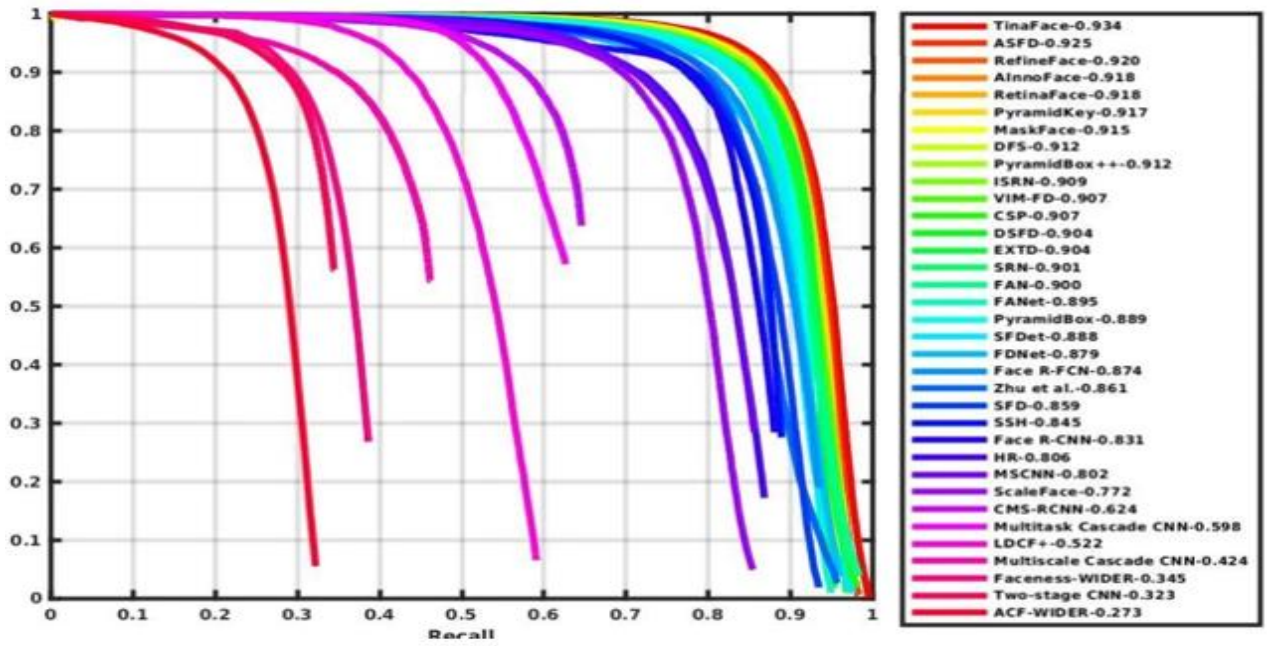


Figure 4. 2: *Wider dataset face detection results [14].*

4.3. Proposed Approach

In this proposes system (*figure 4.3*), we use Yolo detection technique to identify faces in image, real time video or line_cameras. We train the YOLO-v4 custom model for face detection and localization using a large dataset consisting of approximately 14409 images belonging to 3 classes: “masked-faces”, “masked-incorrectly” and “unmasked-faces”. Blood among three others formed our data set:

• wider-face [14] for unmasked faces and masked-faces

• MMD [145] for masked-faces and incorrect masked-faces

• RMFD [146] for masked-faces and incorrect masked-faces

- **Proposed Approach**

In this proposes system (*figure 4.3*), we use Yolo detection technique to identify faces in image, real time video or line_cameras. We train the YOLO-v4 custom model for face detection and localization using a large dataset consisting of approximately 14409 images belonging to 3 classes: “masked-faces”, “masked-incorrectly” and “unmasked-faces”. Blood among three others formed our data set:

• wider-face [14] for unmasked faces and masked-faces

• MMD [145] for masked-faces and incorrect masked-faces

• RMFD [146] for masked-faces and incorrect masked-faces

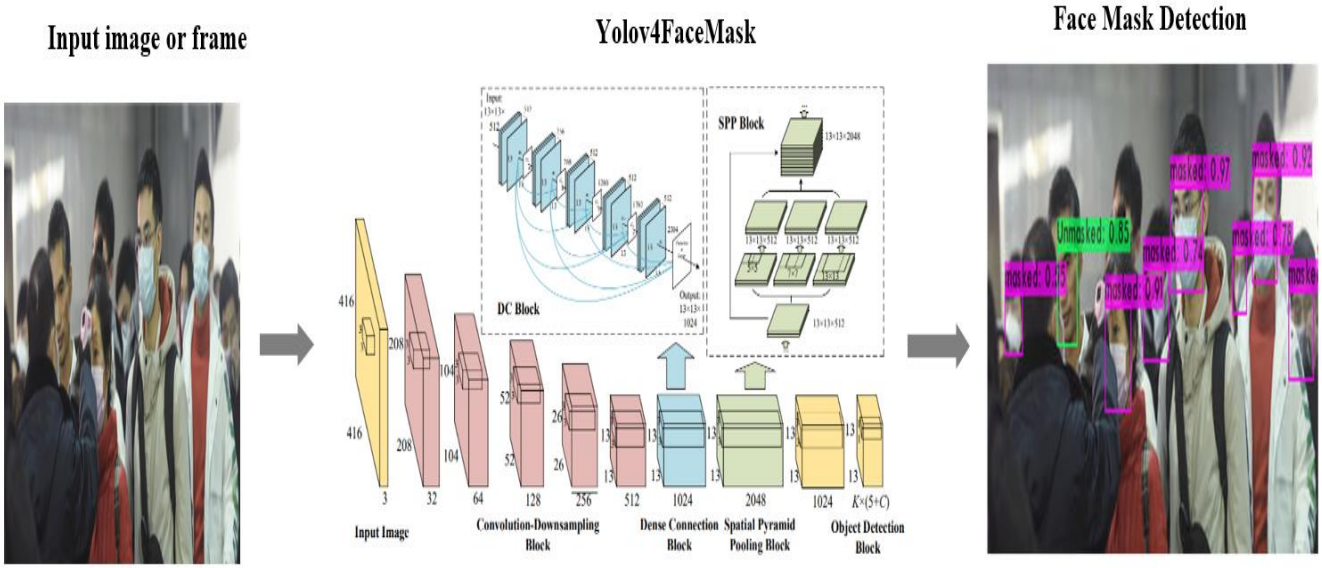


Figure 4. 3: *Proposed Approach Framework*

- **Dataset Description and pre-processing**

This proposed set of data consists of 14409 photos, 12879 of which are raw images from the Wider-face dataset [14]. The Wider-face dataset is a true unmasked-face dataset, however it contains some masked-faces. The remaining 1530 photos in our dataset were downloaded from Kaggle [145], and they all had masked faces and erroneous masks. Figure 4 depicts how challenging it is to use this dataset, which has 32203 photos and 393703 faces with a significant degree of heterogeneity in scale, position, and occlusion. Example of datasets is in *figure 4.4*

Figure 4. 4: Images from the datasets

- **Data pre-processing**

Data pre-processing is the process of converting data from one format to another that is more user pleasant. In our example, we transform the dataset to yolov4 format using the following steps:

- Place all photos in the same file named data;
- Separate the data file into train and test;
- Create file.txt for each image;
- Create train.txt/test.txt/file.names/file.data.
- Make a configuration file.cfg

- **Network Architecture**

YOLOv4 [34], uses :

- Bag-of-Freebies or BoF for back_bone: Cut_Mix and Mosaic data augmentation, Drop_Block

regularization, Class-label smoothing

- Bag-of-Specials or BoS for backbone: Mish_activation, Cross-stage_partial_connections (CSP), Multiinput-weighted-residual-connections (MiWRC)
- Bag-of-Freebies or BoF for detector: CIoU-loss, CmBN, DropBlock regularization, Mosaic data augmentation, Self-Adversarial Training, Eliminate grid sensitivity, Using multiple anchors for a single ground truth, Cosine annealing scheduler , Optimal hyperparameters, Random training shapes _ Bag of Specials (BoS) for detector: Mish_activation, SPP-block, SAM-block, PAN path-aggregation block, DIoU-NMS

Figure 5 depicts our Yolov4FaceMask architecture. We specify the configuration file of the Yolov4 object detector model described in [34], which recommends a detection network with a CSPDarknet53 [24] backbone, a neck, and YOLOv3 [33] heads, to create an effective network for detection and localisation of faces mask. The backbone is a universal feature extractor composed of convolutional neural networks that extracts information from pictures and converts it to feature maps. We utilised CSPDarknet53 as a standard backbone in Yolov4FaceMask. In terms of the neck, it is a component that sits between the backbone and the heads, and it can augment or refine the original feature maps. As a neck, Yolov4FaceMask, SPP [147], and PAN [148] were used. which may extract high-level semantic information and then fuse it into preceding layers' feature maps via an addition operation with a coefficient. Finally, heads represent classifiers, predictors, estimators,and so forth.

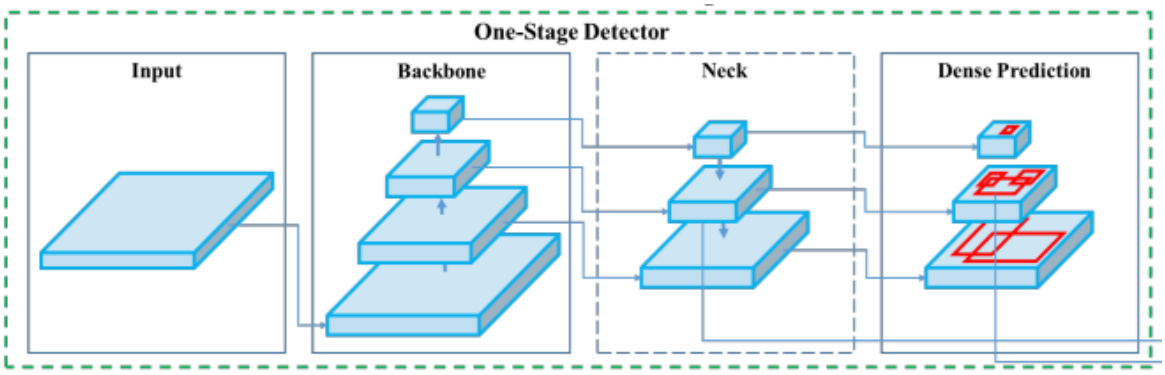


Figure 4. 5: *Yolov4FaceMask Network Architecture*

1) Mish activation function [149]

As for the backbone in our Yolov4FaceMask, we employed the mish_function (a unique self-regularized non-monotonic activation function) [149] instead of the Leaky-ReLu activation function used in the prior model. The fact that the activation function reached any height prevents saturation induced by the cap, allowing a little negative value to provide a superior gradient flow. Furthermore, because the mish_function is non-monotone, we can keep a little negative value to achieve the effect of stabilising the network gradient flow. In terms of actual experimental concerns, a smoother

activation function, such as the Mish_function, can assist us in allowing more information to infiltrate the neural network, to obtain high accuracy of this proposed model. -

$$tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \quad (4.1)$$

$$softmaxplus(x) = \ln(1 + e^{x}) \quad (4.2)$$

$$f(x) = x \,.\tanh(\boldsymbol{softmaxplus}(\boldsymbol{x}) \quad (4.3)$$

In the implementation process:

x: the input data, first passes through the *softplus* function stage, then enters into the Mish function stage after *tanh* operation and is merged.

- **Program code**

Algorithm 1 : Data pre-process and training model

Input : Dataset including masked-faces, unmasked-face and incorrectly-mask

Output : Image illustration indicating the presence of a face

Begin :

for *each image in the three-categories*

1) *Visualise pictures in all categories and theirs labels*
2) *Insert the train image emplacement into the train.txt file.*
3) *Insert the test image emplacement into the file test.txt*

After Build the Yolov4FaceMask model configuration :

batch=64; width=416; height=416; max_batches= 6000.

*[convolutional] filters = (3+5)*3*

[yolo]

Classes=3

end

*Split the data and start training of model./Test the model and evaluate error and map./Extract output results image.**end***

- **Experimental results**

In this paper, the batch size was set to 64 and the subdivisions were set to 16. We cropped the input picture to 416 416 pixels after increasing the size of the input to 512*512 and subsequently to 608 608. We utilised this created model to process the incoming picture. We utilised a momentum of 0.949 and

a weight decay of 0.0005. The learning rate is 0.001 for 600 mini-batches (classes x 2000). We spent 12 hours more on the entire training process using GPU tesla T4 of Google Collaborator. In the presented work, the average precision equal to 83 % with input 416*416, and precision equal to 86.29 % with input-size 512*512, and precision equal to 88.82 % with input-size 608×608 after 6000 iteration and average-loss of 2.8 %, the training result of input-size 416*416 shows in figure 4.6.

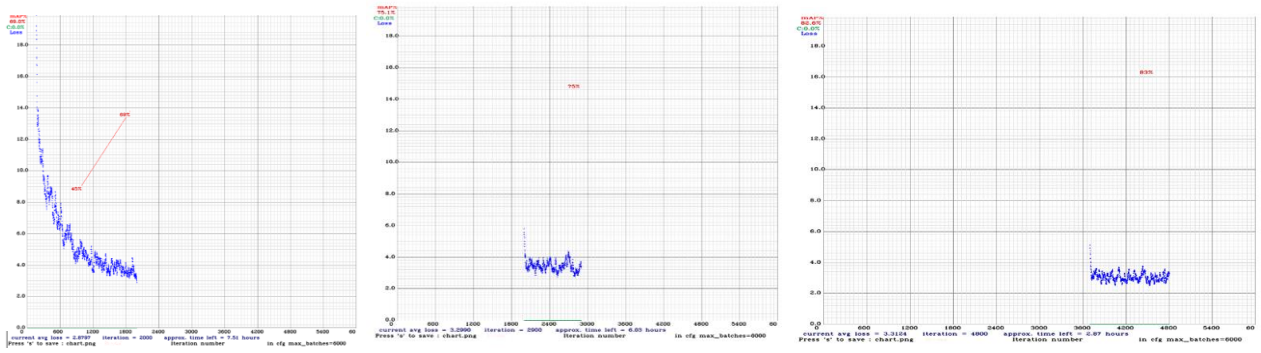


Figure 4. 6: *Detector average loss and mean average precision*

- **Results and discussion on brightness, blurring and proximity in images**

We investigated the performance of our Yolov4FaceMask detectors on photos with problems and barriers such as brightness, blurring, and proximity of faces to camera in the first round of trials.... To show the model's efficacy and accuracy. See Fig 4.8 a), b), d).

We can see that the model produces good outcomes in all of the preceding examples.

In addition, we chose photos with problems and hurdles, such as diverse poses (rotation angles), profiles, and different formats and types of masks, such as transparent masks, to show the model's efficacy and accuracy. See Figure 4.8 a), c), d), h), i).

We can see that the model performs well in all of the aforementioned scenarios, both indoor and outdoor, with the exception of wrongly masking, where we have a lack of accuracy due to the limited amount of photos in this category combined with other groups (masked and unmasked) in the training stage.

Figure 4.8 g) depicts two picture resolutions (low and high). We may deduce that the model accuracy decreased when the image resolution was poor. That was previously stated in Table 4.3.

- **Discuss the results of surveillance video**

We chose indoor and outdoor videos with difficulties and obstacles such as different video resolution, brightness, blurring, different rotation angles of faces, profile, and proximity of the faces... to demonstrate the model's effectiveness and accuracy; the results are shown in fig 4.9 a) for indoor and fig4.9 b) for outdoor.

In terms of accuracy, we can infer that the model precision decreased with low resolution, but in terms of reel time, the model produces good results:

- Input-size **416 × 416 :** FPS = 39.2
- Input-size **512 × 512 :** FPS = 34.4
- Input-size **608 × 608 :** FPS = 29.8

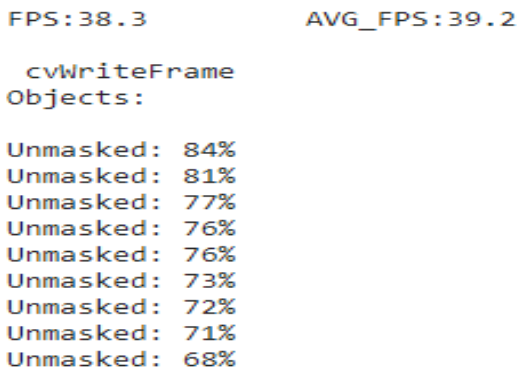


Figure 4. 7 : *Yolov4FaceMask Frame_Per_Seconds*

Table 4.3: Accuracy Of YOLOv4-Face-Model/Different Input-Sizes

Input / Iteration	416×416	512×512	608×608
1000	45.48%	47.87%	51.03%
2000	64.16%	66.28%	67.30%
3000	73.06%	76.51%	72.31%
4000	79.90%	86.27%	83.93%
5000	83.03%	86.15%	87.99%
6000	82.77%	86.29%	**88.82%**

Table 4.4: Proposed YoloV4FaceMask detection Model 416 X 416 Training-Results

Iteration	mAP	Avg Loss
It_1000	45 %	5.190
It_2000	64.16 %	2.870
It_3000	73.05 %	3.200
It_4000	80.6 %	2.900
It_5000	**83.03 %**	2.800
It_6000	82.77 %	**2.800**

TABLE 4.5: COMPARISON OF PROPOSED YOLOV4FACEMASK MODEL WITH STATE OF THE ART MODELS-SIZE

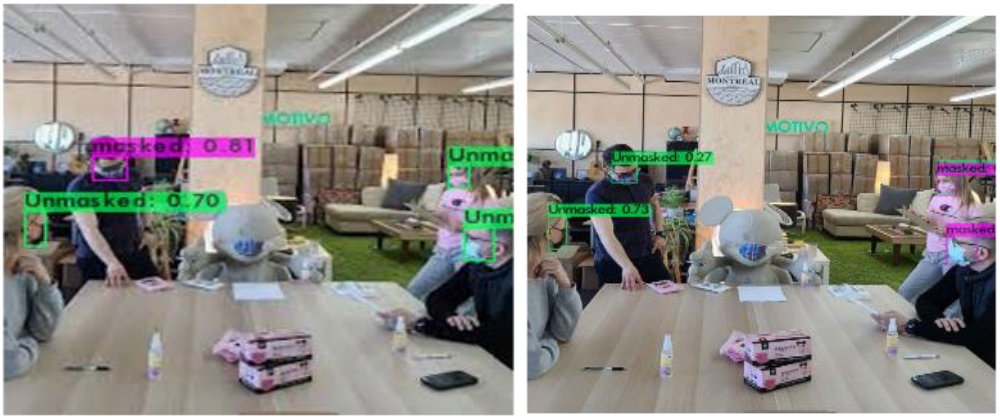
Model name	Publication	Dataset name	Dataset Size	Precision (%)
YOLOv2+ResNet50 [130]	Feb 2021	MMD [145]+FMD [151]	1415	81.000
Face-mask-InceptionV3 [143]	2020	SMFD [152]	1570	99.990
SSDMNV2[150]	Mar 2021	RMFD [40]+PyImageSearch	5521	**92.640**
Face-mask-SRCNet [141]	Sept 2020	-	3835	98.700
Retina_Face_Mask [57]	Jun 2020	Face Mask Dataset [153]	7971	93.400

Yolov4FaceMask (ours)	**2021**	WiderFace [14]+ FMD [151]+ RMFD [146]	**14409**	**88.820**

a) Transparent masks and brightness b) Blurring, proximity indoor c) Profile indoor/outdoor

e) Masks and no masks indoor with pose

f) Correctly, incorrectly and without mask



g) Diffèrent images at low and good resolution

Figure 4. 8: *Visual examples generated by Yolov4FaceMask (Green bounding boxes unmasked faces; red bounding boxes masked faces; orange bounding boxes incorrectly mask faces.)*

Figure 4. 9 : *Real-time surveillance video examples generated by Yolov4FaceMask outdoor (Green bounding boxes represent unmasked faces; Red bounding boxes represent masked faces; orange bounding boxes represent incorrectly mask faces.)*

Conclusion

In this section, we provide a realistic dataset for facemask identification, followed by a novel facemask detector, Yolov4FaceMask, that can contribute to public healthcare. Yolov4FaceMask's architecture is made up of CSPDarknet53 as the backbone, SPP and PAN as the neck, and Yolov3 modules as the heads. The CSPDarknet53 backbone may be utilised for both high and low computing applications. To extract more robust features, we believe that our proposed dataset and model named Yolov4FaceMask could help to prevent the spread of COVID-19 and protect against other infectious diseases, which can be spread by things like speaking at close range, coughing, and sneezing. On our face mask dataset, the suggested model provides state-of-the-art results with an accuracy of 83%. In future work we aspire to: •We expanded our data by include additional photos of masked faces and wrongly masked ones. • Improve the precision of our YoloMaskFace. • Apply the Yolov4FaceMask concept to additional applications.

Chapter 5

Experiment 2: Trying to help reduce the spread of COVID19 based YOLOv4P6FaceMask model and DeepSORT-tracker

Introduction

According to World Health Organisation (WHO) report n.48, COVID-19 illness 2019 has infected over 58 million people and killed over 1.4 million (9 April 2021). With the advent of the COVID-19 coronavirus, many countries, if not all, were forced to implement new social distancing and face mask-wearing guidelines. Because the transmission rate of COVID-19 is growing, governments have required hospitals and other organisations to implement additional infection control measures. The transmission rate, however, may vary depending on the government's efforts and policies. Because COVID-19 is transferred by airdrops and closed contact, governments have begun enforcing new restrictions requiring citizens to avoid sitting too close together and to wear a face mask in order to slow the transmission and spread rate. New coronavirus variations emerged following the relaxation of several nations' adherence to safety laws (India, Nigeria, the United Kingdom, Brazil, and so on), prompting the WHO to encourage the use of Personal Protective Equipment (PPE) among people and in medical treatment. The coronavirus (COVID-19) spreads swiftly in close quarters and busy areas. The proliferation of COVID-19 has an impact on people's lives and the economy. It was identified as a serious public health and economic issue. Countries require instruction and supervision of persons in crowded situations and densely packed public locations to guarantee that face mask rules are followed. This might be implemented using video surveillance systems and deep learning (DL) models. However, most mask detection applications and present mask detection model research focus on tackling the masked face and no masked face identification problems while ignoring incorrectly worn face masks (Table 5.1).

Table 5.1: State-of-art facemask-framework based deep-learning models (***Dm**: detection_model **Tr**: tracking_model, **Imd**: incorrectly_mask_detection **S_ no_ MF**: Save_no_Masked_faces)*

Model-name	***Dm***	***Tr***	***Imd***	***S_ no_ MF***
YOLO v2 + ResNet50	YOLO v2	No.	No.	No.
FaceMask_SRCNet	SRC-Net	No.	Yes.	No.
SSDMNv2	SSD+MobileNetv2	No.	No.	No.
Retina_FaceMask	Retina-Net	No.	Yes.	No.
Goyal_FaceMask	Custom-model	No.	Yes.	No.
Prasad_YOLOv4-FaceMask	YOLO v4	no	No..	No.

The face mask is the focus of this endeavour to reduce COVID-19 transmission and dissemination. The result of the suggested detection model of masked faces/incorrectly masked faces/no masked face region in image or video surveillance. This area is used as an input by the DeepSORT tracker. The Simple Online and Real-time Tracker produces an image or sequence video, but with ID identification

for each face. We save the image of the unmasked face only once in the last stage and after each tracking of the box. (*figure 5.1*).

The superiority of the suggested technique is demonstrated by performance metrics mean average precision (mAP) and mean average recall (mAR), after which the detection and tracking results are compared to previous studies on facemask identification and tracking. This work is assessed using the suggested facemask dataset and publicly available videos/images.

On video sequences of people with masked/incorrectly masked/no masked faces, the detection of masked face sequences and the outcomes (cropped pictures) are assessed.

The sections of this chapter is structured as follows: The first portion is an introduction. Section 2 is devoted to the presentation and examination of the suggested method. Section 3 presents the experimental results and comments, and Section 4 concludes the work with a conclusion.

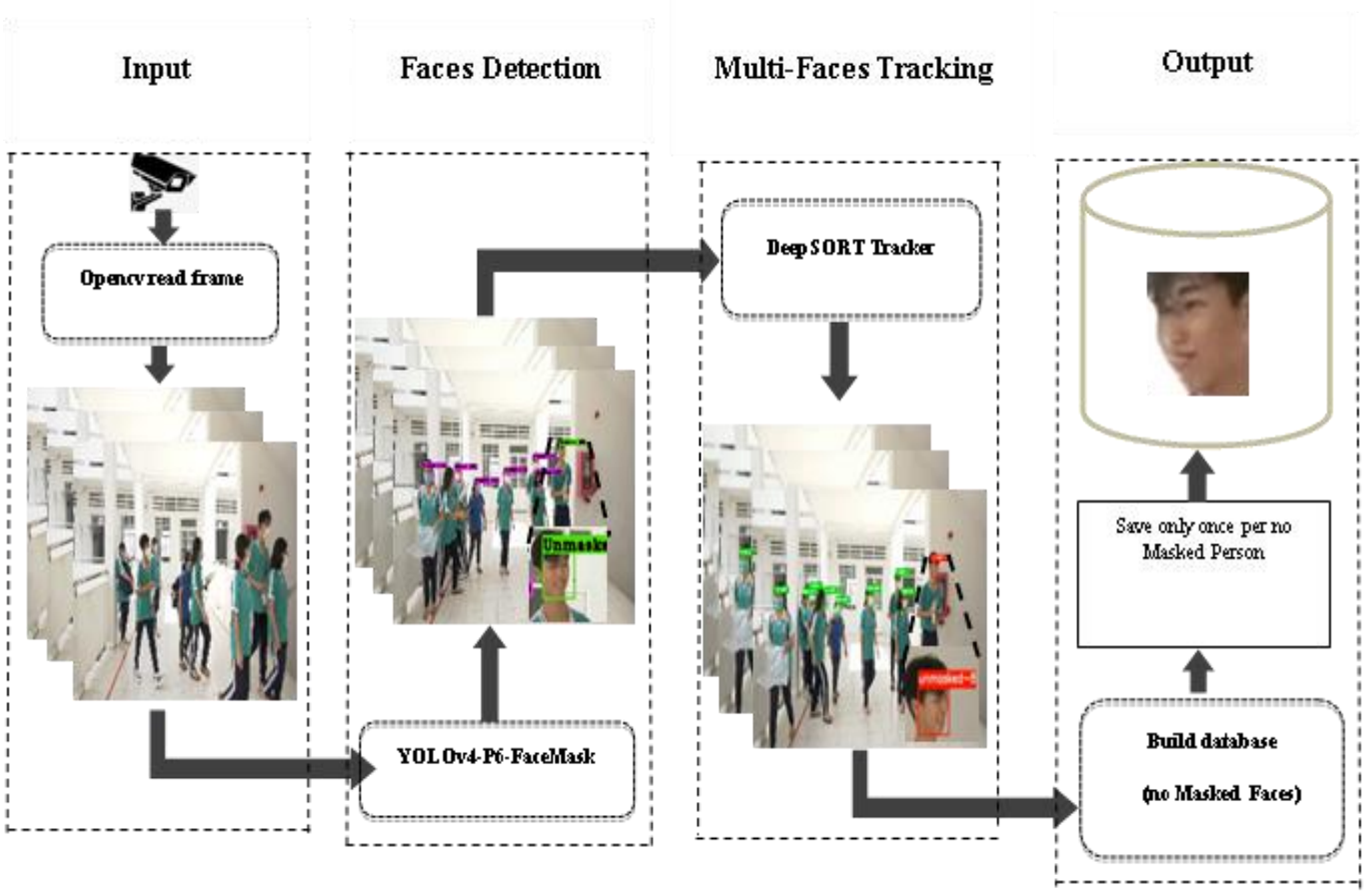


Figure 5. 1: *Our proposed detection and tracking framework*

5.1. Proposed YOLOv4-P6-FaceMask detection and DeepSORT tracking

5.1.1. YOLO detection models

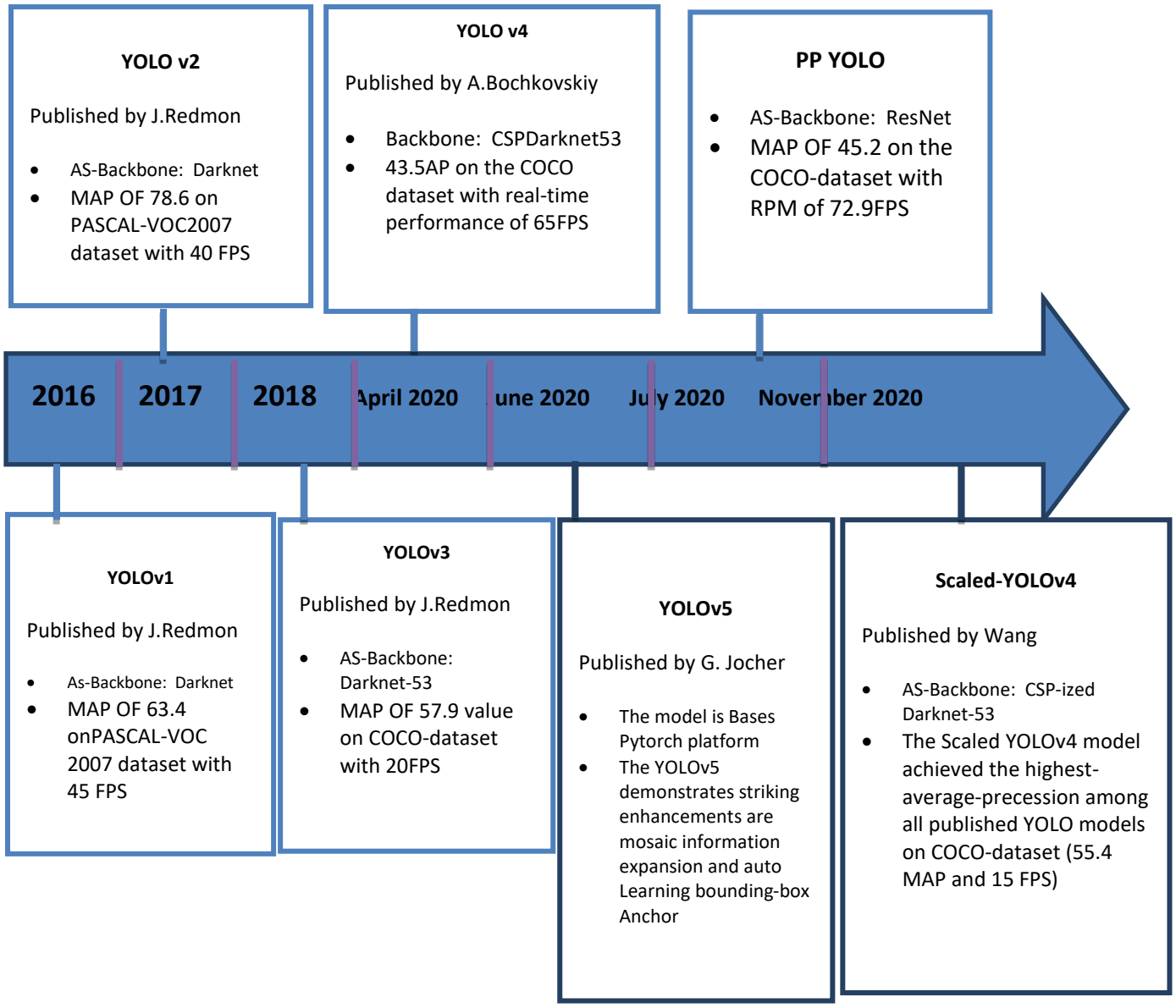


Figure 5. 2 : *Different YOLO models and their average precision*

YouOnlyLookOnce(YOLO) is a real-time, one-stage object identification system that is very accurate. It is designed to be a one-step method for detection and localisation. Following an examination of the input picture, the bounding box and class prediction are done. Backbone, Neck, and Prediction comprise the model's structure (an example of structure). On a GPU-equipped computer, the fastest YOLO architecture can reach 96 Frames per Second (FPS), while the smaller variant, the tiny YOLO,

can reach up to 244 FPS. YOLO'S idea is different from other traditional systems: the bounding box prediction and prediction category are done at the same time. The *figure 5.2* illustrates a state of the art of the different yolo models and their accuracy.

YOLOv4 combines the qualities of YOLOv1, YOLOv2, YOLOv3, and others to achieve the current best in detection speed and detection accuracy compensation.

The combinations between the characteristics of the ResNet-structure and YOLOv3 integrates the residual module into itself and obtains Darknet53. On this premise, YOLOv4 built CSPDarkNet53 in the residual module (input the feature layer and output the top-level feature information), taking into account the higher learning ability of Cross Stage Partial Network (CSP-Net) [24]. The input picture is separated into grids, and a B bound box with a confidence score is defined for each grid cell. The likelihood that an item will exist in each bounding box is represented by reliability, which is defined as:

$$C_S = P_r \times IOU \tag{5.1}$$

Were IOU (Intersection-Over-Union) is a fraction between zero and one, and Average Precision (AP):

$$AP = \sum_{k=1}^{n} P(k)\Delta r(k) \tag{5.2}$$

Where k is the precision at threshold k and Δr(k) is the change in recall.

The Cross-Stage-Partial design was inspired by the DenseNet-architecture, which takes the preceding input and concatenates it with the current input before moving into the dense layer.

Each stage layer of a DenseNet-model has a dense-block and a transition-layer, and each dense-block is composed of k dense-layers.

The output of the i^{th} dense layer will be concatenated with the input of the i^{th} dense layer, with the result being the (i+1) dense layer's input. The following equations illustrate the aforementioned mechanism:

$$x_1 = w_1 * x_0 \tag{5.3}$$

$$x_2 = w_2 * [x_0, x_1] \tag{5.4}$$

$$\vdots$$

Where $*$ is the convolution operator, and [x0, x1, …] means to concatenate x0, x1, …, and w_i and x_i the i^{th} dense layer's weights and outputs, respectively.

The CSP[24] is based on the same principle, except that instead of concatenating the i^{th}output with the i^{th}input, we divided the inputith into two parts, x_0' and x_0", with one part passing through the dense layer x_0' and the second part x_0" being concatenated at the end with the result at the dense layer's output.

This is equal to the following equation in mathematics:

$$x_k = w_k * [x_0, x_1, \dots, x_{k-1}] \tag{5.5}$$

$$x_T = w_T * [x_0, x_1, \dots, x_k] \tag{5.6}$$

$$x_U = w_U * [x_0, x_1, \dots, x_T] \tag{5.7}$$

This will result in different dense layers repeatedly learn copied gradient information.

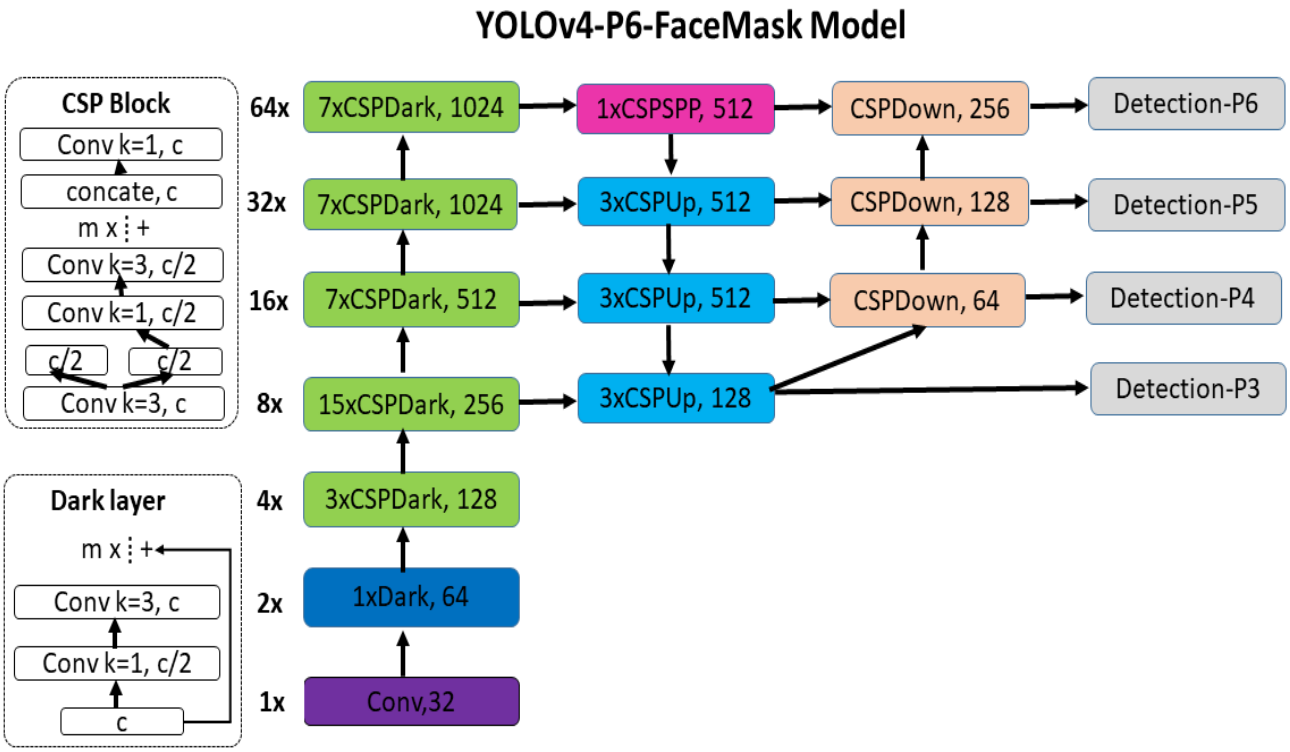


Figure 5. 3: *Architecture of YOLOv4P6FaceMask-model Detection*

Table 5.2: the network architecture of YOLOv4-P6-FaceMask-model

1: parameters	2: anchors	3: backbone
Classes number : 3 Depth multiple: 1.0 Width multiple: 1.0	- [13,17, 31,25, 24,51, 61,45] - [61,45, 48,102, 119,96, 97,189] - [97,189, 217,184, 171,384, 324,451] - [324,451, 545,357, 616,618, 1024,1024]	# [from, number, module, args] [[-1, 1, Conv, [32, 3, 1]], # 0 [-1, 1, Conv, [64, 3, 2]], # 1-P1/2 [-1, 1, BottleneckCSP, [64]], [-1, 1, Conv, [128, 3, 2]], # 3-P2/4 [-1, 3, BottleneckCSP, [128]], [-1, 1, Conv, [256, 3, 2]], # 5-P3/8 [-1, 15, BottleneckCSP, [256]], [-1, 1, Conv, [512, 3, 2]], # 7-P4/16 [-1, 15, BottleneckCSP, [512]], [-1, 1, Conv, [1024, 3, 2]], # 9-P5/32 [-1, 7, BottleneckCSP, [1024]], [-1, 1, Conv, [1024, 3, 2]], # 11-P6/64 [-1, 7, BottleneckCSP, [1024]], # 12

4: head	5: detect
[[-1, 1, SPPCSP, [512]], [-1, 1, Conv, [512, 1, 1]], [-1, 1, nn.Upsample, [None, 2, 'nearest']], [-6, 1, Conv, [512, 1, 1]], [[-1, -2], 1, Concat, [1]], [-1, 3, BottleneckCSP2, [512]], [-1, 1, Conv, [256, 1, 1]], [-1, 1, nn.Upsample, [None, 2, 'nearest']], [-13, 1, Conv, [256, 1, 1]], [[-1, -2], 1, Concat, [1]], [-1, 3, BottleneckCSP2, [256]], [-1, 1, Conv, [128, 1, 1]], [-1, 1, nn.Upsample, [None, 2, 'nearest']], [-20, 1, Conv, [128, 1, 1]], [[-1, -2], 1, Concat, [1]], [-1, 3, BottleneckCSP2, [128]], [-1, 1, Conv, [256, 3, 1]],[-2, 1, Conv, [256, 3, 2]], [[-1, 23], 1, Concat, [1]], [-1, 3, BottleneckCSP2, [256]], [-1, 1, Conv, [512, 3, 1]],[-2, 1, Conv, [512, 3, 2]], [[-1, 18], 1, Concat, [1]], [-1, 3, BottleneckCSP2, [512]], [-1, 1, Conv, [1024, 3, 1]],[-2, 1, Conv, [512, 3, 2]], [[-1, 13], 1, Concat, [1]], [-1, 3, BottleneckCSP2, [512]], [-1, 1, Conv, [1024, 3, 1]],	[[29,33,37,41], 1, Detect, [nc, anchors]], Detect(P6)

5.2. YOLOv4 model scaling technique

Scaling in classic detection models entails changing the model's depth by adding more convolutional layers. The VGGNet, for example, scaled to VGG11, VGG13, VGG16, and VGG19 architectures. However, the scaling strategy now affects the network's depth, breadth, resolution, and structure, resulting in a scaled model, such as ScaledYOLOv4.

To demonstrate the superiority of the chosen model YOLOv4-P6 in terms of backbone, accuracy, real-time performance, and so on, In this experiment section of the thesis, we compare it against the state-of-the-art pedestrian detection algorithms Fast-RCNN, FasterRCNN, YOLOv3, and YOLOv4. The

Scaled YOLOv4 detection algorithm is used in the suggested solution, as shown in figure 5.1, to recognise faces in single images, real-time video, or online cameras.

This chapter will not for discuss the history of previous versions (see chapters 3 and 4) of YOLO (YOLOv1, YOLOv2, and YOLOv3). We trained a customised YOLOv4P6 model for facemask detection and localisation using an enormous dataset of over 18000 photos classified as "masked," "incorrectly mask," and "unmasked (no-masked)."

The suggested dataset (a refinement of the dataset proposed in Chapter 4) is derived from:

- WiderFace-dataset(all no masked faces and part of MaskedFaces)
- FMD(FaceMask-dataset) (masked-faces and incorrectly masked faces)
- RMFD(RealFaceMask-dataset)

Following that, we use the face set identified by YOLOv4-P6-FaceMask as an input to the Deep SORT tracker. For each first detection, the Deep SORT generates a unique identification number ID. If the first detection is an uncovered face or erroneous mask wear, we crop the face and save it in an OpenCV file. Deep SORT then follows each of the faces as they move across frames in a movie, assigning each one a unique ID.

5.3. COVID19 YOLOv4P6FaceMask-model

Figure 5.3 depicts the suggested facemask detector with the re-design of YOLOv4 to YOLOv4CSP to achieve the optimal speed and accuracy TradeOff. Architecture of a Network .

The YOLOv4 detector's network architecture used CSPDarknet53 as a backbone, YOLOv3 as a heads, and SPP and PAN as a neck.

We discuss architectural scaling in order to develop a real-time facemask detection approach. We implement several architectural changes to improve the performance of YOLO v4.

The proposed network architecture of YOLOv4P6FaceMask-model is illustrated in *figure 5.3* and table 5.2. The convolution-layer is responsible of extracting features from the input using kernel (conv-filter).

5.3.1. The backbone :

The Backbone of proposed YOLOv4P6FaceMask-model can be shared into two parts: the kernel-block (ConvolutionBuildingBlock) and the CSPBlock modules (see table 3). The number of Residual- layers owned by each stage in CSPBlock is 1_2_8_8_4 respectively This means that (**1**x CSP-Block _ **2**x CSP-Block _ **8**xCSP-Block_ **8**xCSP-Block_**4**xCSP-Block).

The first CSPStage is converted to original DarknetResidualLayer.

5.3.2. The neck :

In order to minimise calculations, the PAN architecture is CSPized (converted to CrossStagePartial connections form).

5.3.3. The SPP:

It was initially put in the centre of the neck; the same concept has been adopted and applied in CSPPAN as well.

5.4. Tracking of Object.

MultiObjectTracking (MOT) is the challenge of tracking the trajectory of various objects in a sequence, often a video. With the recent emergence of DL, the algorithms that provide a solution to this problem have profited from the representational capacity of DL models. We focus on MOT-based DL techniques and how StateOfTheArt MOTs employ DL approaches based on a survey article [154].

On the MOT-16 dataset [160], the autors evaluate the accuracy and performance of DeepSORT [159]. This dataset assesses tracking performance on seven difficult test video sequences, including frontal-view situations with moving cameras and top-down surveillance configurations.

The tracker is compared with StateOfTheArt _of_tracking_methods (illustrate in table 8):

- AMIR [155]: Savarese.S and others. proposed a racking the untrackable: Learning to track multiple cues with long-term dependencies
- IA [156]: Tan.CC and others. proposed an OnlineMOT with InstanceAware Tracker and DynamicModelRefreshment
- SORT (the SimpleOnlineReal-timeTracking) technique [157]: Ge.Z and others. proposed a Simple online and Real-Time Tracking method.

- EAMTT [158]: Matilla.S and others proposed a tracker named OnlineMulti-TargetTracking with strong and weak detections

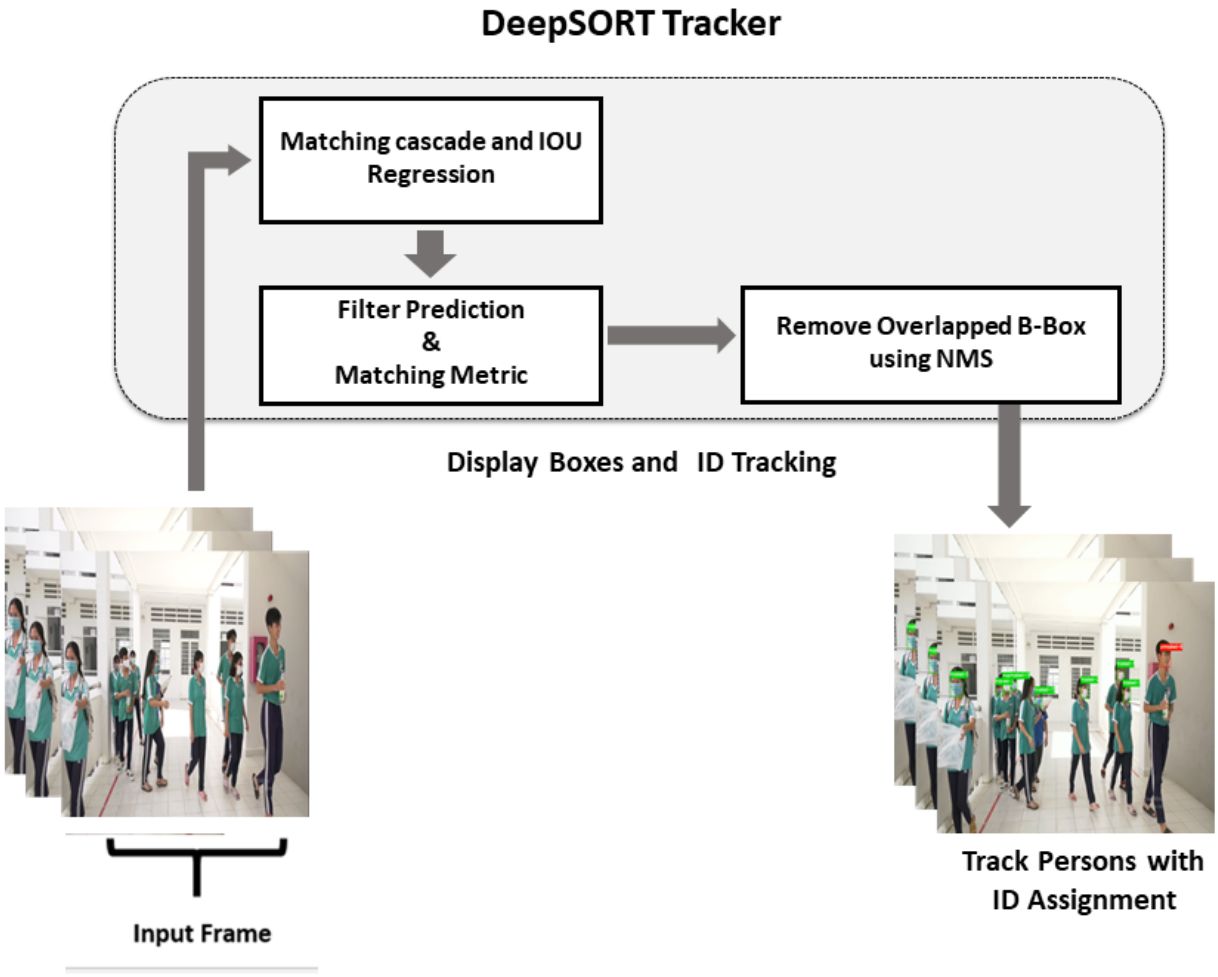


Figure 5. 4: *DeepSORT_Face_Mask_Tracking*

Table 5.3 StateOf Art Trackers (***MOT****: Mot16Dataset [41],* ***MT****: MostlyTracked,* ***ML****: MostlyLost,* ***ID****: IdentificationNumber,* ***Acc:****Accuracy,* ***Pr****:Precision*)

Tracking-Model	MOT_Acc	MOT_Pr	MT %	ML%	ID s
ESNN	33.4	72.1	11.7	30.9	1598
AMIR	47.2	75.8	14	41.6	774
IA	48.8	75.7	15.8	38.1	906
SORT	59.8	79.6	25.4	22.7	1423
DeepSORT	**61.4**	**79.1**	**32.8**	**18.2.**	**781**
EAMTT	52.5	78.8	19	34.9	910

5.5. Faces Tracking

In this step, we use the DeepSORT approach to track faces and ID assignments for each box (figure 5.4). DeepSORT is an online object tracking technique that uses both information about the monitored items' manifestation and the bounding box characteristics of the detection results to connect detections in the frame at time t+1 with tracked objects at time t. As a result,

DeepSORT does not have to process the entire video at once. To create predictions about the current frame, it only considers information from the current and prior frames. The method is allocated to each bounding box indicating a pedestrian with a greater confidence value than a given threshold at the start of the series, i.e. in frame number one. The Hungarian method is a combinatorial optimisation procedure that is used to allocate detections in a new frame to existing tracks in order for the assignment cost function to approach the global minimum.

The cost-function involves the Mahalanobis-spatial-distance $d^{(1)}(i,j)$of the detected bounding-box from the position predicted according to the known position at time t of that object, and a visual distance $d^{(2)}(i,j)$that considers the appearance of the detected object and the history of the appearance of the tracked-object. The expression of Mahalanobis $d^{(1)}(i,j)$ is given by

$$d^{(1)}(i,j) = \left(d_j, y_i\right)^T S_i^{-1}\left(d_j - y_i\right) \qquad (5.8)$$

: λ: is a parameter that can be set to determine the influence of the visual distance $d^{(2)}(i,j)$and the Mahalanobis $d^{(1)}(i,j)$. The cost function c_(i,j) of assigning a detected object j to a track i is given by the expression:

$$c_{i,j} = \gamma d^{(1)}(i,j) + (1-\gamma) d^{(2)}(i,j) \qquad (5.9)$$

Where y_i represent the mean and Si represent the covariance matrix bounding box observations for the ith track d_j represents the jth detected bounding box.

The expression of visual $d^{(2)}(i,j)$that relies on appearance

feature descriptors:

$$d^{(2)}(i,j) = min\left\{1 - r_j^T r_k^{(i)} \middle| r_k^{(i)} \epsilon\, \Re\right\} \qquad (5.10)$$

Where rj is the appearance descriptor extracted from the part of the image within the jth detected bounding box; $\mathscr{R}_i$ is the set of last 100 appearance descriptors $r_k^{(i)}$ associated with the track i.

The cosine distance uses by $d^{(2)}(i,j)$ measure between the jth detection and ith track in the current detection to select the track where visually the most similar detection is

previously found.

When there are more detections in a frame than currently monitored people, new track IDs are created.

The detection cannot be attributed to any track because it is too far away from any track or does not look visually similar to any prior detection.

5.6. Detection results

Because mask detection is fundamentally a classification and localisation problem, it is assessed using standard metrics such as TruePositive (TP), TrueNegative (TN), FalsePositive (FP), and FalseNegative (FN), which are defined as accuracy and recall:

$$\text{Precision} = TP/(TP+FP) \tag{5.11}$$

$$\text{recall} = TP/(TP+FN) \tag{5.12}$$

Furthermore, the evaluation employs IntersectionOverUnion (IoU), which provides the ratio of the overlapping area of the predicted boxes to the matching ground truth; higher IoU values indicate more accurate localization, so IoU = 1 is the best case. Combined with the IoU value, AP50 and AP75 are applied to report the AveragePrecision at IoU = 0.5 and IoU = 0.75 levels. mAP and mAR represent the means of the 10 precision and recall values at IoU, ranging from 0.5 to 0.95 with an interval of 0.05 for detailed performance in each category to further evaluate the overall performance of the facemask detection model. We selected a BatchSize of 64 and a subdivisions of value 8: depending on the performance of used GPU. The InputImage is set to width×height=1280×1280 pixels. We used this prepared model to process the InputImage. A Momentum of value 0.96, a batch_normalization of value = 1, ActivationFunction = mish_function ,weight_decay of value = 0.0004 were used. The

learning_rate is Lr=0.001 for 600 mini-batches-size ; which is calculated using the following method C*2000=6000. We spent 14 hours extra on model training utilising Google-Collaborator's GPU Tesla-T4. After 6000 iterations, the mean average accuracy was 93% with an input-size of 12801280 and an average loss of 1.8%. All training settings and outcomes are included in Tables 5 and 6.

Table 5.4 Parameters of YOLOv4P6FaceMask Model

Parameters	**Value**
TheWidth	1280
TheHeight	1280
theMomentum	0.960
theLearning rate	0.00100
theBatch_size	64
TheSubdivisions	8
theActivation function	MishFunction
TheClasses	3
theMini-batches	600
theWeight decay	0.00040

Table 5.5 YOLOv4P6FaceMask Model 1280x1280 - Results of model Training

Iteration	**mAP%**	**Avg_Loss**
It_1000	47	4.190
It_2000	67.16	2.870
It_3000	79.05	2.20
It_4000	87.6	1.90
It_5000	89.03	1.80
It_6000	93.02	1.80

5.6.1. Comparison of proposed model with State-Of-The-Art.

For a typical evaluation of our proposed model, we trained various cutting-edge models on our suggested dataset using the same platform implementation (tesla T4). The classification accuracy and real-time performance of the YOLOv4P6FaceMask model are compared to the classification accuracy of YOLOv4, YOLOv3, Faster RCNN, EfficientDet, and RenitaFaceMask [60]. Table 7 compares the trained models in the proposed face mask dataset. We discovered that YOLOv4-P6-FaceMask can

outperform Faster RCNN by 13% and can outperform YOLOv3 and YOLOv4 by 11% and 7%, respectively, and can outperform EfficientDet and RenitaFaceMask by 7% and 4% in term of mean average precision (mAP). The results are illustrated in table 11 and figure 9.

Table 5.6 YOLOv4P6FaceMask-modelresult comparison with StateOfTheArt.

Model Name	**AP 50%**	**AP 75%**	**Map %**	**mAR %**	**FPS%**
Faster_RCNN	80	72	80	70	35
YOLOv3	82	75	82	80	30
YOLOv4 416	81	74	80	75	57
YOLOv4 512	83	77	83	83	50
YOLOv4 608	86	80	86	82	44
EfficientDet	89	84	87	87	32
RenitaFaceMask	91	88	89	89	29
YOLOv4P6FaceMask (**Ours**)	**94.0**	**90.0**	**93.0**	**92.0**	**35**

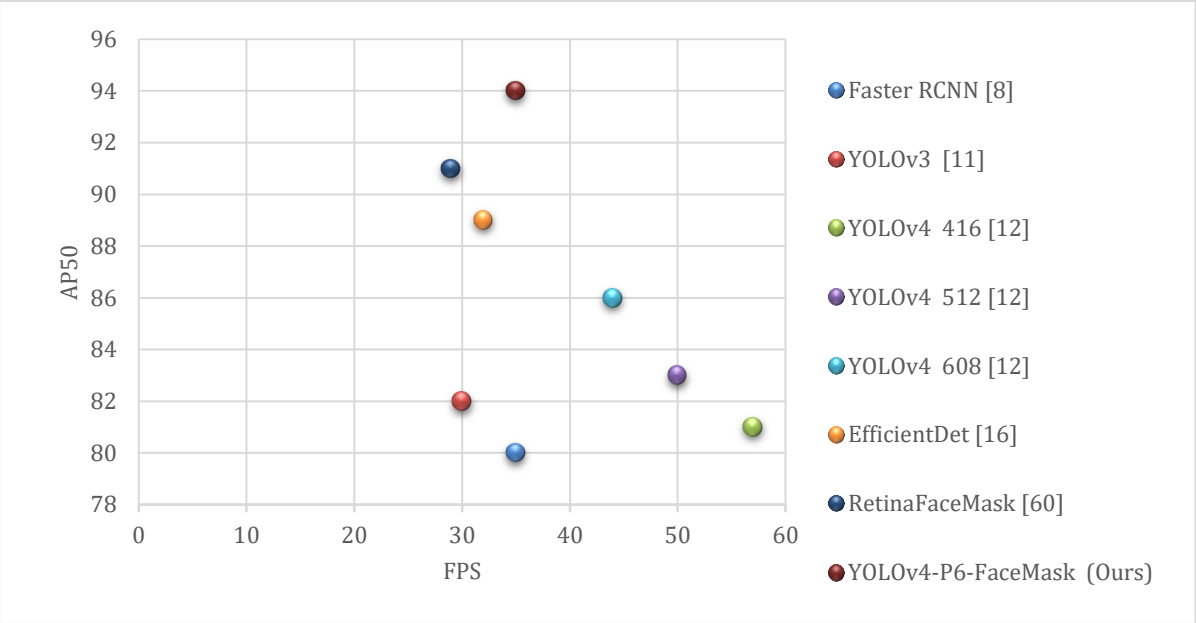


Figure 5. 5*: The proposes YOLOv4-P6-FaceMask comparison with state-of-the-art models*

5.6.2. discussion of obtained results on term of brightness/blurring/noise/proximity in images of proposed YOLOv4P6FaceMask-model

In the 1st part illustrated in *figure 5.7*, the experiments that we conducted is to validate the effectiveness and model accuracy (GreenBoxes: unmasked_faces without IdentificationNumberID; RedBoxes: masked_faces without IdentificationNumberID; orangeoxes: incorrectly_masked_faces without IdentificationNumberID).

We investigated the performance of our YOLOv4-P6-FaceMask detector on photos with problems and barriers such as brightness, blurring, noise, and closeness faces to the camera... to demonstrate the model's efficacy and correctness. We can see that our suggested approach produces outstanding outcomes in all of the issues discussed earlier.

In another part of the experiments, we explored the performance of the YOLOv4-P6-FaceMask detector on images that contained difficulties and obstacles such as different poses (rotation angles), profiles, and different formats and types of masks such as transparent masks... to demonstrate the model's effectiveness and accuracy.

5.6.3. Results and discussion of surveillance video

Indoor and outdoor videos are chosen with problems and hurdles such as varying video resolution, brightness, blurring, different rotation angles of faces, profile, and closeness of the faces... to demonstrate the model's efficacy and accuracy. Figure 5.8 depicts the outcomes.

In terms of the suggested model's accuracy, we can see that the precision decreases when the picture resolution is low, and in terms of real-time performance, we can see that the model provides an acceptable performance of 35 frames per second.

5.6.4. Tracking and output results

We have tested with a number of indoor/outdoor videos in order to achieve the accurateness of the model offered in this experiment section. Figure 5.8 shows the model's tracking performance in both indoor and outdoor settings, with respectable YOLOv4-P6-FaceMask model accuracy. Figure 5.6 illustrates the outcome of unmasked/inadvertently-masked faces that were obtained and saved in a file. In all the video sequences, there is just one inaccuracy that stands out: a face that is covered.

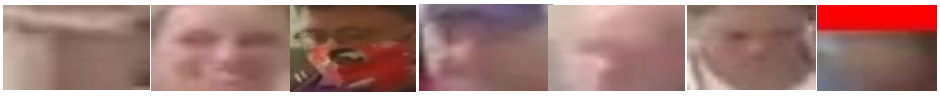


Figure 5. 6: *Outputl Results of crop/save unmasked/incorrectly_masked_faces*

5.7. Implementation platform/used_libraries :

The Python3 programming language is utilised on ONLINE Google-Collab to develop the facemask detection and tracking framework. We used a DarknetProject to train the YOLOv4P6FaceMask detector in the first step.

The detection model was trained and tested on a single Google-Collab Tesla-T4 GPU. Darknet, Keras, Os, OpenCv, NumPy, MatPlotLib, and pillow were the libraries utilised in the implementation procedures.

5.8. Limits of the work :

Among model's limitations:

- The weakness in the detection model's precision for those who mistakenly wear medical masks. This is because there weren't lots of pictures of faces that had been incorrectly_masked during training.
- The lack of a standardised database built specifically for this purpose prevents the face-tracking model from being compared to other models.


Unmasked: 0.94
masked: 0.95
Unmasked: 0.97


Unmasked: 0.35
masked: 0.40
masked: 0.87
masked: 0.69
masked:


Unmasked: 0.97


masked: 1.00
masked: 1.00
masked: 0.64
masked: 0.99
masked: 0.98
masked: 0.97


masked: 0.81
Unmasked: 0.70


masked: 0.90
masked: 0.94
masked: 0.99


Unmasked: 0.85
masked: 1.00
masked: 0.85
masked: 1.00

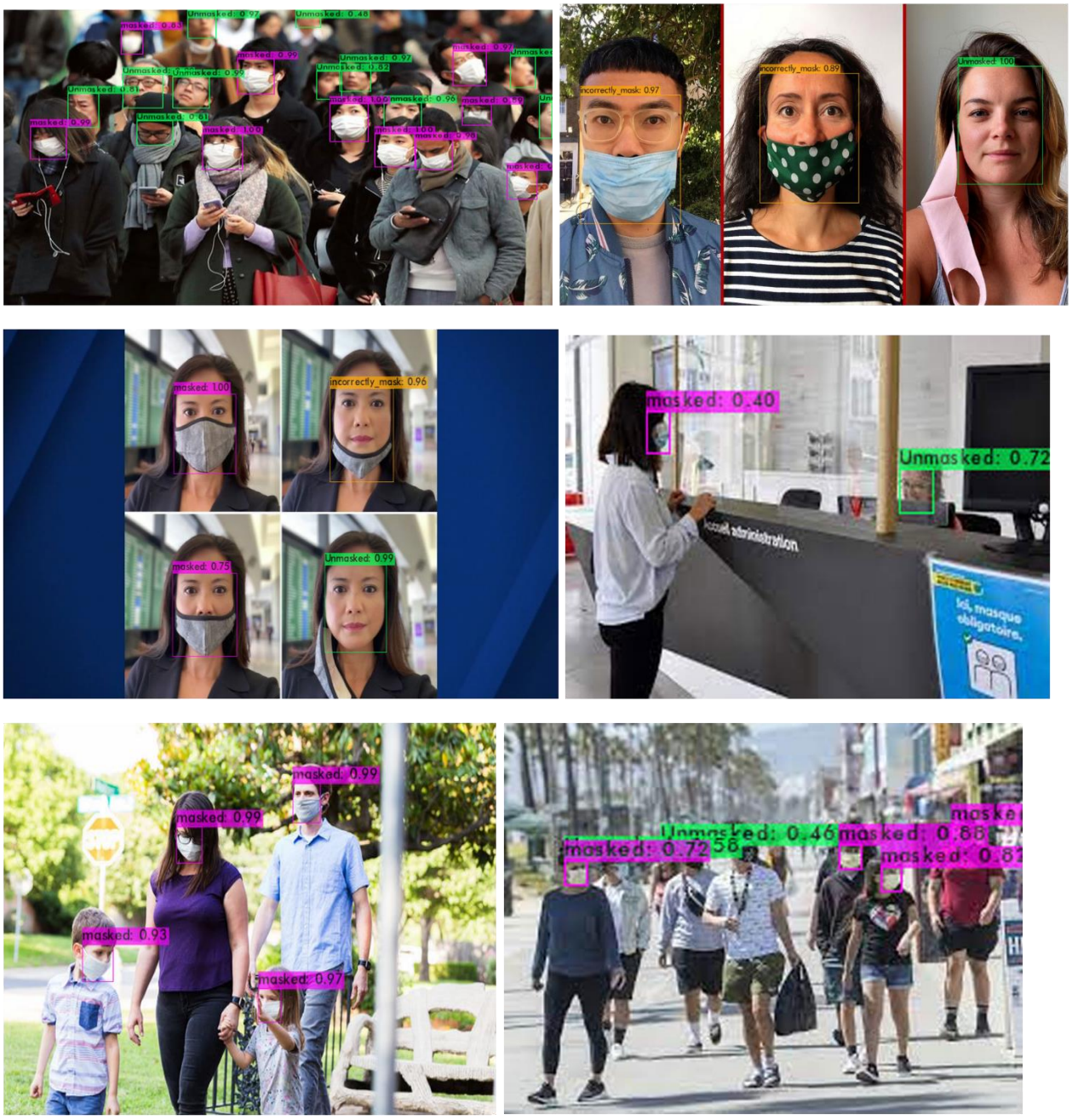
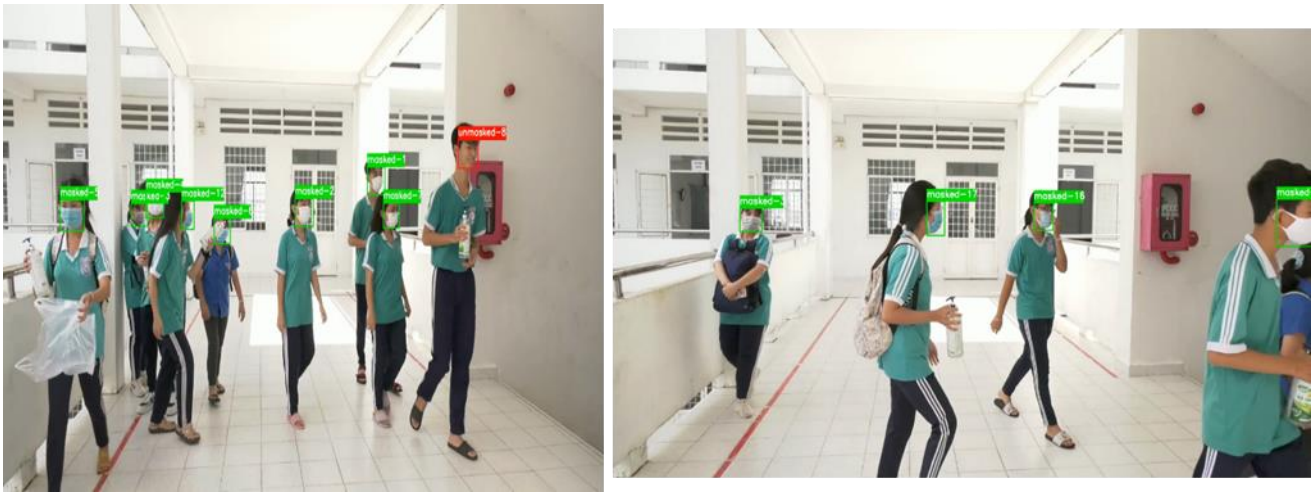


Figure 5. 7: *YOLOv4P6FaceMask detector Images results. (RedBoxes: masked_faces, GreenBoxes: in-masked_faces, orangeBoxes: incorrect_masked_faces).*



a) ndoor-School students

b) Indoor-Airport

c) low conditions videos with bruit

Figure 5. 8: *YOLOv4P6FaceMask-model and DeepSort_tracker videos examples (reel time).*

Conclusion

In the chapter number five, we proposed a new framework of detection and tracking of medical maskes, the proposed work is a automatic TwoStage framework based our YOLOv4P6FaceMask-model and DeepSORTtracker. In addition, we suggested a new dataset of face (with medical mask/without medical mask/ with medical mask incorrectly) with 18000 images, the 1st phase or stage is the training of YOLOv4P6FaceMask face-mask detection-model, which can contribute to public-healthcare. The network-architecture of the proposed DetectionModel uses CSPizedCSPDarknet53 as backbone, CSP_SPP and PAN as the neck and CSPizedYoloV3 module as the heads and we scaling-up the model network and mish as ActivationFunction. In the 2nd stage, the DeepSORTtracker is used to track faces, the tracker helps us to crop and save faces only once per person in all sequences. In order to extract, more robust features we believe that our work propose a dataset, a model named YOLOv4P6FaceMask-model and could contribute to preventing the COVID19 from pervasion for protect against other infectious diseases; which can be prevalent by such things as speaking at close range, coughing, sneezing. The proposed model achieves state-of-the-art results on face-mask datasets, with an accuracy of 93%, a MeanAverageRecall of 92%, a real-time speed of 35 fps with input 1280×1280, and average_loss of 1.8%.

Biometrics is an exciting and complex field. It attempts, utilising frequently sophisticated mathematical techniques, to distinguish between individuals, forcing us to work in a context of great diversity. This diversity is also found in the considerable number of algorithms that have been proposed for facial recognition.

In this thesis, we are interested in the problem of localization and detection-based Deep Learning. A literature research was conducted to learn about several deep-learning models capable of conducting real-time object identification and recognition. An analysis of the different techniques developed in recent years has been presented, in order to highlight the particularities as well as the advantages and disadvantages of each of them.

We have proposed different techniques aimed at taking into account the specificities of our problem. We have developed an approach based on the extraction of regions of interest using Yolov4, and ScaledYoloV4. However, these object detection techniques have drawbacks such as non-exact detection and the need for a large database of fairly large volumes. To remedy these problems, we have proposed a new large database of face-masked, non-masked, and incorrectly masked faces, this database has been provided with both rich annotations, but also a rigorous testing protocol, and a set of benchmark algorithm performance, which allows the community to come up with other algorithms and compare them effectively.

Two models of detection are proposed namely respectively YOLOV4FaceMask and YoloV4-p6-FaceMask. A new technique of detecting and localization and tracking of faces is offered with the combination of two models ScaledYolov4 detection model and the DeepSORT tracker. An other new technique of social distancing and face mask detection is proposed.

In the future, we want to:

- Improve the accuracy of our YOLOv4-P6-FaceMask and Yolov4FaceMask by employing various recent optimisation methods.
- Create apps with the YOLOv4-P6-FaceMask model (social-distance applications, android studio applications, java apps …)
- try to companies between our YOLOv4-P6-FaceMask model and another recent tracker.
- Using the collected photos from the model to determine if the individual is unwell or has sickness symptoms based on his facial features

- Implement our recommended proposed technique in an embedded-system. (Raspberry-Pi, Drone..…).

BIBLIOGRAPHIE

[1] Frédéric, E. (2007). Techniques et technologies de localisation avancées pour terminaux mobiles dans les environnements indoor [Doctoral dissertation, Université Joseph Fourier - Grenoble]. Institutional Repository. https://theses.hal.science/tel-00136064v1

[2] Idrus, S. Z. S., Cherrier, E., Rosenberger, C., & Bours, P. (2014). Soft biometrics for keystroke dynamics: Profiling individuals while typing passwords. Computers & Security, 45, 147–155. https://doi.org/10.1016/j.cose.2014.05.008

[3] Beveridge, R., & Kirby, M. (2005). Biometrics and face recognition. IS&T Colloquium, 25.

[4] Ross, A. A., Jain, A. K., & Flynn, P. (2007). Handbook of biometrics. Springer. https://doi.org/10.1007/978-0-387-71041-9

[5] Nandakumar, K., Jain, A. K., & Ross, A. A. (2011). Introduction to biometrics. Springer. https://doi.org/10.1007/978-0-387-77326-1

[6] Pentland, A., & Choudhury, T. (2000). Personalizing smart environments: Face recognition for human interaction. IEEE Computer [Special issue on Biometrics]. http://www.media.mit.edu/-pentland

[7] Buyssens, P. (2006). Fusion de différents modes de capture pour la reconnaissance du visage appliquée aux e_transactions [Doctoral dissertation, Université de Caen]. Institutional Repository. https://theses.hal.science/tel-01079134/

[8] Hietmeyer, R. (2000). Biometric identification promises fast and secure processing of airline passengers. International Civil Aviation Organization Journal, 55(9), 10–11.

[9] Li, S. Z., & Jain, A. K. (2005). Handbook of face recognition. Springer-Verlag. https://static.googleusercontent.com/media/research.google.com/fr//pubs/archive/36368.pdf

[10] Khan, F. H., Pasha, M. A., & Masud, S. (2021). Advancements in microprocessor architecture for ubiquitous AI—An overview on history, evolution, and upcoming challenges in AI implementation. Micromachines, 12(6), 665. https://doi.org/10.3390/mi12060665

[11] Mahesh, B. (2020). Machine learning algorithms—A review. International Journal of Science and Research (IJSR), 9, 381–386. DOI: 10.4236/jdaip.2023.114021

[12] Simeone, O. (2018). A very brief introduction to machine learning with applications to communication systems. IEEE Transactions on Cognitive Communications and Networking, 4(4), 648–664. https://doi.org/10.1109/TCCN.2018.2881442

[13] Abirami, S., & Chitra, P. (2020). Energy-efficient edge-based real-time healthcare support system. In Advances in computers (Vol. 117, pp. 339–368). Elsevier. https://doi.org/10.1016/bs.adcom.2019.09.007

[14] Yang, S., Luo, P., Loy, C.-C., & Tang, X. (2016). Wider face: A face detection benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5525–5533). https://doi.org/10.1109/CVPR.2016.596

[15] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition (pp. 248–255). IEEE. https://doi.org/10.1109/CVPR.2009.5206848

BIBLIOGRAPHIE

[16] Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine Learning (pp. 160–167). ACM. https://doi.org/10.1145/1390156.1390177

[17] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems (pp. 1097–1105). https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[18] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. International Journal of Computer Vision (IJCV), 115(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[19] Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In European Conference on Computer Vision (pp. 818–833). Springer. https://doi.org/10.48550/arXiv.1311.2901

[20] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1–9). https://doi.org/ 10.1109/CVPR.2015.7298594

[21] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770–778). https://doi.org/ 10.1109/CVPR.2016.90

[22] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv preprint arXiv:1602.07261.

[23] Zhou, T., Zhao, Y., & Wu, J. (2021). Resnext and res2net structures for speaker verification. In 2021 IEEE Spoken Language Technology Workshop (SLT). IEEE. https://doi.org/ 10.1109/SLT48900.2021.9383531

[24] Wang, C.-Y., Liao, H.-Y. M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., & Yeh, I.-H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPR Workshop). https://doi.org/10.48550/arXiv.1911.11929

[25] Koonce, B. (2021). EfficientNet. In Convolutional neural networks with Swift for TensorFlow (pp. 109–123). Apress. https://doi.org/10.1007/978-1-4842-6168-2

[26] Liang, M., & Hu, X. (2015). Recurrent convolutional neural network for object recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3367–3375). https://doi.org/10.1109/CVPR.2015.7298958

[27] Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1440–1448). https://doi.org/10.1109/ICCV.2015.169

[28] Jiang, H., & Learned-Miller, E. (2017). Face detection with the faster R-CNN. In 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017) (pp. 650–657). IEEE. https://doi.org/ 10.1109/FG.2017.82

[29] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2117–2125). https://doi.org/ 10.1109/CVPR.2017.106

BIBLIOGRAPHIE

[30] Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. In Advances in Neural Information Processing Systems (pp. 379–387). https://doi.org/10.48550/arXiv.1605.06409

[31] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 779–788). https://doi.org/10.1109/CVPR.2016.91

[32] Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7263–7271). https://doi.org/10.1109/CVPR.2017.690

[33] Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[34] Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

[35] Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2021). Scaled-YOLOv4: Scaling cross stage partial network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 13029–13038). https://doi.org/10.48550/arXiv.2011.08036

[36] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In European Conference on Computer Vision (pp. 21–37). Springer. https://doi.org/10.48550/arXiv.1512.02325

[37] Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. arXiv preprint arXiv:1708.02002.

[38] Tan, M., Pang, R., & Le, Q. V. (2020). EfficientDet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10781–10790). https://doi.org/10.1109/CVPR42600.2020.01079

[39] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In European Conference on Computer Vision (pp. 740–755). Springer. https://doi.org/xxxx

[40] Du, H., Shi, H., Liu, Y., Wang, J., & Mei, T. (2022). The elements of end-to-end deep face recognition: A survey of recent advances. ACM Computing Surveys (CSUR), 54(10s), 1–42. https://doi.org/xxxx

[41] Li, H., Lin, Z., Shen, X., Brandt, J., & Hua, G. (2015). A convolutional neural network cascade for face detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5325–5334). https://doi.org/xxxx

[42] Shi, X., Shan, S., Kan, M., Wu, S., & Chen, X. (2018). Real-time rotation-invariant face detection with progressive calibration networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2295–2303). https://doi.org/xxxx

[43] Zeng, D., Liu, H., Zhao, F., Ge, S., Shen, W., & Zhang, Z. (2019). Proposal pyramid networks for fast face detection. Information Sciences, 495, 136–149. https://doi.org/xxxx

[44] Chen, D., Hua, G., Wen, F., & Sun, J. (2016). Supervised transformer network for efficient face detection. In Proceedings of the European Conference on Computer Vision (Vol. 9909, pp. 122–138). Springer. https://doi.org/10.48550/arXiv.1607.05477

BIBLIOGRAPHIE

[45] Ge, S., Li, J., Ye, Q., & Luo, Z. (2017). Detecting masked faces in the wild with LLE-CNNs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2682–2690). https://doi.org/10.1109/CVPR.2017.53

[46] Zhang, C., Xu, X., & Tu, D. (2018). Face detection using improved faster R-CNN. arXiv preprint arXiv:1802.02142.

[47] Wang, H., Li, Z., Ji, X., & Wang, Y. (2017). Face R-CNN. arXiv preprint arXiv:1706.01061.

[48] Jiang, H., & Learned-Miller, E. (2017). Face detection with the faster R-CNN. In Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition (pp. 650–657). https://doi.org/10.1109/FG.2017.82

[49] Sun, X., Wu, P., & Hoi, S. C. H. (2018). Face detection using deep learning: An improved faster R-CNN approach. Neurocomputing, 299, 42–50. https://doi.org/10.1016/j.neucom.2018.03.030

[50] Najibi, M., Singh, B., & Davis, L. S. (2019). FA-RPN: Floating region proposals for face detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7723–7732). https://doi.org/10.48550/arXiv.1812.05586

[51] Hao, Z., Liu, Y., Qin, H., Yan, J., Li, X., & Hu, X. (2017). Scale-aware face detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 6186–6195). IEEE. https://doi.org/10.48550/arXiv.1706.09876

[52] Qin, H., Yan, J., Xiu, L., & Hu, X. (2016). Joint training of cascaded CNN for face detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3456–3465). IEEE. https://doi.org/10.1109/CVPR.2016.376

[53] Zhang, S., Zhu, X., Lei, Z., Wang, X., & Li, S. Z. (2018). Detecting face with densely connected face proposal network. Neurocomputing, 284, 119–127. https://doi.org/10.1016/j.neucom.2018.01.038

[54] Tang, X., Du, D. K., He, Z., & Liu, J. (2018). PyramidBox: A context-assisted single shot face detector. In Proceedings of the European Conference on Computer Vision (pp. 797–813). Springer.

[55] Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., & Li, S. Z. (2017). FaceBoxes: A CPU real-time face detector with high accuracy. In Proceedings of the IEEE International Joint Conference on Biometrics (pp. 1–9). IEEE.

[56] Chi, C., Zhang, S., Xing, J., Lei, Z., Li, S. Z., & Zou, X. (2019). Selective refinement network for high performance face detection. In Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 8231–8238. https://doi.org/10.1609/aaai.v33i01.33018231

[57] Deng, J., Guo, J., Ververas, E., Kotsia, I., & Zafeiriou, S. (2020). RetinaFace: Single-shot multi-level face localisation in the wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5203–5212). IEEE. DOI: 10.1109/CVPR42600.2020.00525

[58] Li, J., Wang, Y., Wang, C., Tai, Y., Qian, J., Yang, J., Wang, C., Li, J., & Huang, F. (2019). DSFD: Dual shot face detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5060–5069). IEEE.

[59] Zhang, S., Wen, L., Bian, X., Lei, Z., & Li, S. Z. (2018). Single-shot refinement neural network for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4203–4212). IEEE. https://doi.org/10.48550/arXiv.1711.06897

[60] Yang, S., Luo, P., Loy, C. C., & Tang, X. (2015). From facial parts responses to face detection: A deep learning approach. In Proceedings of the IEEE International Conference on Computer Vision (pp. 3676–3684). IEEE. https://doi.org/10.1109/ICCV.2015.419

[61] Ranjan, R., Patel, V. M., & Chellappa, R. (2019). HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(1), 121–135. https://doi.org/10.1109/TPAMI.2018.2807452

[62] Li, Y., Sun, B., Wu, T., & Wang, Y. (2016). Face detection with end-to-end integration of a ConvNet and a 3D model. In Proceedings of the European Conference on Computer Vision (pp. 420–436). Springer.

[63] Zhu, C., Zheng, Y., Luu, K., & Savvides, M. (2017). CMS-RCNN: Contextual multi-scale region-based CNN for unconstrained face detection. In Deep Learning for Biometrics (pp. 57–79). Springer. https://doi.org/10.48550/arXiv.1606.05413

[64] Wan, S., Chen, Z., Zhang, T., Zhang, B., & Wong, K. (2016). Bootstrapping face detection with hard negative examples. arXiv preprint arXiv:1608.02236.

[65] Opitz, M., Waltner, G., Poier, G., Possegger, H., & Bischof, H. (2016). Grid loss: Detecting occluded faces. In Proceedings of the European Conference on Computer Vision (Vol. 9907, pp. 386–402). Springer.

[66] Wang, Y., Ji, X., Zhou, Z., Wang, H., & Li, Z. (2017). Detecting faces using region-based fully convolutional networks. arXiv preprint arXiv:1709.05256.

[67] Zhu, C., Tao, R., Luu, K., & Savvides, M. (2018). Seeing small faces from robust anchor's perspective. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5127–5136). IEEE.

[68] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10), 1499–1503. https://doi.org/10.1109/LSP.2016.2603342

[69] Farfade, S. S., Saberian, M. J., & Li, L. J. (2015). Multi-view face detection using deep convolutional neural networks. In Proceedings of the 5th ACM on International Conference on Multimedia Retrieval (pp. 643–650). ACM.

[70] Huang, L., Yang, Y., Deng, Y., & Yu, Y. (2015). DenseBox: Unifying landmark localization with end-to-end object detection. arXiv preprint arXiv:1509.04874.

[71] Yu, J., Jiang, Y., Wang, Z., Cao, Z., & Huang, T. (2016). UnitBox: An advanced object detection network. In Proceedings of the 24th ACM International Conference on Multimedia (pp. 516–520). ACM.

[72] Hu, P., & Ramanan, D. (2017). Finding tiny faces. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 1522–1530). IEEE.

[73] Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., & Li, S. Z. (2017). S3FD: Single shot scale-invariant face detector. In Proceedings of the IEEE International Conference on Computer Vision (pp. 192–201). IEEE.

[74] Wang, J., Yuan, Y., & Yu, G. (2017). Face attention network: An effective face detector for the occluded faces. arXiv preprint arXiv:1711.07246.

[75] Liu, Y., Li, H., Yan, J., Wei, F., Wang, X., & Tang, X. (2017). Recurrent scale approximation for object detection in CNN. In Proceedings of the IEEE International Conference on Computer Vision (pp. 571–579). IEEE. https://doi.org/10.1109/ICCV.2017.69

[76] Song, G., Liu, Y., Jiang, M., Wang, Y., Yan, J., & Leng, B. (2018). Beyond trade-off: Accelerate FCN-based face detector with higher accuracy. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7756–7764). IEEE.

BIBLIOGRAPHIE

[77] Zhang, S., Chi, C., Lei, Z., & Li, S. Z. (2020). RefineFace: Refinement neural network for high performance face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. https://doi.org/10.1109/TPAMI.2020.2978428

[78] Li, Z., Tang, X., Han, J., Liu, J., & He, R. (2019). PyramidBox++: High performance detector for finding tiny face. arXiv preprint arXiv:1904.00386.

[79] Xu, Y., Yan, W., Sun, H., Yang, G., & Luo, J. (2019). CenterFace: Joint face detection and alignment using face as point. arXiv preprint arXiv:1911.03599.

[80] Zhao, J., Cheng, Y., Cheng, Y. P., Yang, Y., Lan, H., Zhao, F., Xiong, L., Xu, Y., Li, J., Pranata, S., Shen, S., Xing, J., Liu, H., Yan, S., & Feng, J. (2019). Look across elapse: Disentangled representation learning and photorealistic cross-age face synthesis for age-invariant face recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 9251–9258. https://doi.org/10.1609/aaai.v33i01.33019251

[81] Zhang, S., Zhu, R., Wang, X., Shi, H., Fu, F., Wang, S., Mei, T., & Li, S. Z. (2019). Improved selective refinement network for face detection. arXiv preprint arXiv:1901.06651.

[82] Zhang, B., Li, J., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Xia, Y., Pei, W., & Ji, R. (2020). ASFD: Automatic and scalable face detector. arXiv preprint arXiv:2003.11228.

[83] Liu, Y., Tang, X., Han, J., Liu, J., Rui, D., & Wu, X. (2020). HAMBox: Delving into mining high-quality anchors on face detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 13043–13051). https://doi.org/10.1109/CVPR42600.2020.01306

[84] Zhuang, C., Zhang, S., Zhu, X., Lei, Z., Wang, J., & Li, S. Z. (2019). FLDet: A CPU real-time joint face and landmark detector. In Proceedings of the International Conference on Biometrics (pp. 1–8). https://doi.org/10.1109/ICB45273.2019.8987289

[85] Ge, S., Li, J., Ye, Q., & Luo, Z. (2017). Detecting masked faces in the wild with LLE-CNNs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2682–2690). https://doi.org/xxxx

[86] Ming, X., Wei, F., Zhang, T., Chen, D., & Wen, F. (2019). Group sampling for scale-invariant face detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3446–3456). https://doi.org/10.1109/CVPR.2019.00356

[87] Zhang, F., Fan, X., Ai, G., Song, J., Qin, Y., & Wu, J. (2019). Accurate face detection for high performance. arXiv preprint arXiv:1905.01585.

[88] Liu, Y., Tang, X., Han, J., Liu, J., Rui, D., & Wu, X. (2020). HAMBox: Delving into mining high-quality anchors on face detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 13043–13051). https://doi.org/xxxx

[89] Law, H., & Deng, J. (2018). CornerNet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (pp. 734–750). https://doi.org/10.48550/arXiv.1808.01244

[90] Tian, Z., Shen, C., Chen, H., & He, T. (2019). FCOS: Fully convolutional one-stage object detection. In Proceedings of the IEEE International Conference on Computer Vision (pp. 9626–9635). https://doi.org/10.1109/ICCV.2019.00972

[91] Nowrin, A., Afroz, S., Rahman, M. S., & Mahmud, M. (2021). Comprehensive review on facemask detection techniques in the context of COVID-19. IEEE Access, 9, 106839–106864. https://doi.org/10.1109/ACCESS.2021.3100072

[92] Ge, S., Li, J., Ye, Q., & Luo, Z. (2017). Detecting masked faces in the wild with LLE-CNNs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2682–2690). https://doi.org/10.1109/CVPR.2017.53

BIBLIOGRAPHIE

[93] Inamdar, M., & Mehendale, N. (2020). Real-time face mask identification using FaceMaskNet deep learning network. SSRN. https://doi.org/10.2139/ssrn.3663305

[94] Khandelwal, P., Choudhury, P., & Singh, A. (2020). Using computer vision to enhance safety of workforce in manufacturing in a post-COVID world. arXiv preprint arXiv:2005.05287. http://arxiv.org/abs/2005.05287

[95] Agarwal, C., Kaur, I., & Yadav, S. (2023). Hybrid CNN-SVM model for face mask detector to protect from COVID-19. In Artificial Intelligence on Medical Data (pp. 419–426). Springer. https://doi.org/10.1007/978-981-19-0151-5_35

[96] Ongsulee, P. (2017). Artificial intelligence, machine learning, and deep learning. In 2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE) (pp. 1–6). IEEE. https://doi.org/10.1109/ICTKE.2017.8259629

[97] Deng, L., Yu, D., & Platt, J. (2014). Deep learning: Methods and applications. Foundations and Trends in Signal Processing, 7(3–4), 197–387. https://doi.org/10.1561/2000000039

[98] Bengio, Y., Courville, A., & Vincent, P. (2009). Learning deep architectures for AI. Foundations and Trends in Machine Learning, 2(1), 1–127. https://doi.org/10.1561/2200000006

[99] Xiong, H. Y., Alipanahi, B., Lee, L. J., Bretschneider, H., Merico, D., Yuen, R. K., ... & Hughes, T. R. (2015). The human splicing code reveals new insights into the genetic determinants of disease. Science, 347(6218), 1254806. https://doi.org/10.1126/science.1254806

[100] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484–489. https://doi.org/10.1038/nature16961

[101] Buggenthin, F., Buettner, F., Hoppe, P. S., Endele, M., Kroiss, M., Strasser, M., ... & Hilsenbeck, O. (2017). Prospective identification of hematopoietic lineage choice by deep learning. Nature Methods, 14(4), 403–406. https://doi.org/10.1038/nmeth.4182

[102] Gibney, E. (2017). Google reveals secret test of AI bot to beat top Go players. Nature, 541(7636), 142. https://doi.org/10.1038/541142a

[103] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., ... & Zhang, J. (2016). End-to-end learning for self-driving cars. arXiv preprint arXiv:1604.07316.

[104] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115–118. https://doi.org/10.1038/nature21056

[105] Hazlett, H. C. (2013). Early brain development in infants at high risk for autism spectrum disorder. Biological Psychiatry, 73(9), 115S. https://doi.org/10.1038/nature21369

[106] Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576.

[107] Dechter, R., & Pearl, J. (1986). The cycle-cutset method for improving search performance in AI applications. University of California, Computer Science Department.

[108] Barjouei, H. S., Ghorbani, H., Mohamadian, N., Davoodi, S., Wood, D. A., & Alvar, M. A. (2021). Prediction performance advantages of deep machine learning algorithms for two-phase flow rates through wellhead chokes. Journal of Petroleum Exploration and Production, 11(3), 1233–1261. https://doi.org/10.1007/s13202-020-01075-0

BIBLIOGRAPHIE

[109] Aizenberg, I., Aizenberg, N. N., & Vandewalle, J. P. (2013). Multi-valued and universal binary neurons: Theory, learning and applications. Springer Science & Business Media.

[110] Van den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. In Advances in Neural Information Processing Systems (pp. 2643–2651).

[111] Mohamed, I. S. (2017). Detection and tracking of pallets using a laser rangefinder and machine learning techniques [Master's thesis, European Master on Advanced Robotics (EMARO+), University of Genova, Italy].

[112] Villaseñor, C., Armenta, A., Flores, A., & Lozano, R. (2020). Environment classification for unmanned aerial vehicle using convolutional neural networks. Applied Sciences, 10(14), 4991. https://doi.org/10.3390/app10144991

[113] Rocco, I., Arandjelovic, R., & Sivic, J. (2017). Convolutional neural network architecture for geometric matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6148–6157).

[114] Baheti, P. (2022). Activation functions in neural networks (12 types and use cases). V7 Labs. Retrieved June 10, 2022, from https://www.v7labs.com/blog/neural-networks-activation-functions

[115] Akbari, A., Awais, M., Bashar, M., & Kittler, J. (2021). How does loss function affect generalization performance of deep learning? Application to human age estimation. In International Conference on Machine Learning (pp. 141–150). PMLR.

[116] Ren, J., Zhang, Y., Li, X., & Liu, Y. (2022). Balanced MSE for imbalanced visual regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7926–7935). https://doi.org/10.48550/arXiv.2203.16427

[117] Hyeon, E., Kim, S., & Park, S. (2022). Loss function design for data-driven predictors to enhance the energy efficiency of connected and automated vehicles. IEEE Transactions on Intelligent Transportation Systems, 23(8), 12345–12356. https://doi.org/10.1109/TITS.2022.

[118] Wang, Q., Li, X., Zhang, Y., & Liu, Y. (2022). A comprehensive survey of loss functions in machine learning. Annals of Data Science, 9(2), 187–212. https://doi.org/10.1007/s40745-022-00365-2

[119] Wikidocs. (n.d.). Introduction to YOLOv3. Retrieved from https://wikidocs.net/167695

[120] Katsamenis, I., Protopapadakis, E., Doulamis, A., & Voulodimos, A. (2023). TraCon: A novel dataset for real-time traffic cones detection using deep learning. In Novel & Intelligent Digital Systems Conferences (pp. 123–134). Springer. https://doi.org/10.1007/978-3-031-xxxxxx

[121] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (pp. 5998–6008). https://doi.org/10.48550/arXiv.1706.03762

[122] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[123] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. arXiv preprint arXiv:1801.06146.

[124] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140), 1–67. http://jmlr.org/papers/v21/20-074.html

[125] Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2021). Transformers in vision: A survey. arXiv preprint arXiv:2101.01169.

BIBLIOGRAPHIE

[126] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030.

[127] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv preprint arXiv:1602.07261.

[128] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770–778). https://doi.org/10.1109/CVPR.2016.90

[129] Vapnik, V. N., & Kotz, S. (1982). Estimation of dependences based on empirical data (Vol. 40). Springer-Verlag.

[130] Loey, M., Manogaran, G., Taha, M., & Khalifa, N. (2020). 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, 102600. https://doi.org/10.1016/j.scs.2020.102600

[131] Zou, Z., Shi, Z., Guo, Y., & Ye, J. (2019). Object detection in 20 years: A survey. arXiv preprint arXiv:1905.05055.

[132] Zhao, Z.-Q., Zheng, P., Xu, S.-T., & Wu, X. (2019). Object detection with deep learning: A review. IEEE Transactions on Neural Networks and Learning Systems, 30(11), 3212–3232. https://doi.org/10.1109/TNNLS.2018.2876865

[133] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) challenge. International Journal of Computer Vision, 88(2), 303–338. https://doi.org/10.1007/s11263-009-0275-4

[134] Markuš, N., Frljak, M., Pandžić, I. S., Ahlberg, J., & Forchheimer, R. (2013). A method for object detection based on pixel intensity comparisons. Pattern Recognition, 46(12), 3340–3350. https://doi.org/10.1016/j.patcog.2013.05.008

[135] Belahcene, M. (2015). 2D and 3D face recognition based on IPC detection and patch of interest regions. In 2014 International Conference on Connected Vehicles and Expo (ICCVE) (pp. 1–6). IEEE. https://doi.org/10.1109/ICCVE.2014.7297515

[136] Ghiasi, G., & Fowlkes, C. (2014). Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1899–1906). https://doi.org/10.1109/CVPR.2014.245

[137] Yang, B., Yan, J., Lei, Z., & Li, S. Z. (2015). Convolutional channel features. In Proceedings of the IEEE International Conference on Computer Vision (pp. 82–90). https://doi.org/10.1109/ICCV.2015.17

[138] Dollár, P., Appel, R., Belongie, S., & Perona, P. (2014). Fast feature pyramids for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(8), 1532–1545. https://doi.org/10.1109/TPAMI.2014.2300479

[139] Zhu, C., Zheng, Y., Luu, K., & Savvides, M. (2017). CMS-RCNN: Contextual multi-scale region-based CNN for unconstrained face detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 57–79). https://doi.org/10.1109/CVPR.2017.12

[140] Li, X., Yang, Z., & Wu, H. (2020). Face detection based on receptive field enhanced multi-task cascaded convolutional neural networks. IEEE Access, 8, 174922–174930. https://doi.org/10.1109/ACCESS.2020.3025999

[141] Qin, B., & Li, D. (2020). Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19. arXiv preprint arXiv:2004.xxxxx.

BIBLIOGRAPHIE

[142] Ejaz, M. S., Islam, M. R., Sifatullah, M., & Sarker, A. (2019). Implementation of principal component analysis on masked and non-masked face recognition. In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT) (pp. 1–5). https://doi.org/10.1109/ICASERT.2019.8934567

[143] Belahcene, M. (2013). Biometric identification and authentication [Doctoral dissertation, Mohamed Khider University, Biskra].

[144] Chowdary, G. J., & Punn, N. S. (2020). Face mask detection using transfer learning of InceptionV3. arXiv preprint arXiv:2009.08369.

[145] Kaggle. (2020). Face mask detection. Retrieved from https://www.kaggle.com/andrewmvd/face-mask-detection

[146] Wang, Z., Wang, G., Huang, B., Xiong, Z., Hong, Q., Wu, H., ... & Che, H. (2020). Masked face recognition dataset and application. arXiv preprint arXiv:2003.09093.

[147] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9), 1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824

[148] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 580–587). https://doi.org/10.1109/CVPR.2014.81

[149] Misra, D. (2019). Mish: A self-regularized non-monotonic activation function. arXiv preprint arXiv:1908.08681.

[150] Nagrath, P., Jain, R., Madan, A., Arora, R., Kataria, P., & Hemanth, J. (2021). SSDMNV2: A real-time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. Sustainable Cities and Society, 66, 102692. https://doi.org/10.1016/j.scs.2020.102692

[151] Kaggle. (2020). FMD (Face Mask Detection). Retrieved from https://www.kaggle.com/andrewmvd/face-mask-detection

[152] Kaggle. (2020). SMFD (Synthetic Masked Face Dataset). Retrieved from [URL] (Accessed May 25, 2020).

[153] Chiang, D. (2020). Detect faces and determine whether people are wearing masks. Kaggle. Retrieved from [URL]

[154] Ciaparrone, G., Sánchez, F. L., Tabik, S., Troiano, L., Tagliaferri, R., & Herrera, F. (2020). Deep learning in video multi-object tracking: A survey. Neurocomputing, 381, 61–88. https://doi.org/10.1016/j.neucom.2019.11.023

[155] Sadeghian A, Alahi A, Savarese S (2017) Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In Proceedings of the IEEE International Conference on Computer Vision, pp. 300-311. https://doi.org/10.1109/ICCV.2017.41

[156] Chu P, Fan H, Tan CC, Ling H (2019) Online multi-object tracking with instance-aware tracker and dynamic model refreshment. In 2019 IEEE winter conference on applications of computer vision (WACV) (pp. 161-170). IEEE. https://doi.org/10.48550/arXiv.1902.08231

[157] Bewley A, Ge Z, et all (2016) Simple online and realtime tracking. In 2016 IEEE international conference on image processing, ICIP, pp. 3464-3468. IEEE. https://doi.org/10.1109/ICIP.2016.7533003

[158] Sanchez-Matilla R, Poiesi F, Cavallaro A (2016) online multi-target tracking with strong and weak detections. In European Conference on Computer Vision, pp. 84-99. Springer, Cham. https://doi.org/10.1007/978-3-319-48881-3_7

[159] Wojke N, Bewley A, Paulus D (2017) Simple online and realtime tracking with a deep association metric. In 2017 IEEE international conference on image processing, ICIP, pp. 3645-3649. IEEE. https://doi.org/10.48550/arXiv.1703.07402

BIBLIOGRAPHIE

[160] Milan A, Leal-Taixé L, et all (2016) MOT16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831.