FACE MASK DETECTION USING DEEP LEARNING METHODS

YOUNUS ALQADIRI

JUNE, 2022

FACE MASK DETECTION USING DEEP LEARNING METHODS

A THESIS SUBMITTED TO THE

GRADUATE SCHOOL

OF

BAHÇEŞEHİR UNIVERSITY

YOUNUS ALQADIRI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR

THE DEGREE OF MASTER OF COMPUTER ENGINEERING

IN THE DEPARTMENT OF NATURAL AND APPLIED SCIENCES

JUNE, 2022

T.C.

BAHCESEHIR UNIVERSITY

GRADUATE SCHOOL

...../...../.....

MASTER THESIS APPROVAL FORM

Program Name:	Computer Engineering
Student's Name and Surname:	Younus Alqadiri
Name of The Thesis:	FACE MASK DETECTION USING DEEP LEARNING METHODS
Thesis Defense Date:	24/06/2022

This thesis has been approved by the Graduate School, which has fulfilled the necessary conditions as a Master thesis.

Prof. Dr. Ahmet ÖNCÜ

Institute Director

This thesis was read by us, quality, and content as a Master's thesis has been seen and accepted as sufficient.

	Title/Name	**Signature**
Thesis Advisor's	Assist. Prof. Zafer İŞCAN	
Member's	Assoc. Prof. C. Okan ŞAKAR	
Member's	Prof. Dr. Zümray DOKUR	

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Younus ALQADIRI :

Signature :

ACKNOWLEDGEMENT

At the outset, I would like to extend my sincere thanks and appreciation to my supervisor, Assist. Prof. Zafer İşcan who provided me with his advice, guidance, endless assistance, and valuable comments that raised my work to a high level.

I would also like to thank my wife Rawia, who was patient and took care of me and my children, tolerated my absence, and encouraged me throughout my study period. I also extend my thanks to my dear parents, whose prayers and support for me have not ceased since childhood to this day, and without whom I would not have reached this success. I would also like to thank the academic staff and members of the faculty of engineering and natural sciences at Bahçeşehir University for their cooperation and kindness to me. Finally, I thank all my friends and colleagues who gave me encouragement and assistance, directly or indirectly.

Istanbul, 2022 Younus ALQADIRI

ABSTRACT

FACE MASK DETECTION USING DEEP LEARNING METHODS

Younus ALQADIRI

Computer Engineering Master Program

Thesis Supervisor: Assist. Prof. Zafer İşcan

June 2022, 45 Pages

The Covid-19 outbreak, which began in Wuhan, China, has become a worldwide public health problem. The virus spreads mostly through inhaling respiratory aerosols caused by sneezing, coughing, breathing, or talking, and it can cause health complications and even death. The World Health Organization suggested that people wear face masks for personal safety and as a public health strategy to control and prevent the expansion of infection. In this study, EfficientNet-B7 model, which is based on deep learning and in particular transfer learning from the OpenCV library was applied to face mask detection problem. The selected model limits the spread of COVID-19 or any upcoming disease by detecting the face in the image or real-time video and determining whether the people are wearing any facial masks or not in public places, health facilities, transportation, factories, etc. This model has been compared to other models using the MAFA dataset for masked faces and LFW dataset for unmasked faces. The selected model achieved 98.87% average accuracy in recognizing people with and without a face mask while other models got an accuracy of 95.91% for the MobileNetV2 model, 86.83% for the VGG19 model, and 68.31% for the ResNet50 model. Although, the threat caused by the Covid-19 may fade in the near future, especially as a result of virus mutations, new pandemics might arise. Thus face mask detection will always be an issue.

Keywords: Covid-19, Face Mask Detection, Deep learning, Transfer Learning, OpenCV

ÖZ

DERİN ÖĞRENME YÖNTEMLERİYLE YÜZ MASKESİ TESPİTİ

Younus ALQADIRI

Bilgisayar Mühendisliği Yüksek Lisans Programı

Tez Danışmanı: Dr. Öğr. Üyesi. Zafer İşcan

Haziran 2022, 45 Sayfa

Çin'in Vuhan kentinde başlayan Covid-19 salgını, dünya çapında bir halk sağlığı sorunu haline gelmiştir. Virüs çoğunlukla hapşırma, öksürme, nefes alma veya konuşmanın neden olduğu solunum aerosollerinin solunması yoluyla yayılmakta, sağlık sorunlarına ve hatta ölüme neden olmaktadır. Dünya Sağlık Örgütü, insanların kişisel güvenliğini sağlamak, enfeksiyonun yayılmasını kontrol etmek ve önlemek için bir halk sağlığı stratejisi olarak yüz maskeleri takılmasını önermektedir. Bu çalışmada, OpenCV kütüphanesinde yer alan, derin öğrenme ve özellikle transfer öğrenmeyi temel alan EfficientNet-B7 modeli yüz maskesi algılama problemine uygulanmıştır. Seçilen model, görüntüdeki veya gerçek zamanlı videodaki yüzü tespit ederek ve insanların halka açık yerlerde, sağlık tesislerinde, ulaşımda, fabrikalarda vb. herhangi bir yüz maskesi takıp takmadığını belirleyerek COVID-19'un veya yaklaşan herhangi bir hastalığın yayılmasını sınırlamaktadır. Bu model, maskeli yüzler için MAFA veri setini ve maskesiz yüzler için LFW veri setini kullanarak diğer modellerle karşılaştırılmıştır. Seçilen model yüz maskesi olan ve olmayan kişileri tanımada %98.87 ortalama doğruluk elde ederken, diğer modeller MobileNetV2 modeli için %95.91, VGG19 modeli için %86.83 ve ResNet50 modeli için %68.31 doğruluk elde etmiştir. Covid-19'un neden olduğu tehdit yakın gelecekte, özellikle virüs mutasyonlarının bir sonucu olarak ortadan kalkabilecek olsa da, yeni pandemiler ortaya çıkabilir. Bu nedenle yüz maskesi tespiti her zaman bir sorun olacaktır.

Anahtar Kelimeler: Covid-19, Yüz Maskesi Algılama, Derin öğrenme, Transfer Öğrenme, OpenCV

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
ASMs	Active Shape Models
CNN	Convolutional Neural Network
CV	Computer Vision
DL	Deep Learning
DNN	Deep Neural Network
FC	Fully Connected
FMLD	Face-Mask-Label Dataset
HOG	Histogram of Oriented Gradients
IoU	Intersection over Union
LFW	Labeled Faces in the Wild
LLE-CNNs	Locally Linear Embedding-Convolutional Neural Networks
MAFA	Masked Faces
ML	Machine Learning
PCA	Principle Component Analysis
ReLU	Rectified Linear Unit
ROI	Region of Interest
SSD	Single Shot Multibox Detector
TL	Transfer Learning
TLU	Threshold Logic Unit
WHO	World Health Organization

Chapter 1: Introduction

1.1 Background

At the end of 2019, a new viral disease broke out called Covid-19, and the first cases were discovered in Wuhan, China, and it spreads all around the world. The infection is transmitted through flying droplets from one person to another in cases of talking, coughing, sneezing, or by touching one of the surfaces contaminated with the virus and finally transmitting it to the respiratory system. People were affected by the quarantine measures, as well as the restriction on travel and movement between countries and cities that produce successive economic crises globally since many businesses cannot be performed remotely and require the presence of people together in a closed place. Therefore, the World Health Organization (WHO), as well as most governments of countries around the world and experts in the medical field, instructed wearing face masks compulsorily for all people. They include those who have respiratory health problems or who care for infected people or medical workers, as the mask reduces about 80% of infections (Liang et al., 2020). Some countries have produced and developed effective vaccines with rates that may reach 90% to prevent the spreading of infection, but the infected people with the disease still exist, as several new mutations of the virus have appeared that made the vaccine insufficient and requires continuing to adhere to protective measures, including face masks. Although, the risk posed by the Covid-19 may fade in the near future, especially as a result of virus mutations, the chance of new pandemics arising remains a possibility, thus face mask detection will always be an issue (Batagelj, Peer, Štruc, & Dobrišek, 2021a; Higuchi, Taniguchi, Kawasaki, & Sonoda, 2021; Loey, Manogaran, Taha, & Khalifa, 2021a; Rahman, Manik, Islam, Mahmud, & Kim, 2020).

In order to follow up on people's compliance to preventive measures, including the necessity of wearing the face masks correctly, many systems for detecting face masks have appeared, and their techniques have been developed to become efficient in good proportions to help determine whether the person wears a face mask or not.

1.2 Face Detection

Recent years have seen an increase in interest in face recognition as one of the best promising apps for image processing. Facial detection may make up a significant portion of face recognition procedures. Its power is to concentrate computational resources on the part of an image that contains a face. Because of the variety contained in human faces, such as behavior, facial expression, position and orientation, color of the skin, the presence of glasses or beard and mustache, contrast in-camera gain, lighting situations, and image quality, the process of face recognition in photos is challenging.

Face detection is the primary and initial step in face recognition. The objective of face detection algorithms is to determine whether or not an image contains a face. It has applications in many different domains, including law enforcement, entertainment, personal safety, biometrics, security, and other similar areas (Dwivedi, 2018).

1.2.1 Face detection methods. Face detection entails splitting an image into two pieces, one of which contains the face and the other contains the background. It's complicated because, while there are some similarities across faces, they might differ significantly in terms of aging, color of skin, as well as facial expression (Sharma, 2014). Face detection techniques are divided into two groups: feature-based and image-based techniques (Hjelmås & Low, 2001).

1.2.1.1 Feature-based techniques: Feature-based methods recognize people based on their facial features. This technique is divided into three categories: low-level analysis, feature analysis, and active shape model.

1. Low-level analysis: This type of analysis works with segmenting visual features utilizing pixels properties, grayscale level, and motion information. For verification, edge-based algorithms rely on labeled edges that are matched to the face model. Extraction techniques might check for the local minima in places such as the brows, eyes, and lips because these areas are typically darker than the surrounding areas. Local maxima, on the other hand, can be utilized to identify bright face features like nose tips (Brimblecombe, 2002). The detection is then implemented utilizing grayscale thresholding at a low-level.

2. Feature analysis: Using an extra information for the face then eliminates any uncertainty that low level analysis generates. The first comprises consecutive feature seeking algorithms on the basis of each facial feature related location (Brimblecombe, 2002). After determining the most prominent face features, less prominent features might be postulated.

3. Active shape models: They describe the actual physical and the higher-level appearance of features. When they released in close proximity to a feature, the active shape models (ASMs) interact with local image features, and they distort progressively to take the shape of the feature (Hjelmås & Low, 2001). ASMs are object shape models that morph repeatedly fitting an instance of the item in another new image. They function in two ways: Look around every point inside the image looking for a better place, then change the parameters of the model to finest fit with these newly found positions.

1.2.1.2 Image-based techniques: Due to the unpredictability of faces and environmental variables, face detection using explicit modeling is a relatively simple approach. As a result, more robust techniques, proven ability to work in hostile circumstances, such as detecting many faces against cluttered backgrounds, are required. Face detection using images has sparked a new study field, and as a result, face detection is regarded as a generic pattern recognition issue. Various methodologies, such as neural networks, example-based learning, and support vector machines, are used in the image-based approach (Brimblecombe, 2002; Sharma, 2014).

1.3 Aims of this Thesis

1. Preparing and training a model for face mask detection with high accuracy.
2. To compare different face detection techniques using deep learning.

Chapter 2: Literature Review

2.1 Theoretical Background

2.1.1 Machine learning. In the beginnings of machine learning (ML), researchers processed input data using crafted features (A. C. Müller & Guido, 2016). For instance, the Viola-Jones facial detection technique uses edges as a simple illustration of a constructed feature that may be extracted from an image and used for face detection. (Viola & Jones, 2001). However, with the advancement of ML and the creation of methods like Convolutional Neural Networks (CNNs), these crafted features are no longer required, since these techniques can learn features from the images (O'Shea & Nash, 2015). ML is a field of study that investigates how computers can learn with minimal or without the need for human interaction (Géron, 2019; Watt, Borhani, & Katsaggelos, 2020).

2.1.2 Machine learning's challenges. In ML, there are two fundamental challenges that must be properly addressed. The first is the "model selection" that best matches the problem, the second element is the dataset that will be utilized to train the model. After a suitable model is selected, the next step is to train it. During the training period, there are several challenges.

2.1.2.1 Overfitting and underfitting. We want to build a model that can properly recognize the labels for unseen data during the training phase. Overfitting describes a model with a large variance that accurately catches the patterns within the training set and then fails to generalize on the unseen data. It could result from getting too numerous parameters, leading to a model that is too complicated with the data, Moreover, the cost function may be equal to or very close to zero. Complex models can detect even the tiniest patterns within data. Unlike the overfitting, the underfitting is a model with high-bias which isn't complicated enough to catch the patterns within the training set and makes relatively basic predictions about the data (Géron, 2019; A. C. Müller & Guido, 2016; Raschka, 2015).

Overfitting causes the model to try to match every single case perfectly, resulting in a large variance and an extremely "wiggly" curve. The model is relatively simplistic

in underfitting, and it tries to fit the training example with a straight line, which results in a model with a large bias (Julian, 2016; Raschka, 2015).

To summarize, "variance" refers to the consistency of a model's prediction for a specific instance if it were retrained on a different portion of the training dataset. When the model is retrained on a different dataset, "Bias" refers to the difference between the model predictions as well as the real values.

2.1.2.2 Quantity of data. To work properly, most ML algorithms need a huge amount of data. Even for extremely simple ML problems, thousands of samples will frequently be needed. Complex ML problems, such as object or voice recognition, need a higher number of training examples. Getting that much data isn't always easy (Géron, 2019).

The researchers (Banko & Brill, 2001) showed that the quantity of data can sometimes be more important than using a sophisticated algorithm. They provided sufficient training data for simple ML methods, which worked admirably on the hard challenge of language disambiguation. This concept was further validated in (Halevy, Norvig, & Pereira, 2009).

2.1.3 Computer vision techniques. Computer Vision (CV) is one of the subfields of ML. The artificial extraction of information from images is known as a CV. The information could be object detection, object recognition, and 3D models. It attempts to imitate human vision, in which humans can easily recognize objects by recognizing patterns of lighting and shading that play through the object's surface, whereas CV is susceptible to making mistakes and difficult to control. This is related to the fact that computer vision is an inverse issue, in which the computer attempts to detect or recognize unfamiliar things with limited information (Prince, 2013; Solem, 2012; Szeliski, 2010).

Object detection is a wide phrase that applies to a variety of CV-related tasks. It includes the identification and classification of objects. In general, CV is about "What are the objects and where are they located". Object detection is in charge of locating and classifying objects. It draws a bounding box around an object of a specified class (like

humans, animals, and cars, etc.) in a supplied image and labels it with object classification. For instance, human detection, face detection, and so on. Object recognition, on the other hand, takes a labeled object and determines its type. For instance, the name of a person whose face was detected, a car's model name for a car that was previously identified, and so on (Andreopoulos & Tsotsos, 2013; Treiber, 2010; Zou, Shi, Guo, & Ye, 2019).

2.1.3.1 Object detection. The process of detecting and labeling several items in a single image is known as object detection. The objective is to build a system capable of detecting and providing the locations and classes of objects, assuming we have a relevant object class and an image to analyze. Usually, a collection of boundary boxes with classification values will be returned. The object detection is made up of two parts: object localization, which is in charge of locating a particular object, and object classification, which is in charge of classifying that object. Object detection, which combines object localization and object classification, is responsible for recognizing and identifying multiple objects (Brownlee, 2019; Geiger, Lenz, & Urtasun, 2012).

Object detection can be done in a traditional approach or using a deep learning approach.

2.1.3.1.1 Traditional approach. The traditional method includes two parts: the feature extraction technique which encodes the image areas as a descriptors, and the classifier that labels image sections based on feature extraction outcomes (Felzenszwalb, Girshick, McAllester, & Ramanan, 2010; Tiwari, Kumar, & Saraswat, 2013).

To be able to classify an object, feature extraction develops feature vectors (descriptors). Local and global descriptors (feature vectors) are the two types of descriptors. Local descriptors may represent even extremely small portions in the image, whereas global descriptors try to describe the entire image and perform poorly when a change occurs in any part of the image, making them unsuitable for object detection problems. Objects can be classified using local descriptors based on color, edges, texture, or a set of these features (Pedregosa et al., 2011; Runia, 2015).

Feature extraction stands for minimizing the number of features in the dataset by creating new ones from the old ones (and then discarding the old ones). The most popular technique is Principle Component Analysis (PCA). To lower the dataset's dimension, it applies the dimensionality reduction algorithm, which is an unsupervised ML technique (Kuncheva & Faithfull, 2014).

In the features area, a question arises: “What are the numbers of features that are sufficient for the model to generalize a newly unseen data adequately?". An excessive amount of features causes overfitting, while an inadequate number of features leads to underfitting, as explained in section (2.1.2.1). To face this issue, a feature selection procedure is used to choose only the most significant features or to limit the impact of features entirely, allowing all features to participate in the detection task in some way (Pedregosa et al., 2011).

The feature extraction procedure has continually increased detection quality over the last years, according to researchers in (Benenson, Omran, Hosang, & Schiele, 2014). Building a more complex algorithm is less important than creating a better descriptor (feature extractor). This was demonstrated by comparing previous research in the field and discovering that, using a stronger feature representation of the dataset, the ML method did exceptionally well in determining the decision boundary that divides the object classes.

The classifier is the next aspect of the traditional technique. The procedures to train a classifier are depicted in Figure 1. Gathering training data is the initial step. A huge number of images of the target object are collected with a big number of non-target images inside the binary classification issue. The feature extraction procedure is then utilized to construct feature vectors (descriptors) from the available labels, which are subsequently put into the ML algorithm in order to create the model (Pedregosa et al., 2011).

The primary objective of the object detection challenge for whole images is to identify tiny objects that exist in the image. With each of these features, the object is classified using a process known as bag of words. If an object has a large number of features, it is classified into one of several classes based on the features found. Sliding

windows are utilized to scan the entire image and construct a label for every sub-window, which is then sent to the classifier. Because the sub-windows number can become out of hand, a step of four or eight pixels is used. The sliding windows problem is that the sub-window and the object do not always seem to be the same size. The image pyramid is one such solution, in which the image is rescaled to dissimilar sizes and the classifier is applied to each version. As described in Figure 2 (O'Mahony et al., 2020; Pedregosa et al., 2011), it will result in having several detections for the same object.

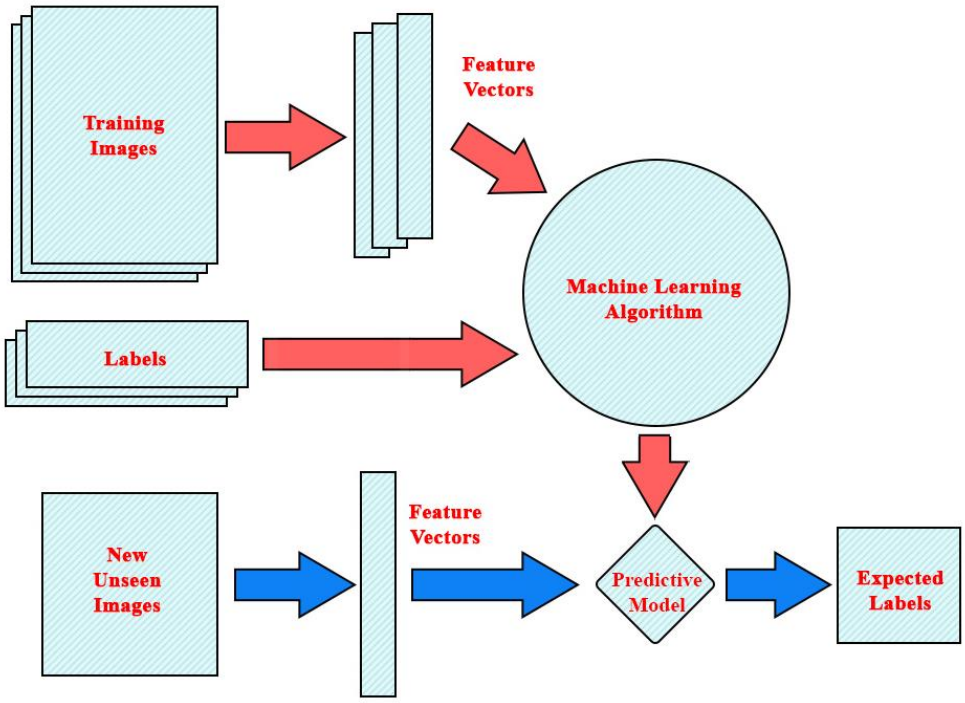

Figure 1 Process of Training and Classification

Non-maxima suppression techniques have been used to resolve this problem by discarding overlapping detections with lower confidence level and keeping just the detection with the highest level (Géron, 2019).

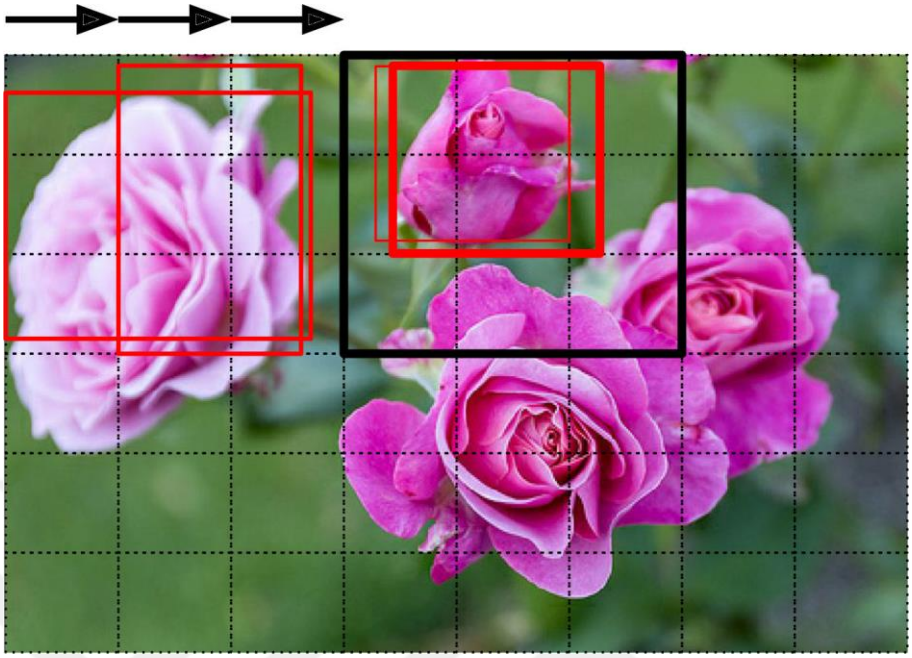

Figure 2 Several Detections for One Object (Géron, 2019)

Classifiers include, for example: Viola and Jones classifier (Viola & Jones, 2001) that utilizes Haar-like features. Histogram of Oriented Gradients (HOG) (Dalal & Triggs, 2005) utilizes the HOG feature. The deformable part-based model (Felzenszwalb, Girshick, & McAllester, 2010), with many more classifiers (Géron, 2019; Prince, 2013; Zou et al., 2019), builds a histogram of oriented gradients for every cell, then normalizes the output with the block-wise pattern, after that returning a descriptor for every cell.

2.1.3.1.2 Deep learning approach. Deep learning (DL) is an ML subfield. Artificial Neural Network (ANN) is a crucial component of DL. ANN attempts to emulate the function of neurons in the human brain. The threshold logic unit (TLU) is a basic neural network used for binary classification problems, as shown in Figure 3. A step function is used to compare the result to a threshold after a linear combination of the inputs is calculated. If the result exceeds the threshold, the output is positive; otherwise, it is negative. There is only one layer in TLU. The term "fully connected (FC) layer" refers to a layer that has all of its inputs linked to a previous layer. The layers between the input layers and the output layers are the "hidden layers", and the entire network is called as a Deep Neural Network (DNN). The DNN's initial and last layers are defined as "the input layers" and "the output layers" respectively (Géron, 2019; Raschka, 2015).

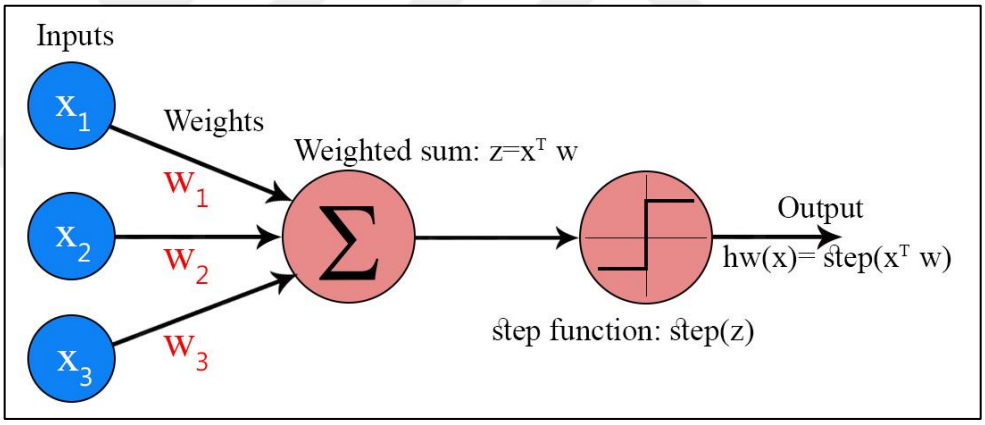

Figure 3 TLU Architecture (Géron, 2019)

Each input will be multiplied by a parameter known as "weight," which is represented by the letter w, as seen in Figure 3. If we need to build an FC neural network that resolves the image classification problem, the input image size will affect the number of the weights. If there is a colored image of one megapixel, there will be an image of 1000 × 1000 × 3, so the input will be three million dimensional. If each hidden layer has a thousand neurons, there are three billion parameters to train. It's challenging to gather enough data to keep the neural network from overfitting with so many parameters. Furthermore, the memory requirements for training three billion parameters

are impractical (Alzubaidi et al., 2021). Figure 4 depicts an FC layer network, in which all units are linked to earlier units (neurons).

The problem was solved by employing a Convolutional Neural Network (CNN). Regardless of the image size, the number of parameters is now decided. CNN is made up of numerous convolutional layers at the input layer, FC layers in the last few layers, and pooling layers in the middle. The following is a brief explanation of CNN and all of its connected topics:

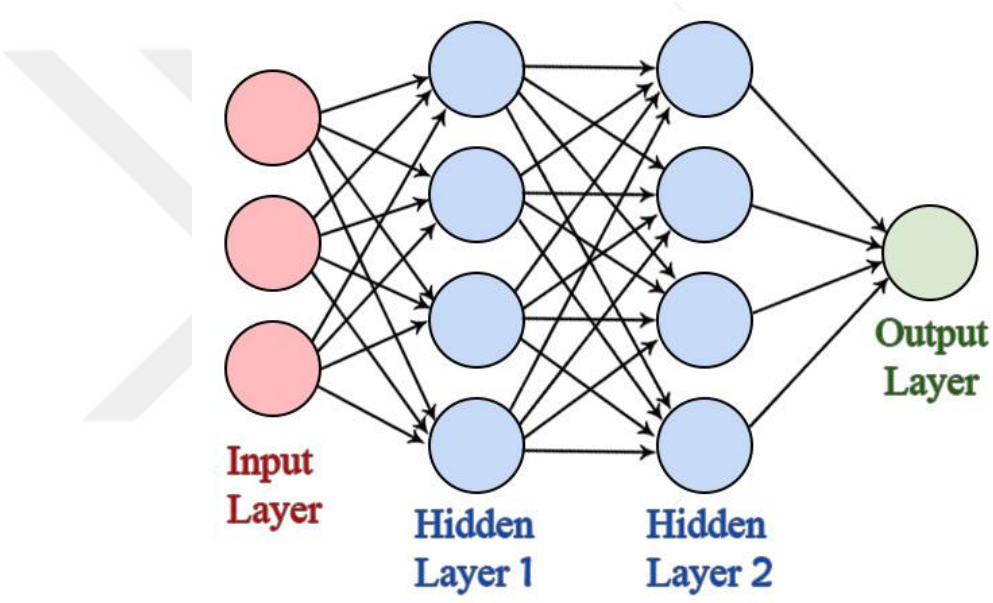

Figure 4 FC Layer Network

1- Convolutional layers: It is the most significant part of CNN architecture. It consists of several convolutional filters (kernels). The input image is convolved with these filters, which are provided as N-dimensional metrics. This will result in the creation of a feature map. Firstly the input image will be sent toward the first layer, it's where the F numbers of the filters found. The most significant distinction among the CNN and the traditional neural network lies in the fact that the filters' size is chosen independently of the size of the image. The size of the final matrix will be as (N-F+1 × N-F+1) if the input image was (N × N × 3) (when 3 signifies 3 channels of the RGB image) and F × F × 3 was the filter. For instance, if a 6 × 6 × 3 image is convolved with a 3 × 3 × 3 filter, the resulting matrix is 4 × 4. To build the feature map, a hidden layer

could have any number of filters, each of which extracts a distinct feature. In the previous example, one filter was 3 × 3 in size, resulting in 16 parameters. Regardless of the image size, the number of parameters in this example remains constant. In the same case, convolution was expected to be performed with a step of one (the number for the steps of sliding window moved in time), resulting in a smaller matrix dimension (dimension of the feature map). To avoid this problem, padding is chosen to perform before convolution to retain the dimensions of the feature. Another advantage of choosing the padding is that the further padding you apply, the extra corner pixels you'll have to give to the feature map (Alzubaidi et al., 2021; Bisong, 2019; Brownlee, 2019; Géron, 2019; A. Kumar, Upadhyay, & Kumar, 2020).

2- Pooling Layers: Its main purpose is to down sample the feature maps so this strategy will minimize the feature maps size. At the same time, in each level of the pooling process, it keeps the bulk of the principal information (features). Both of the stride and the kernel (filter) are given the size before pooling process, just like the convolutional operation. There are a number of pooling strategies at various pooling levels. The max pooling is the most popular one. Figure 5 depicts the operations of the pooling layer (Alzubaidi et al., 2021; Bisong, 2019; Krizhevsky, Sutskever, & Hinton, 2012; A. Kumar et al., 2020).

3- FC Layer: The FC layer is seen in Figure 4. It's also known as the dense layer because it is the final layer in the CNN architecture. It performs as a classifier. The FC layer receives input from the preceding convolutional layer. This data have been converted to a vector format. The softmax function is used in FC layer to provide the likelihood of a specific class element (Bisong, 2019).

4- Non-linear activation functions: All examples of the non-linear activation functions have the primary function of mapping the input to the output. One example of a non-linearity function is a sigmoid function. Its result will be between 0 and 1 and the input is a real value. Equation 2.1 shows its mathematical representation. The Rectified Linear Unit (ReLU) function, another example of non-linearity function, turns the input to a positive number. Because of its minimal computing requirements, it is commonly

utilized in CNN. It's written as an equation 2.2 (Alzubaidi et al., 2021; Bisong, 2019; Géron, 2019).

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (2.1)$$

$$f(x)_{ReLu} = max\,(0, x) \quad (2.2)$$

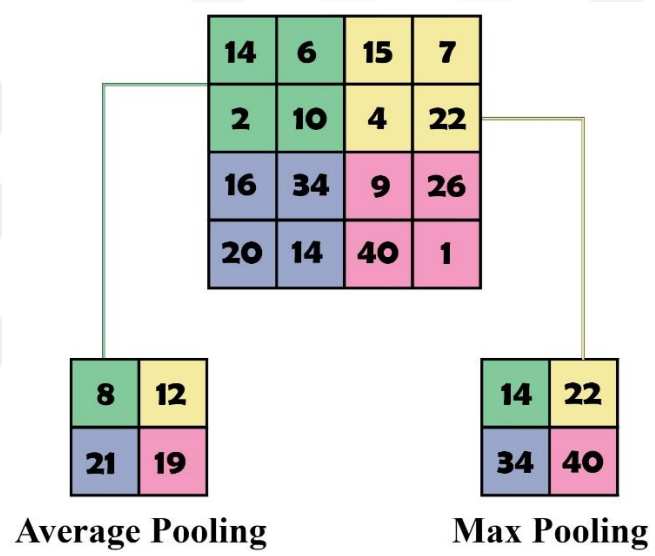

Figure 5 Pooling Operation Explained

5- CNN regularization: In ML, regularization is commonly used to avoid overfitting. There are numerous regularization approaches in CNN. The term dropout is the most commonly used. During each training cycle, neurons are lost at random. Consequently, the power of feature selection was distributed equally across the whole neurons group, forcing the model to be learned on numerous features independently. Data augmentation is a different method for overcoming overfitting. It uses a variety of data operations including scaling (zooming in and zooming out), rotation, flipping, and translation to create extra images (Alzubaidi et al., 2021; Bisong, 2019; Wang, Ma, Zhang, Gao, & Wu, 2018).

6- Backpropagation: This is a technique for minimizing the cost function. The cost function is the difference between expected output and the real output. Backpropagation attempts to reduce this difference by altering the values of the weights over the entire neural network till the error is as little as possible (Alzubaidi et al., 2021; Bisong, 2019; Wang et al., 2018). The backpropagation mechanism is represented in Figure 6.

7- Transfer Learning (TL): The disadvantage of DL is the necessity of huge amount of training data to obtain good performance. Data collecting is a hard task. TL came up with a solution to this problem. To train the CNN model for a specific task, the TL technique needs a big amount of data. The weights that have been trained can now be employed for other task. Instead of training weights from scratch, the TL technique allows using of pre-trained weights.

This method speeds up the training process because it eliminates the requirement for a large dataset to train the network. To fix a specific problem, we now employ a pre-trained model that we retrain it on our dataset.

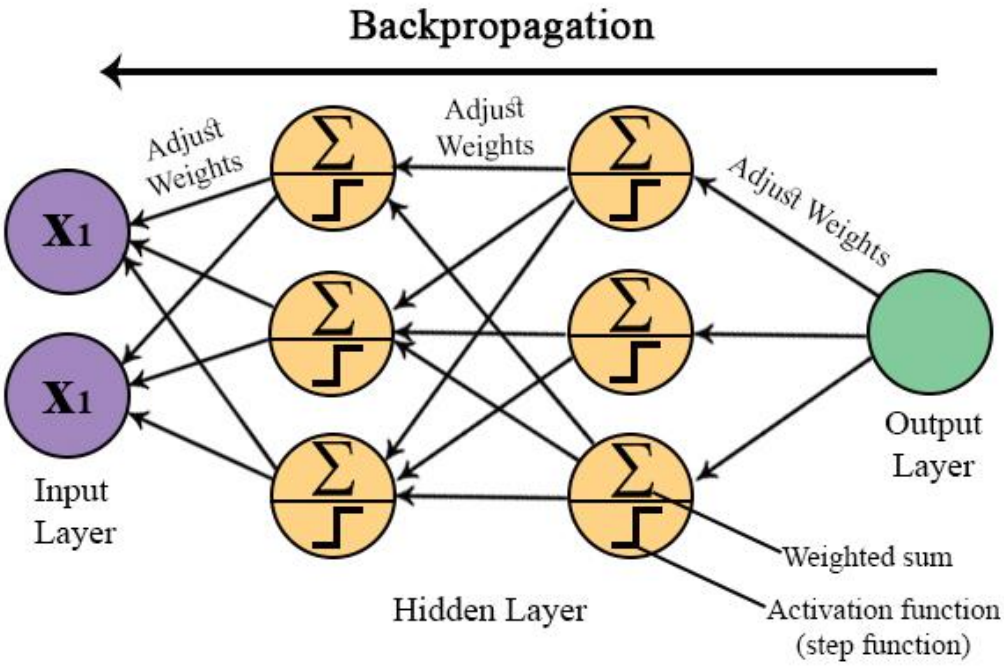

Figure 6 The Backpropagation Process

The complexity of this strategy as with the traditional approach is that we should first select the optimal set of features that depict the object we intend to classify. In the DL approach, end-to-end learning is utilized. The computer only provided a dataset of images that have previously been labeled and included a specific object. The neural network will then turn the data into an abstract representation to determine the best features that represent this object. The contrast between the standard strategy and the DL approach is shown in Figure 7 (Géron, 2019; O'Mahony et al., 2020; Wang et al., 2018).

DNNs have been demonstrated to outperform traditional algorithms, especially with the expansion of computing capabilities, which has shortened the lengthy training period required for DNNs, and the availability of the training data, which has facilitated the training process of the DNN (Géron, 2019; O'Mahony et al., 2020; Zou et al., 2019).

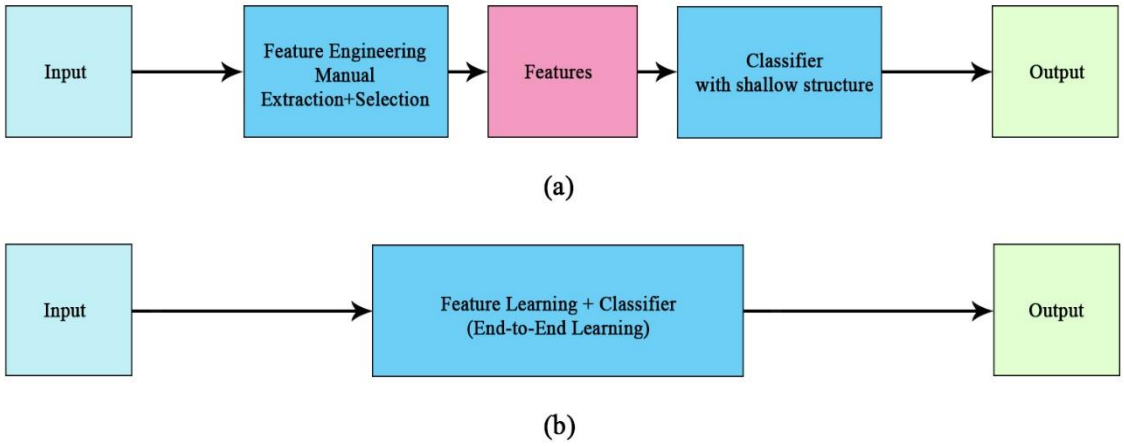

Figure 7 a) The Traditional Approach and b) The DL Approach

AlexNet (Krizhevsky et al., 2012), a CNN, was born into the world in 2012. It is one of the first deep CNNs to be utilized for feature extraction. AlexNet is made up of three FC layers and five convolutional layers. This CNN is capable of classifying thousand different objects. The output of the final FC layer is given to softmax function, which normalizes all inputs in the range of 0 to 1.

The first convolutional layer applies a filter of size 11 × 11 × 3 with stride = 4 on a 224 × 224 input image. The next convolutional layers create feature maps using additional learned filters, and later the max pooling is used to attain a reduced copy of the image. Edges and other low-level features will be learned in the layers nearest to the

input, however, going deeper into the layers it begins to learn exactly more complicated features such as shapes (Krizhevsky et al., 2012).

2.1.4 Selected model EfficientNet-B7. (Tan & Le, 2019) from Google Research introduced the EfficientNet model in their research paper "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks" between 2019 and 2020. These researchers investigated model scaling and discovered that properly balancing the network's depth, width, and resolution can improve the performance. EfficientNet scales up models using a simple but effective technique called compound scaling. Compound scaling equally scales each dimension with a particular set of scaling coefficients, rather than scaling up width, depth, or resolution randomly. The authors created seven models with multiple dimensions using scaling method, which outperformed earlier CNNs in terms of accuracy and efficiency.

They observed that, though scaling single dimensions improves model performance, balancing the scale in all three dimensions (width, depth, and image resolution) while taking into account the changeable available resources improves overall model performance the most. The term depth means how deep a network is , in other words, the extent of the number of the layers in a network. The term width means how the network is wide, as for width measurement is the number of convolution layer channel. Whereas the term of resolution means the resolution of the image that passed to the CNN. Compound scaling is shown in the Figure 8.

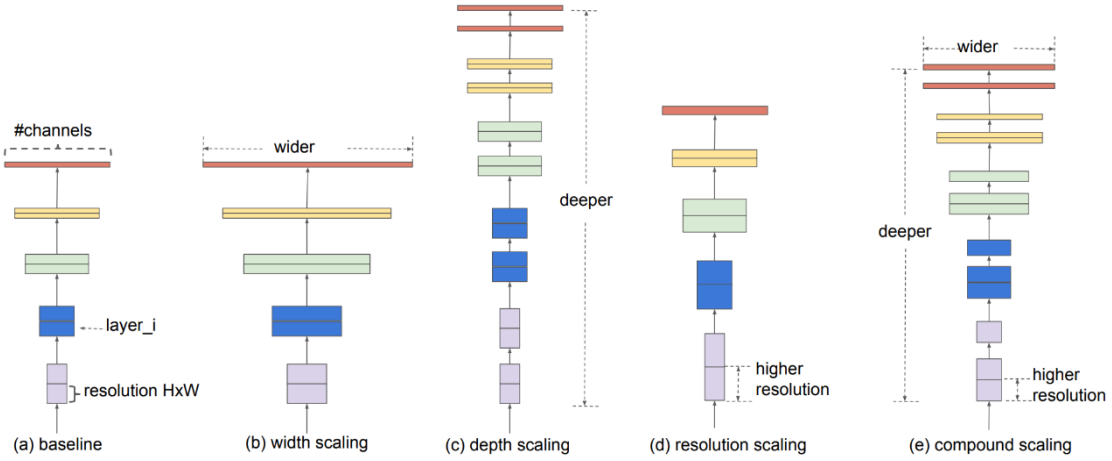

Figure 8 Compound Scaling vs. Different Scaling Methods (Tan & Le, 2019)

In comparison to other random scaling techniques, the compound scaling method improved the model efficiency and accuracy of prior CNN models such as MobileNet and ResNet by about 1.4% and 0.7% ImageNet accuracy, respectively (Tan & Le, 2019).

On the ImageNet and CIFAR-100 datasets, the largest EfficientNet model which is EfficientNet-B7, achieved state-of-the-art performance. On ImageNet, it scored about 84.4% top-1 and 97.3% top-5 accuracy. In addition, EfficientNet models were 8.4 times smaller and 6.1 times faster than the previous best CNN model. On the CIFAR-100 dataset, it achieved 91.7% accuracy, while on the Flowers dataset, it achieved 98.8% accuracy. The total number of parameters of the EfficientNet-B7 is 66 million parameters (Tan & Le, 2019).

The total number of layers in the EfficientNet-B0 is 237, while in EfficientNet-B7 is 813 but all of these layers are composed of the 5 modules shown in Figure 9. The EfficientNet-B7 start like the other EfficientNet models with stem and end with the final layers as shown in Figure 10. These modules are assembled together to form the sub-blocks as shown in Figure 11, which in turn will form the block parts of the EfficientNet-B7 architecture as shown in Figure 12 (Agarwal, 2020).

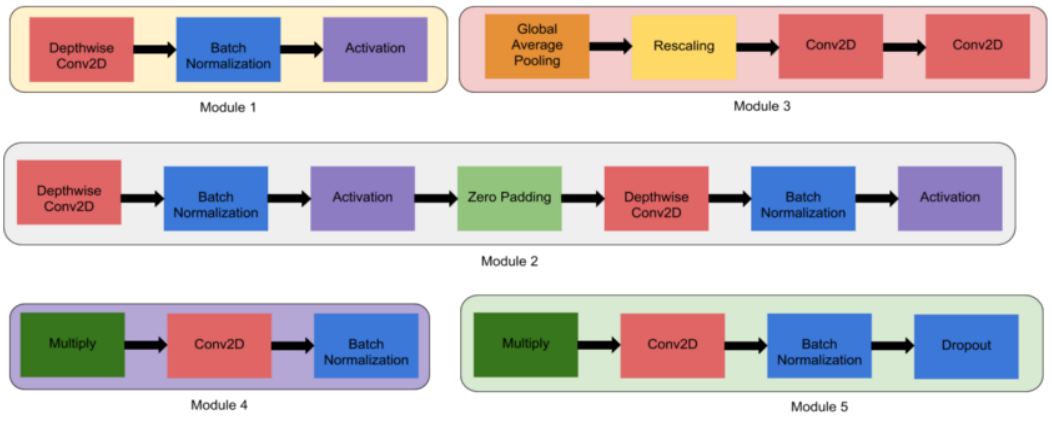

Figure 9 The 5 Modules That Used to Make the EfficientNet-B7 Architecture (Agarwal, 2020) Copyright Taken

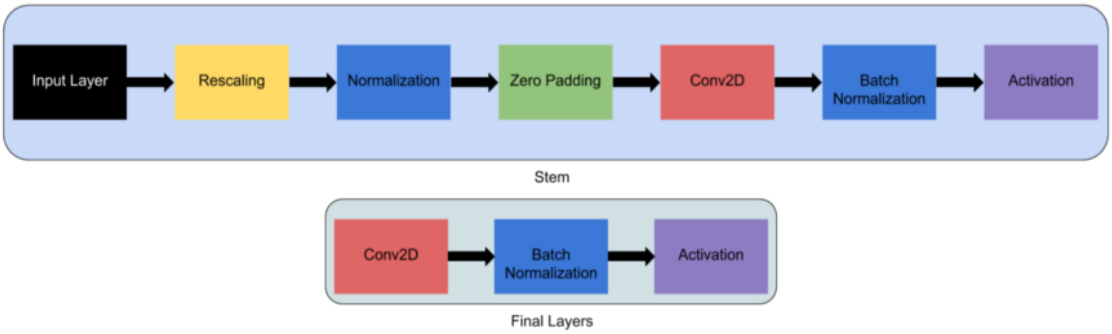

Figure 10 The Stem and Final Layers (Agarwal, 2020) Copyright Taken

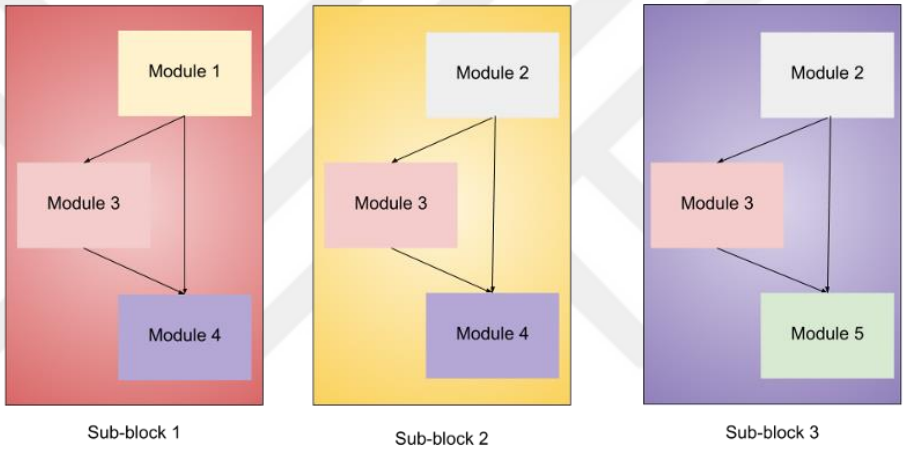

Figure 11 EfficientNet-B7 Sub-blocks (Agarwal, 2020) Copyright Taken

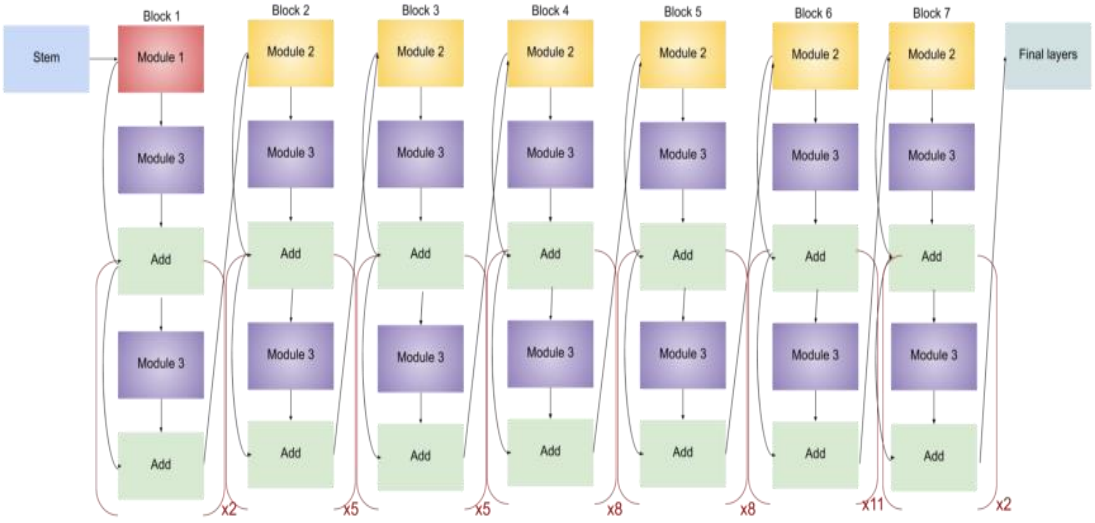

Figure 12 EfficientNet-B7 Architecture (Agarwal, 2020) Copyright Taken

2.1.5 Comparative models

2.1.5.1 VGG19. The VGG model has been proposed by the Visual Geometry Group at the University of Oxford in 2014 (Simonyan & Zisserman, 2014). VGG19 is a version of the VGG model that includes 19 layers in total. VGG19 is a pre-built model that has been trained on the ImageNet dataset to recognize one thousand different images. ImageNet is a collection with about 14 million photos divided into over one thousand categories. As initial weights for the proposed DNN model, ImageNet pre-trained weights were provided. In addition to ImageNet, the VGGNet performs baselines with a wide range of datasets and tasks. Furthermore, it remains one of the most popular image recognition models today.

The basic architecture of the VGG19 model consists of an input layer followed by 2 convolution layers with 64 filters ending with max-pooling followed by 2 convolution layers with 128 filters ending with max-pooling followed by 4 convolution layers with 256 filters ending with max-pooling followed by 4 convolution layers with 512 filters ending with max-pooling followed by the same previous 4 layers followed by 2 fully connected layers with 4096 nodes then 1 fully connected layers with softmax activation of 1000 nodes.

The input image size for the VGG19 is 224 × 224 pixels. VGG's convolutional layers utilize a small receptive field 3 × 3, the smallest size that still captures up/down and left/right movement. Then there's a ReLU activation function. The VGG19 is made up of three layers that are all connected. Each of the first two layers has 4096 nodes, whereas the third layer has 1000 nodes, which is the same to the total number of classes in the ImageNet dataset. The total number of parameters of the VGG19 is 143 million parameters.

2.1.5.2 ResNet50. As the deep learning-based network develops, its structure is going to be deeper; while this allows the network to perform more complicated feature pattern extraction, it also may cause the problem of gradient disappearance or gradient explosion. Gradient explosion or gradient disappearance can lead to the following problems: (1) Longer training period, but network convergence becomes difficult or even not convergent. (2) Network performance will progressively become saturated and going to be deteriorated, which is known as the deep network degradation problem. To resolve these issues, (He, Zhang, Ren, & Sun, 2015) presented a new DNN in 2015 that uses skip connections to connect layers with a large number of channels, allowing information to be passed deeper through the neural network. A residual block is created by using these skip connections (residual connections). The Residual Neural Network is made up of a collection of these residual blocks (ResNet). The authors discovered that residual blocks improve network performance and efficiency even when the number of network layers is very large (even more than 1000 layers).

ResNet50 is one of the ResNet types that consists of 48 convolution layers: 3 stages (2 convolution layers with 64 filters one convolution layer with 256 filters) followed by 4 stages (2 convolution layers with 128 filters one convolution layer with 512 filters) followed by 6 stages (2 convolution layers with 256 filters one convolution layer with 1024 filters) followed by 3 stages (2 convolution layers with 512 filters one convolution layer with 2048 filters) with one MaxPool layer, and one Average Pool layer then one fully connected layer with softmax activation of 1000 nodes (He et al., 2015).

ResNet-50 is a pre-built model that was trained on the ImageNet dataset to recognize 1,000 different images. As initial weights for the proposed DNN model, ImageNet pre-trained weights were provided. ResNet50's residual layers play a significant role in transferring big gradient values to their preceding neighboring layers. The model can extract complicated and relevant patterns and solve the vanishing gradient problem. The total number of parameters of the ResNet50 is 25.6 million.

2.1.5.3 MobileNetV2. MobileNet (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018) is an efficient and lightweight CNN architecture that is applied in real-world applications. MobileNets often use depthwise separable convolutions as opposed to the conventional convolutions utilized by earlier architectures in order to improve the lighter models. MobileNets introduces two additional global hyperparameters (resolution multiplier and width multiplier) that enable model developers to trade-off between accuracy or latency for small size and speed, based on their needs. MobileNets are constructed using convolution layers that are depth-separable. Each depthwise separate convolution layer consists of a depthwise convolution with a pointwise convolution.

A MobileNet has a total of 28 layers when depthwise convolutions and pointwise convolutions are calculated independently, while MobileNetV2 has a total of 53 layers. MobileNet parameters can be reduced to 4.2 million by adjusting the width multiplier hyperparameter whereas the amount of parameters in MobileNetV2 is 3.4 million. On ImageNet, MobileNetV2 performed effectively, scoring 71.8% top-1 and 97.3% top-5 accuracy. More details about the architecture of the MobileNetv2 model can be obtained from its research paper (Sandler et al., 2018).

2.2 Related Work

Face mask detection is a significant challenge due to the lack of big datasets on masked faces, as well as a lack of facial signals in the masked areas. The authors (Ge, Li, Ye, & Luo, 2017) presented the MAFA dataset (Masked Faces). Faces inside this dataset come with a range of orientations and occlusion levels and at least one part of each face is obscured. They also suggested Locally Linear Embedding-Convolutional Neural Networks (LLE-CNNs) for masked face detection based on this dataset. Many lost face signals may be retrieved in this way, and the effects of noisy signals supplied by various masks might be considerably reduced. As a result, the accuracy for the MAFA testing set reaches up to 76.4%.

In (Batagelj et al., 2021a) they tested many contemporary face detectors for their performance with masked-facial images. They also looked into the use of a variety of the off deep-learning models for identifying proper face-mask positioning. Eventually, they created a comprehensive pipeline for determining whether or not face masks are worn

appropriately, and compared the pipeline's efficiency with standard face mask detection models. They used the publicly accessible MAFA (Ge et al., 2017), Wider Face (Yang, Luo, Loy, & Tang, 2016) databases to build a large collection of face images which is Face-Mask-Label-Dataset (FMLD) (Batagelj, Peer, Štruc, & Dobrišek, 2021b). The accuracy for masked faces is 73%, whereas the accuracy for unmasked faces is over 90%.

To get high-performance results in (Loey, Manogaran, Taha, & Khalifa, 2021b), they used the YOLOv2 based ResNet-50 model. By incorporating mean Intersection over Union (IoU) to estimate the appropriate number of anchor boxes, the suggested model increases detection performance. They created a new dataset based on two available masked face datasets to train and validate their detector in a supervised state. For SGDM (I. Sutskever, Geoffrey H., G. Dahl, J. Martens) (Sutskever, Martens, Dahl, & Hinton, 2013) and Adam optimizer tests, performance indicators such as average precision and log-average miss rates score were also investigated. They demonstrated that the suggested YOLOv2 with ResNet-50 model architecture is an effective model for detecting the face mask. As a result, by using YOLOv2 with ResNet-50 based on Adam optimizer the average accuracy was 81%.

An autonomic face mask detection system supported by image tracking approach used for augmented reality development is suggested in (Benitez-Baltazar, Pacheco-Ramírez, Moreno-Ruiz, & Nuñez-Gurrola, 2021) as a method to demand the right usage of face masks to permit access to individuals to crucial regions. To accomplish this, an ML model based on CNNs was created on top of an internet of things framework to ensure proper face mask wear in specific locations, as required by law in some places. Samples from the Kaggle face mask detection dataset (Gurav, 2020) were used to train their model. This study got 96% as an average accuracy for face mask detection.

In (Nagrath et al., 2021) they presented an image classifier called SSDMNV2 that uses OpenCV DNN, TensorFlow, Keras, and MobileNetV2 architecture for face mask detection. SSDMNV2 uses OpenCV DNN, which includes SSD when SSD is a Single Shot Multibox Detector using as a face detector with the ResNet-10 model as a backbone and it is able to detect faces from any angles. When MobileNetV2 is applied,

it provides light and accurate classification to predict whether there is a mask or not. SSDMNV2 can tell the difference between images with masks on the frontal faces and images without masks. The accuracy score for the technique used in the paper is 92.64%.

(Saravanan, Karthiha, Kavinkumar, Gokul, & Mishra, 2022) proposed the VGG16 model which is a pretrained DL model. The suggested method is simple to implement because it uses all of the layers in the VGG16 model but it just trains the last layer, the FC layer, when it cuts down on training time and effort. They have used two datasets for face masks (one containing 1484 photos and 7200 for the other) to train and evaluate the suggested technique. To improve accuracy on a smaller dataset, enhanced images were used. The proposed model has been tested on unknown images and properly could predict if the person in the image is put on a face mask or not. During testing on a small dataset, the suggested model achieves an accuracy of 96.50%. And an accuracy of 91% when it was tested for a medium dataset. The performance and accuracy of the dataset are improved by employing the VGG16 pre-trained model with image augmentation.

The authors (Sanjaya & Rakhmawan, 2020) introduced face mask detection, which can be used by authorities to prevent the diffusion of COVID-19. Face mask detection is implemented in this work utilizing an ML technique with the MobileNetV2 image classification model. This study applies 1916 images with a face mask and 1930 images without a face mask to build a classification model. The image is cropped until the face of the object is the only thing visible. The procedures for creating the model are collecting the data, pre-processing, splitting the data, testing the model, and applying the model. The model has a 90.52% accuracy rate in detecting if the person is wearing a mask or not.

As a result of the consequence of the Covid-19 pandemic, and in order to monitor people's adherence to the procedures of wearing a face mask, which reduces the risk of infection by 40%, according to the researchers (Su et al., 2022). They proposed a deep learning face mask detection system and combined Efficient-Yolov3 with transfer learning. They used EfficientNet as a backbone network to extract features from the image. In order to train the model, the MAFA dataset (Ge et al., 2017) was used for the

images with the face mask, while the WIDER face dataset (Yang et al., 2016) was used for the images without the face mask. They achieved an accuracy of 96.02% for face mask detection using Efficient-Yolov3 and for mask classification they used MobileNet with transfer learning and they achieved an accuracy of 97.84%. The research explained that the algorithms used can efficaciously detect the face mask and contribute to limiting the spread of this pandemic.

(Habib et al., 2022) suggested an effective model in order to perform face detection, and they relied on the MobileNetV2 architecture, which extracts features from the image and they used the augmentation technique in order to increase the number of samples for training. An FMD (Gurav, 2020) dataset for real face masks was used, also an FM (Oumina, el Makhfi, & Hamdi, 2020) dataset for artificial face masks was used too. This model was compared with some recent models, including EffecientNet (without mentioning which version was used) and it obtained an accuracy of 87.02% for the original data and 96.97% for the augmented data for the FMD dataset. It obtained an accuracy of 83.43% for the original data and 91.49% for the augmented data for the FM dataset.

(Balasubramaniam, 2021) proposed a system to automate the face mask detection process based on deep learning by using the EfficientNet-B0 architecture. A dataset of masked faces (Gurav, 2020) was used to conduct training and testing, and this system achieved an accuracy of 97.12%.

(Setyanto, Kusrini, Sasongko, Permana, & Saputra, 2021) made comparisons to identify some of the face mask detection algorithms, including the EfficientNet-B0 model. It was trained on two datasets of compelling and unconvincing images. It achieved with dataset 1 (Jangra, 2020) a validation accuracy of 66.54%, while with dataset 2 (Kumar, 2021) it achieved a validation accuracy of 91.54%.

(Eyiokur, Ekenel, & Waibel, 2021) have created a system that detects the face mask as well as touching the face by hand. They also created two datasets for face masks (ISL-UFMD) and for hand-face interaction (ISL-UFHD) based on previous datasets and tested them later. They also made comparisons with recent face detection models, where the Inception-v3 model had an accuracy of 98.20%, while the EfficientNet-B3 model

achieved an accuracy of 98.19% on the ISL-UFMD dataset. When testing the presence of the hand on the face, the EfficientNet-B2 model scored the highest classification accuracy of 93.35% on the ISL-UFHD dataset.

Chapter 3: Methodology

3.1 Datasets

In ML models, the dataset is the most crucial aspect. Without a high-quality training dataset, even the most complex computer algorithms could be impractical. Therefore, the quality of training data is more crucial than other things in ML approaches. Training data refers to the initial data that was used for the model to improve itself. The model's continued development relies heavily on the dataset's quality which will assist to build a strong model for future purposes that may use the exact training data. Training data requires human input to review and develop the dataset for ML methods.

In this work, an EfficientNet-B7 model proposed by (Tan & Le, 2019) was applied to face mask detection problem. This model is based on DL and, in particular, transfer learning. The selected model has been compared to other models using the same dataset. Basic identification features are usually included in the dataset. These features will make it simpler for the model to be trained and learned. As a consequence of this, the extracted features from the dataset with the quality of the labeling, have a worthy impact on the accuracy of the model and the capability for detection of the outcomes and making high-quality predictions. As an outcome of this, the fundamental objective of this study is to build a model that is capable of classifying the faces of people by employing a face mask detector. This technique could be employed to encourage the uses of the face masks.

In order to train this model, 3000 images (1500 masked and 1500 unmasked face images) have been used from two types of datasets. These datasets will be discussed in the next section.

3.1.1 Masked faces. MAFA (MAsked FAces) (Ge et al., 2017) is a masked face dataset made up of photos gathered from the internet. MAFA includes 35,806 masked faces and 30,811 photos. Faces in the MAFA dataset feature a range of orientations and occlusion levels and at least one part of each face is obscured. Every image comprises minimally one face covered by different forms of masks during the annotation process, and the six key labels are recorded of each masked face, including positions of eyes, faces, and the masks, face direction, occlusion level, and the type of the mask. The majority of the faces in the MAFA dataset are frontal faces, as seen in Figure 13, with only some faces being entirely left or right. Face detectors that depend on accurate eye location may be examined in this way, and difficult scenarios like right-front and left-front faces could be utilized to assess their strength further. Furthermore, most people's faces have moderate occlusion, which is the most prevalent condition we see in cold weather conditions. The MAFA dataset includes a wide range of masked face scenarios that we see on a regular basis. MAFA may be utilized to establish a comprehensive baseline of face mask detectors for all types of the masked faces in this way.

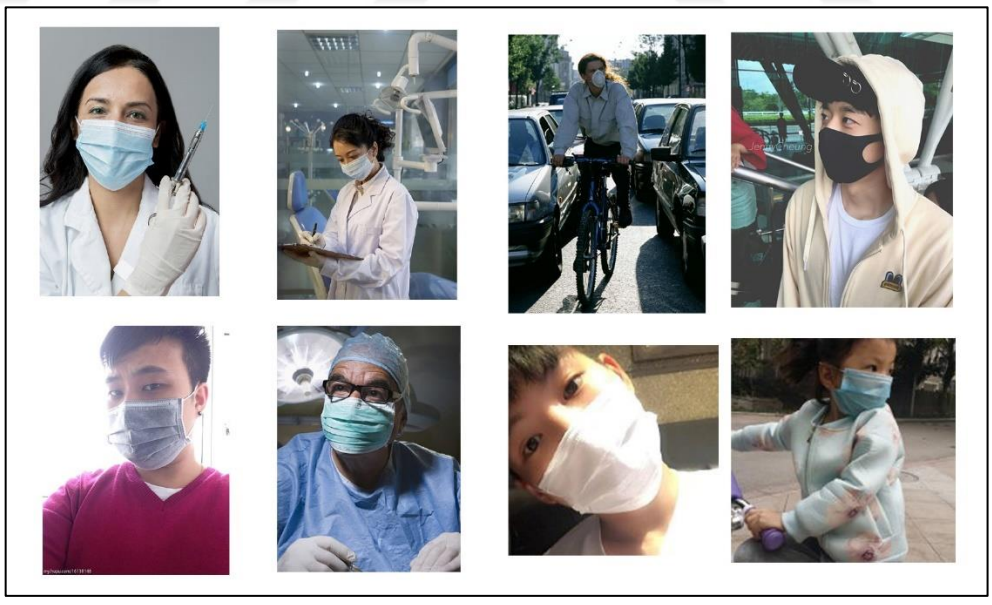

Figure 13 The Selected Images from the MAFA Dataset

3.1.2 Unmasked faces. LFW (Labeled Faces in the Wild) is a database of face photos created to investigate the challenge of unconstrained face recognition. The dataset includes over 13,000 photos of unmasked faces gathered from the internet. Figure 14 shows samples of the LFW images. LFW was first released in 2007 with the aim of encouraging face detection research, specifically for the problem of face verification with unconstrained images. The majority of face databases were developed under controlled conditions to assist in the study of specific parameters related to face recognition. Position, stance, illumination, background, photo quality, and gender are some of these parameters. The database comprises captioned face images that depict a wide range of situations that people face on a daily basis. Pose, illumination, ethnicity, accessories, occlusions, and environment all show "natural" variation in the database. It not only describes the database in detail but also gives appropriate experimental paradigms whereby the database is appropriate. This is done to ensure that database-based research is as consistent and similar as feasible (Huang, 2007; Huang, Mattar, Berg, & Learned-Miller, 2008).

Figure 14 The Selected Images Sample from the LFW Dataset

3.2 Model Architecture

This is the path that our EfficientNet-B7 model follows to achieve our goal. Many of these stages are concerned with preparing the data and ensuring that it is suitable for

the selected model. The flowchart in Figure 15 shows the process of the general operational architecture for the selected face mask detection model.

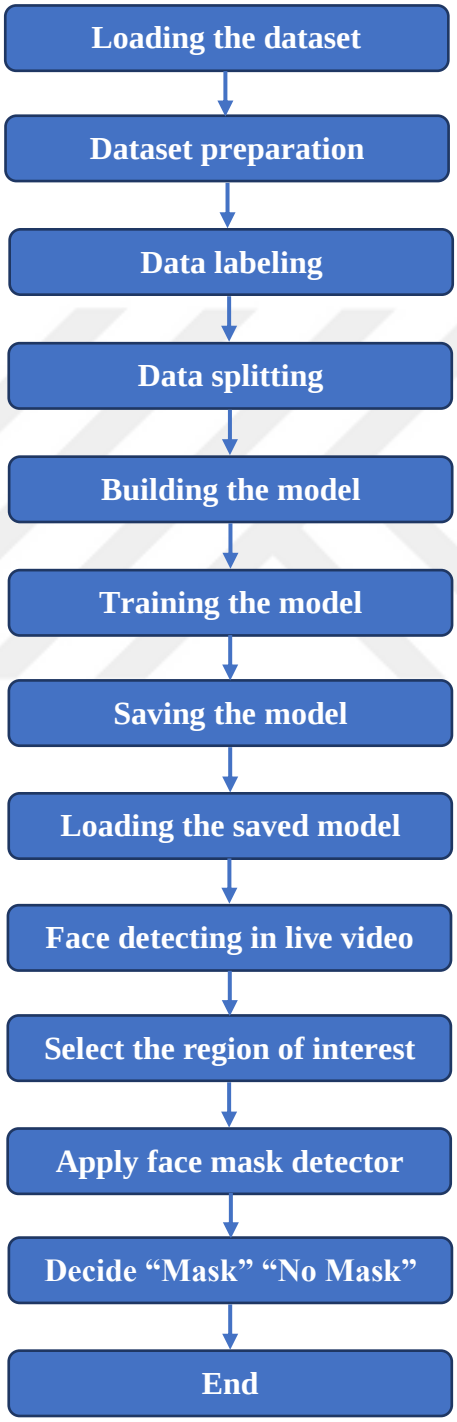

Figure 15 The General Operational Architecture

3.3 Dataset Preparation

3.3.1 Data preprocessing. After loading the dataset, it is common to resize images to smaller dimensions. Therefore in this work, they were resized to 128 by 128 to permit mini-batch learning while also staying within the computational limitations. This step lowered the input size, which is required by EfficientNet-B7. Although the size shrinks, the pixel value remains constant, hence this factor has no great impact on the quality of the input or the outcomes. Then we will change the type of the data to (float32) in order to normalize the image to a range between [0, 1] by dividing the input value by 255. This step is necessary to avoid large calculations when the input values are multiplied by weight values.

3.3.2 Data labeling. Despite the fact that the images are available, we must adapt them to this model for training. The use of "X" and "Y" as shown below is required to do this (X, Y).

X refers to images.

Y refers to labels.

We have two labels [‘0’ represents ‘with_mask’, ‘1’ represents ‘without_mask’]. The model requires both the images we have imported and the labels that go with them to distinguish between images of people with and without face masks. The images are then separated using the labels that have been assigned to them.

3.3.3 Data splitting. All of the images, and their corresponding labels, will undergo training and testing. In order to determine the efficiency of the selected model, we use 80% of the total images for training and 20% for testing. At this point, we divide the dataset into two parts: the training and testing sets. The training set contains the images that will be used to train the selected model, while the testing set will contain images that will be used to assess the performance of the selected model.

The model will first read the labels, examine the patterns, and then classify them as zeros and ones. These values indicate that if the model is given X as a Numpy array input, it will take it in the form of zeros and ones to return either "with mask" or

"without mask" output. Our X and Y have been fully adjusted for the selected ML model.

3.4 The Implementation of the Selected Model

3.4.1 Building the model. To import EfficientNet-B7 in the work, CNN structures from Tensorflow and the Keras library are used. With transfer learning, the selected model has already pre-learned on 1000 classes through the Imagenet dataset, thus the last layer, which is totally prediction-oriented, will be removed, leaving only the layers with adequate knowledge of feature extraction. Furthermore, instead of the thousands of output neurons associated with transfer learning, the selected model will just have two output neurons to identify between "with-mask" and "without-mask".

3.4.2 Training the model. First we will use Adam optimizer which is a neural network adaptive learning rate algorithm that scales the learning rates by adjusting the learning rates per individual parameter.

Then the learning rate will be set to 0.0001 since the learning rate is a major parameter that affects the construction of neural networks. The purpose of the learning rate is to change the model in response to estimated mistakes. The model changes the weights every updated cycle. In summary, learning rate refers to how quickly a model learns to achieve suppressed loss and hence achieve minimal error. Selecting a value that is too low for the learning rate results in a lengthy training procedure with stutters and lags. Setting a value for the learning rate that is too high speeds up the training process but degrades reliability.

Next, as initial weights for our model, ImageNet pre-trained weights were used. Since EfficientNet-B7 is a pre-built model that has been trained on the ImageNet dataset to recognize one thousand different images. As is the case with other images classification models such as including but not limited to VGG19, MobileNetV1, MobileNetV2, VGG16, and ResNet50.

After that, we will set the value of epochs to 30 since epochs are the number of times the model is allowed to traverse over the images in the datasets. Our neural network model must process the data multiple times in order to maximize or optimize

learning. Underfitting is caused by a low epoch value. On the contrary more epochs will increase the probability of overfitting.

The dataset is too large for our neural network model to process it all at once. Batches coordinate this operation by splitting the dataset into smaller chunks. Therefore, we set the batches to 16 in the selected model.

3.5 Mask Detection Testing Using Video

Now after the model has been trained, it can be implemented for detecting the presence of a mask through a real-time video stream. We ran the tests on an Intel(R) Core i7-8750H CPU @ 2.20GHz processor and 16 GB of RAM with an NVIDIA GeForce GTX 1,060 graphics card and Windows 11. Python 3.7 was used as the programming language in this work.

First of all, the pre-trained model will be loaded. Then we will prepare the camera to receive the video, as is known, each video is made up of a set of frames and each frame will be treated as a separate image. In this work the camera frame rate is 30 fps. Then, these images will be passed as an input to the selected model. It will extract the hidden image features and label them as 'Mask' or 'No Mask'.

But before that, we need to perform face detection using one of the available models included in the OpenCV library which is Caffemodel (Jia et al., 2014) for our implementation of the selected model. It is built on the Single Shot Multi-box Detector (SSD) using the 'ResNet-10' architecture as a backbone. It is capable of detecting face and facial features in an image or in real-time video from a variety of angles due to its efficiency and speed. After implementing the selected face detection model, it extracted the region of interest (ROI) and received the total number of faces found and their bounding boxes' locations and also the prediction's confidence score. These outputs will be inputs for the selected face detection model. This will allow it to predict the presence of the mask in different orientations and sizes in real-time video with high accuracy.

The preprocessing steps will be summed up by converting the BGR image to an RGB image since OpenCV uses BGR image format for reading the image and then resizing the image down to 128 by 128 similar to the sizing that opted for the images in

the training process. Also, it will apply the normalization process in order to ensure that it is suitable for the selected model.

If the return value from the selected detection model is greater than the threshold point which is 0.5, then the model will label the image with “No Mask” otherwise if the returned value is less than the threshold point the model will label the image with "Mask".

Finally, if the person is wearing a face mask it will appear in a green color square with the predictive value as shown in

Figure 16 otherwise, it will determine the person is not wearing a face mask with a red color square with the predictive value as shown in

Figure 17. Also in Figure 18 it shows face mask detection for multiple colors of mask and multiple face directions.

Figure 16 The Selected Model Detection (Front/Side, White Mask)

Figure 17 The Selected Model Detection (Front/Side, No Mask)

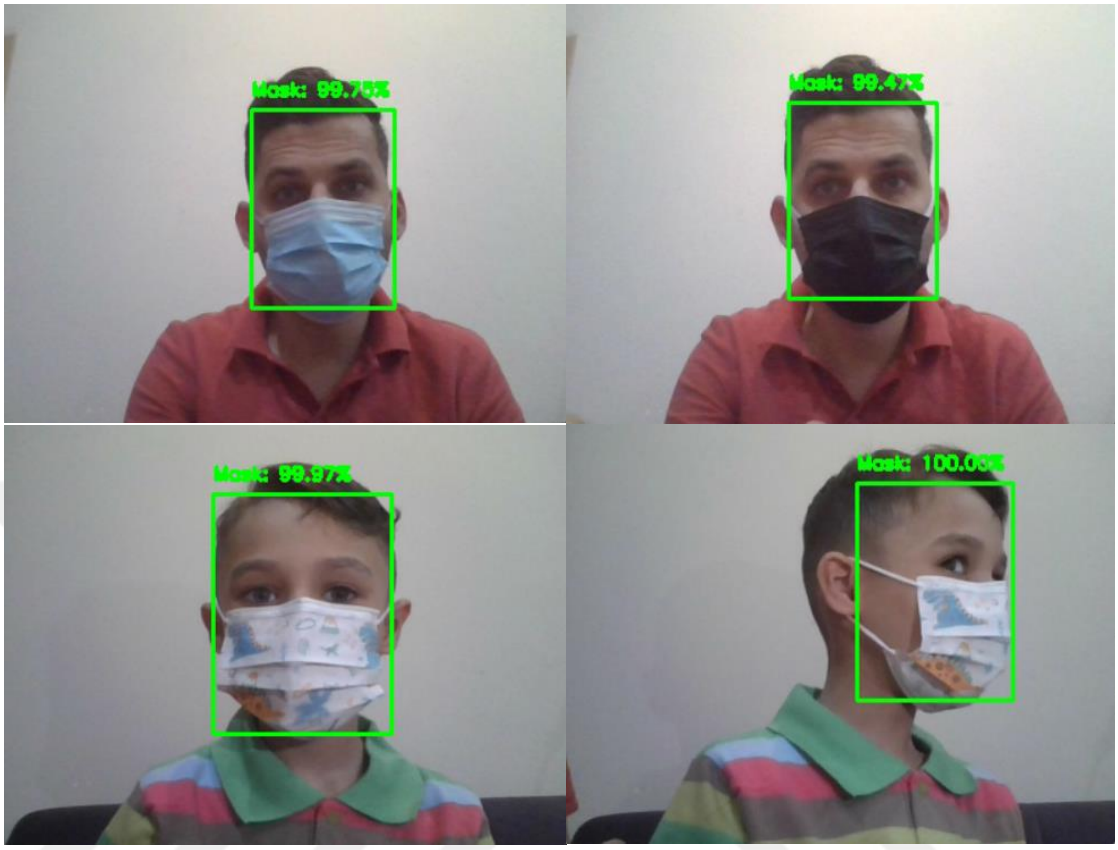

Figure 18 Face Mask Detection for Multiple (Mask Color, Face Direction)

Chapter 4: Results

In order to obtain reliable results, we trained and tested the selected model "EfficientNet-B7" for a value of epochs up to 30 and for 3000 samples of the dataset (MAFA & LFW). Besides, we did the same for the three other models which are (VGG19, MobileNetV2, and ResNet50). In order to get more consistent results, we ran the algorithms 10 times for each model. We reported the average testing accuracy, validation accuracy, and training accuracy values as shown in the Table 1.

Table 1 The Average Accuracy for the Models

Model	Average training accuracy	Average validation accuracy	Average testing accuracy
EfficientNet-B7	99.999	**98.866**	**98.87**
MobileNetV2	99.999	95.916	95.917
VGG19	99.387	86.833	86.834
ResNet50	71.32	68.316	68.317

A one-way ANOVA on test accuracies showed that there are differences in the means of those 4 models ($p < 0.001$).

We applied a paired t-Test between the two models (i.e. EfficientNet-B7, and MobileNetV2) which generated the highest average test accuracy results. The result of this comparison showed that the EfficientNet-B7 model was superior to the MobileNetV2 ($p < 0.001$).

With the selected model "EfficientNet-B7" we achieved an average accuracy of 98.87% as presented in Figure 19 while the average accuracy in the MobileNetV2 model was 95.91% as shown in Figure 20 and in the VGG19 model was 86.83% as present in Figure 21 and in the ResNet50 model was 68.31% as shown in Figure 22.

The epochs versus average loss functions were given in (Figure 23, Figure 24, Figure 25, Figure 26) for EfficientNet-B7, MobileNetV2, VGG19, and ResNet50 model, respectively. By observing the stability or the deterioration of the loss function for all

models by calculating the average of the validation losses for the ten training times. It was noted that the optimal epoch value for EfficientNet-B7 was at 40, while it was 18 for the MobileNetV2 model, 10 for VGG19, and 39 for ResNet50 and the accuracies for each model are shown in the Table 2.

Table 2 Models Accuracy of Optimal Epoch Values

Model	Training accuracy	Validation accuracy	Testing accuracy
EfficientNet-B7	99.999	**99.11**	**99.111**
MobileNetV2	99.999	96	95.999
VGG19	94	86.83	86.833
ResNet50	69.75	68.33	68.333

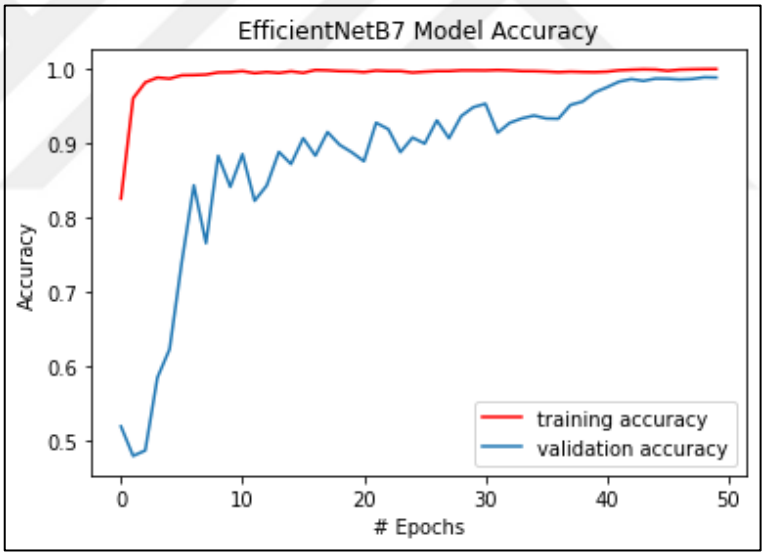

Figure 19 EfficientNet-B7 Average Accuracy

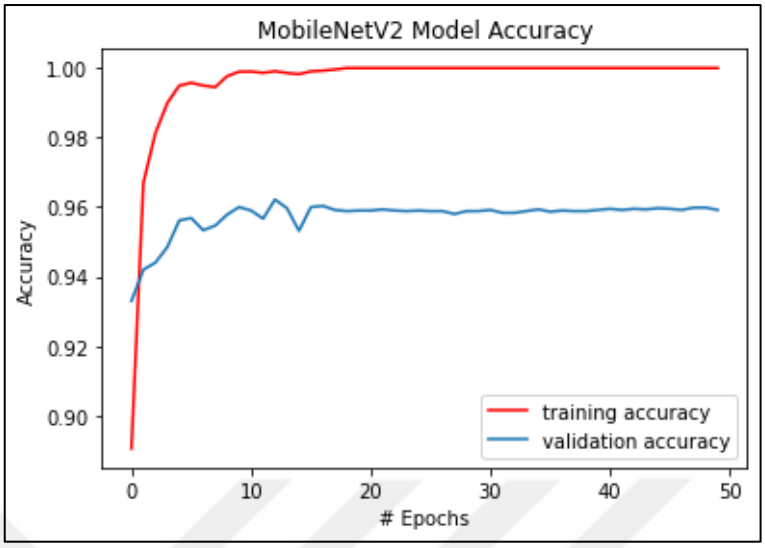

Figure 20 MobileNetV2 Average Accuracy

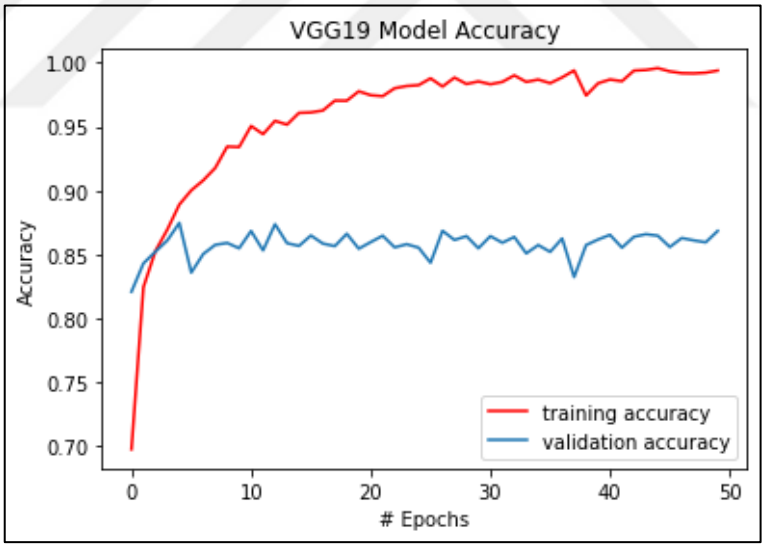

Figure 21 VGG19 Average Accuracy

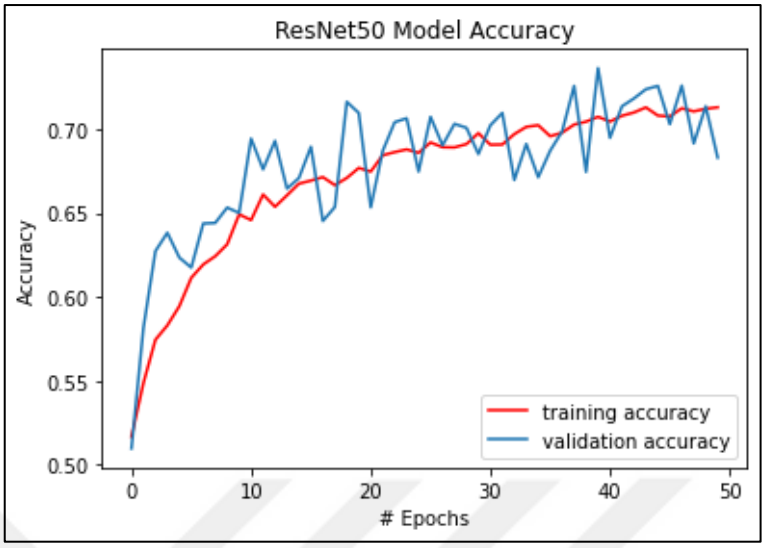

Figure 22 ResNet50 Average Accuracy

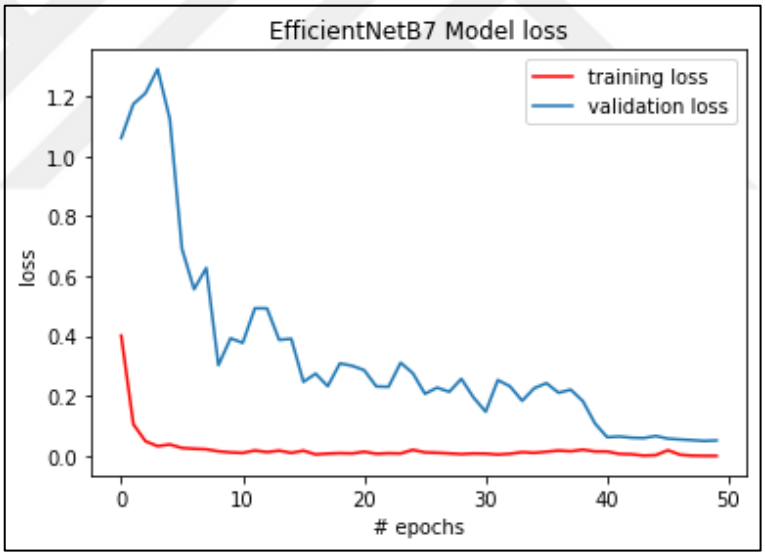

Figure 23 EfficientNet-B7 Average Loss

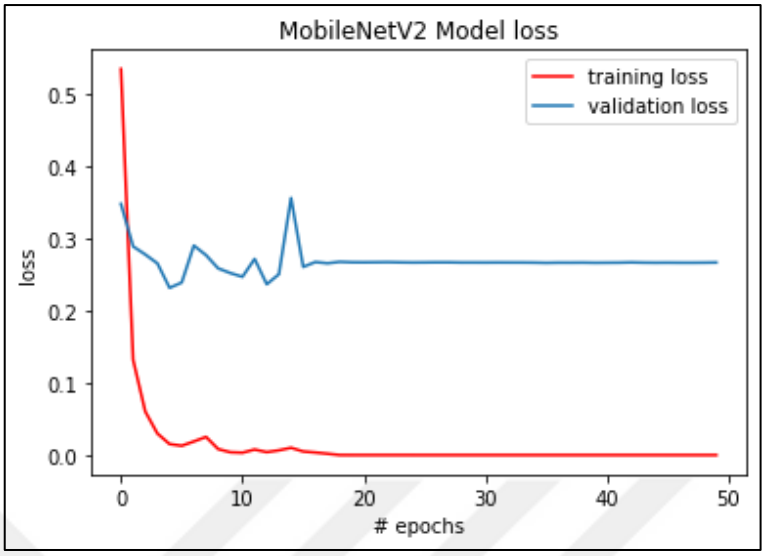

Figure 24 MobileNetV2 Average Loss

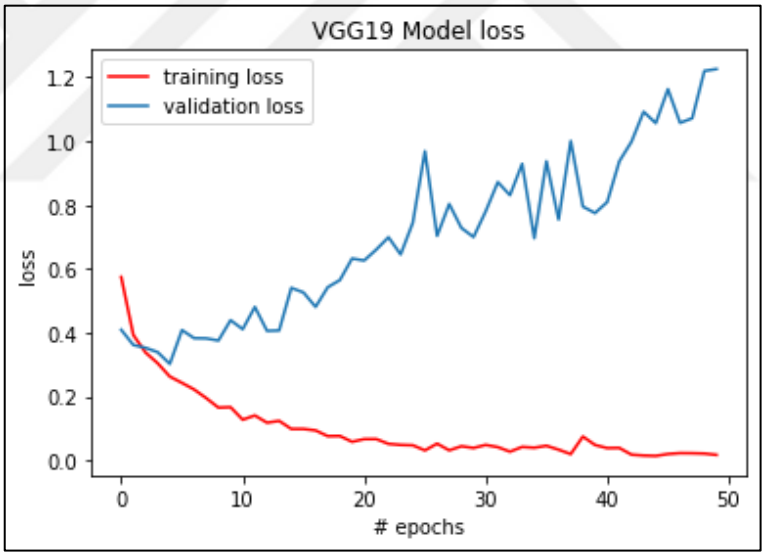

Figure 25 VGG19 Average Loss

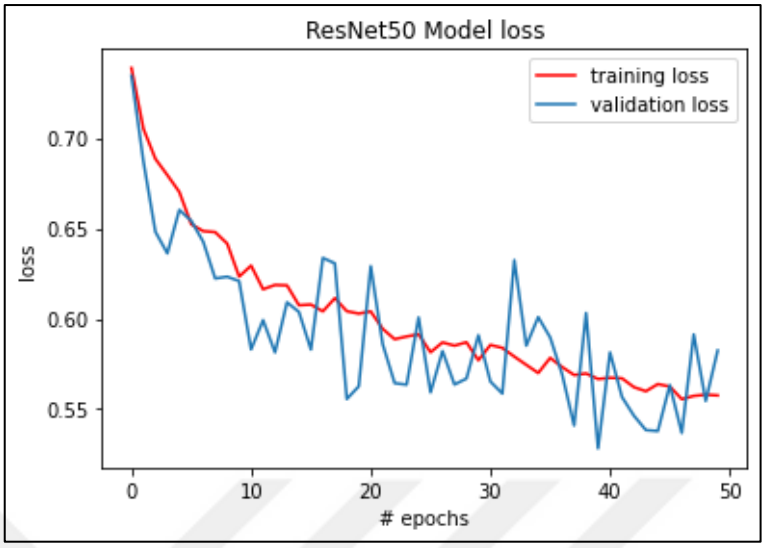

Figure 26 ResNet50 Average Loss

Some processes were performed in order to calculate the images that were misclassified after training and testing the model ten times by summing all the images that appeared repeatedly, as shown in Figure 27. Through visual inspection, it is seen that some faces have been classified incorrectly, either because of the presence of different face classes in the background, or because of gestures in the face or mouth, or through the disappearance of one of the eyes, or because some masks have many details, or because some parts of the face are covered, or because the face is rotated or flipped excessively.

Figure 27 Misclassified Images after Testing the Model

Chapter 5: Discussion

One of the factors that have an important role in the model training process is the selection of effective datasets. We selected MAFA Dataset for the masked faces and LFW for the unmasked faces. The most important feature of the MAFA dataset is that it contains a large number of real masked faces, while some research and training have been based on artificial masked faces, which do not give real results since the faces that will be tested later are real face masks. MAFA dataset is also distinguished by its content having a variety of orientations and degrees of occlusion, multiple lighting exposures, and different skin colors of faces, in addition to some faces being entirely left or right. It also contains images of different depths, including near and far faces, therefore this is what we may face when the system will be used in practice as shown in Figure 16. As a result, using the MAFA dataset with LFW has a significant impact on the model's accuracy and capability to detect the outcomes and make accurate predictions.

EfficientNet-B7, submitted by Google AI scientist Quoc V. Le and software scientist Mingxing Tan, was adopted as the selected model in this work for an important purpose. Previously the other CNNs models were scaled up randomly to achieve better accuracies on most benchmarking datasets. So they are scaled in terms of depth, width, and resolution but it requires manual tuning and many man-hours, often resulting in little or no performance increase. Unlike them, EfficientNet-B7 scales each dimension with a fixed set of scaling coefficients in a simple but effective manner using a technique called the compound scaling. While scaling individual dimensions improves model performance, balancing the scale in all three dimensions (width, depth, and image resolution) improves overall model performance the most.

Therefore, the selected EfficientNet-B7 model surpassed the accuracy of the compared CNNs, and with better efficiency. The selected model achieved an accuracy rate of 98.87%, while the rest of the models that we compared with it obtained an accuracy of 95.91% for the MobileNetV2 model, 86.83% for the VGG19 model, and 68.31% for the ResNet50 model.

As mentioned in the previous chapter, since the selected model was compared with three other models for higher reliability and accuracy, all models were trained 10 times on the same datasets. A statistical analysis has been applied using one-way ANOVA to check whether there are any statistically significant differences between the means of the 4 models. As a result, there is sufficient evidence to admit that not all of the models' means are equal.

Once we found that there is a significant difference between the models' means then we applied a paired t-Test between the two models that generated the highest accuracy and the result showed that the EfficientNet-B7 model had significantly higher accuracy values than the MobileNetV2 model.

After a review of some literature in which EfficientNet models were used for several versions, Table 3 was made that shows the accuracy of each model with the dataset used as well as the used version of EfficientNet models. The promising performance of these models has been shown with various dataset of masked and unmasked face images.

Table 3 A Comparison Between the Selected Model & other EfficientNet Models in the Literatures

Author(s)	Model	Average Acc.	Dataset
(Su et al., 2022)	Efficient-Yolov3	96.02%	MAFA (Ge et al., 2017) WIDER (Yang et al., 2016)
(Habib et al., 2022)	EffecientNet	96.97% 91.49%	FMD (Gurav, 2020) FM (Oumina et al., 2020)
(Balasubramaniam, 2021)	EfficientNet-B0	97.12%.	FMD (Gurav, 2020)
(Setyanto et al., 2021)	EfficientNet-B0	66.54% 91.54%.	Dataset 1 (Jangra, 2020) Dataset 2 (Kumar, 2021)
(Eyiokur et al., 2021)	EfficientNet-B3	98.19%	ISL-UFMD (Eyiokur et al., 2021)
This study	**EfficientNet-B7**	**98.87%**	MAFA (Ge et al., 2017) LFW (Huang et al., 2008)

Chapter 6: Conclusion and Future Work

Due to one of the world's largest viral outbreaks in the previous five decades which is Covid-19, decision-makers around the world are urging residents to follow social distancing regulations and preventive measures. However, when viral transmissions become more widespread, greater pressure must be applied on people to respect the guidelines. Since locking down permanently isn't feasible, the WHO recommends mask use as a precautionary measure to reduce virus spread. Although, the risk posed by the Covid-19 may fade in the near future, especially as a result of virus mutations, the chance of new pandemics arising remains a possibility, thus face mask detection will always be an issue.

In this work, an EfficientNet-B7 model for face-mask detection is selected, which is based on DL and, in particular, transfer learning. The selected model implements face mask detection in real-time to check whether the person wearing a mask or not in order to test the violations of these rules set by WHO. The datasets that have been used for training and testing are MAFA for masked faces and LFW for unmasked faces. The selected model has been compared to other models in which we trained them using the same dataset as the selected model. The selected model achieved promising results compared to other comparative models. The selected model obtained an average accuracy of 98.87% while other models got an accuracy of 95.91% for the MobileNetV2 model, 86.83% for the VGG19 model, and 68.31% for the ResNet50 model. A statistical analysis has been applied using ANOVA and paired t-Test, it explained that there is a significant difference between the models' means accuracy values.

The selected model has significant implications for the current time of Covid-19 pandemic for developing detectors to monitor people's compliance with the obligation to wear a face mask in the hospitals, health facilities, laboratories, manufactories, transportation, and other places.

As a suggestion, future works aim to better improve the performance of the face mask detector model to achieve higher accuracy in detecting the face mask. Furthermore, to implement a facial recognition model that can be used in any facility

where personal identification is necessary, it aims to determine the personal identification when the person is wearing the mask. Finally, it is recommended the EfficientNet-B7 model be used in the upcoming facial or object detection issues due to its high efficiency and accuracy.

REFERENCES

Agarwal, V. (2020). Complete Architectural Details of all EfficientNet Models | by Vardan Agarwal | Towards Data Science. Retrieved May 1, 2022, from https://towardsdatascience.com/complete-architectural-details-of-all-efficientnet-models-5fd5b736142

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., … Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data 2021 8:1*, 8(1), 1–74. Retrieved April 6, 2022 from https://doi.org/10.1186/S40537-021-00444-8

Andreopoulos, A., & Tsotsos, J. K. (2013). 50 Years of object recognition: Directions forward. *Computer Vision and Image Understanding*, 117(8), 827–891. Retrieved April 6, 2022 from https://doi.org/10.1016/J.CVIU.2013.04.005

Balasubramaniam, V. (2021). Facemask Detection Algorithm on COVID Community Spread Control using EfficientNet Algorithm. *Journal of Soft Computing Paradigm*, 3(2), 110–122. Retrieved July 3, 2022 from https://doi.org/10.36548/JSCP.2021.2.005

Banko, M., & Brill, E. (2001). Scaling to very very large corpora for natural language disambiguation. *Presented at the Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, 26–33. Retrieved April 5, 2022 from https://dl.acm.org/doi/10.3115/1073012.1073017

Batagelj, B., Peer, P., Štruc, V., & Dobrišek, S. (2021a). How to Correctly Detect Face-Masks for COVID-19 from Visual Information? *Applied Sciences*, 11(5), 2070.

Batagelj, B., Peer, P., Štruc, V., & Dobrišek, S. (2021b). How to Correctly Detect Face-Masks for COVID-19 from Visual Information? *Applied Sciences 2021, Vol. 11, Page 2070*, 11(5), 2070. Retrieved January 16, 2022 from https://doi.org/10.3390/APP11052070

Benenson, R., Omran, M., Hosang, J., & Schiele, B. (2014). Ten Years of Pedestrian Detection, What Have We Learned? *Lecture Notes in Computer Science (Including*

Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8926, 613–627. Retrieved April 6, 2022 from https://doi.org/10.48550/arxiv.1411.4304

Benitez-Baltazar, V. H., Pacheco-Ramírez, J. H., Moreno-Ruiz, J. R., & Nuñez-Gurrola, C. (2021). Autonomic Face Mask Detection with Deep Learning: an IoT Application. *REVISTA MEXICANA DE INGENIERÍA BIOMÉDICA* , 42(2), 160–170. Retrieved from https://doi.org/10.17488/RMIB.42.2.13

Bisong, E. (2019). *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Apress. Retrieved April 5, 2022 from https://doi.org/10.1007/978-1-4842-4470-8

Brimblecombe, P. (2002). *Face Detection using Neural Networks* (H615 ed.). Meng Electronic Engineering School of Electronics and Physical Sciences.

Brownlee, J. (2019). *Deep Learning for Computer Vision - Image Classification, Object Detection and Face Recognition in Python* (1.4). Machine Learning Mastery. Retrieved April 6, 2022 from https://machinelearningmastery.com/deep-learning-for-computer-vision/

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, I, 886–893. Retrieved April 6, 2022 from https://doi.org/10.1109/CVPR.2005.177

Dwivedi, D. (2018). Face Detection For Beginners. In the past few years, face recognition… | by Divyansh Dwivedi | Towards Data Science. Retrieved January 16, 2022, from https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9

Eyiokur, F. I., Ekenel, H. K., & Waibel, A. (2021). Unconstrained Face-Mask & Face-Hand Datasets: Building a Computer Vision System to Help Prevent the Transmission of COVID-19. Retrieved July 3, 2022 from https://doi.org/10.48550/arxiv.2103.08773

Felzenszwalb, P. F., Girshick, R. B., & McAllester, D. (2010). Cascade object detection with deformable part models. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2241–2248. Retrieved April 6, 2022 from https://doi.org/10.1109/CVPR.2010.5539906

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 1627–1645. Retrieved April 6, 2022 from https://doi.org/10.1109/TPAMI.2009.167

Ge, S., Li, J., Ye, Q., & Luo, Z. (2017). Detecting Masked Faces in the Wild with LLE-CNNs. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017-January, 426–434. Retrieved January 16, 2022 from https://doi.org/10.1109/CVPR.2017.53

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? the KITTI vision benchmark suite. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 3354–3361. Retrieved April 6, 2022 from https://doi.org/10.1109/CVPR.2012.6248074

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. Retrieved April 5, 2022 from https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/

Gurav, O. (2020). Face Mask Detection Dataset | Kaggle. Retrieved July 3, 2022, from https://www.kaggle.com/datasets/omkargurav/face-mask-dataset

Habib, S., Alsanea, M., Aloraini, M., Al-Rawashdeh, H. S., Islam, M., & Khan, S. (2022). An Efficient and Effective Deep Learning-Based Model for Real-Time Face Mask Detection. *Sensors 2022, Vol. 22, Page 2602*, 22(7), 2602. Retrieved July 3, 2022 from https://doi.org/10.3390/S22072602

Halevy, A., Norvig, P., & Pereira, F. (2009). The Unreasonable Effectiveness of Data. *IEEE Intelligent Systems*, 24(02), 8–12. Retrieved April 5, 2022 from https://doi.org/10.1109/MIS.2009.36

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December, 770–778. Retrieved April 20, 2022 from https://doi.org/10.48550/arxiv.1512.03385

Higuchi, S., Taniguchi, S., Kawasaki, Y., & Sonoda, A. (2021). Image Processing for the Prevention of Infectious Diseases: Determination of Mask Wearing, Measurement of Hand Washing Time, and Disinfection Support System. *New Generation Computing*, 39(3–4), 1. Retrieved January 16, 2022 from https://doi.org/10.1007/S00354-021-00137-Z

Hjelmås, E., & Low, B. K. (2001). Face Detection: A Survey. *Computer Vision and Image Understanding*, 83(3), 236–274. Retrieved January 16, 2022 from https://doi.org/10.1006/CVIU.2001.0921

Huang, G. (2007). Labeled Faces in the Wild Database (LFW) . Retrieved April 21, 2022, from http://vis-www.cs.umass.edu/lfw/

Huang, G., Mattar, M., Berg, T., & Learned-Miller, E. (2008). Labeled Faces in the Wild: A Database forStudying Face Recognition in Unconstrained Environments. *Tech. Rep.*

Jangra, A. (2020). Face Mask Detection ~12K Images Dataset | Kaggle. Retrieved July 3, 2022, from https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., … Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia*, 675–678. Retrieved May 1, 2022 from https://doi.org/10.1145/2647868.2654889

Julian, D. (2016). *Designing machine learning systems with Python*. Retrieved April 5, 2022 from https://www.packtpub.com/product/designing-machine-learning-systems-with-python/9781785882951

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 60(6), 84–90. Retrieved April 6, 2022 from https://doi.org/10.1145/3065386

Kumar, vijay. (2021). Face Mask Detection | Kaggle. Retrieved July 3, 2022, from https://www.kaggle.com/vijaykumar1799/face-mask-detection

Kumar, A., Upadhyay, P., & Kumar, A. S. (2020). *Fuzzy Machine Learning Algorithms for Remote Sensing Image Classification*. *Fuzzy Machine Learning Algorithms for Remote Sensing Image Classification*. CRC Press. Retrieved April 5, 2022 from https://doi.org/10.1201/9780429340369

Kuncheva, L. I., & Faithfull, W. J. (2014). PCA feature extraction for change detection in multidimensional unlabeled data. *IEEE Transactions on Neural Networks and Learning Systems*, 25(1), 69–80. Retrieved April 6, 2022 from https://doi.org/10.1109/TNNLS.2013.2248094

Liang, M., Gao, L., Cheng, C., Zhou, Q., Uy, J. P., Heiner, K., & Sun, C. (2020). Efficacy of face mask in preventing respiratory virus transmission: A systematic review and meta-analysis. *Travel Medicine and Infectious Disease*, 36, 101751. Retrieved from https://doi.org/10.1016/j.tmaid.2020.101751

Loey, M., Manogaran, G., Taha, M. H. N., & Khalifa, N. E. M. (2021a). 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. Retrieved January 16, 2022 from https://doi.org/10.1016/J.SCS.2020.102600

Loey, M., Manogaran, G., Taha, M. H. N., & Khalifa, N. E. M. (2021b). 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. Retrieved January 16, 2022 from https://doi.org/10.1016/J.SCS.2020.102600

Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: a guide for data scientists*. Retrieved April 5, 2022 from https://smartnet.niua.org/sites/default/files/webform/ai-strategy/pdf-introduction-to-

machine-learning-with-python-a-guide-for-data-sc-andreas-c-mller-sarah-guido-pdf-download-free-book-aa67cdb.pdf

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. Retrieved April 20, 2022 from https://doi.org/10.1016/J.SCS.2020.102692

O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., … Walsh, J. (2020). Deep Learning vs. Traditional Computer Vision. *Advances in Intelligent Systems and Computing*, 943, 128–144. Retrieved April 5, 2022 from https://doi.org/10.1007/978-3-030-17795-9_10

O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. Retrieved April 5, 2022 from https://arxiv.org/pdf/1511.08458

Oumina, A., el Makhfi, N., & Hamdi, M. (2020). Control the COVID-19 Pandemic: Face Mask Detection Using Transfer Learning. *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science, ICECOCS 2020*. Retrieved July 3, 2022 from https://doi.org/10.1109/ICECOCS50124.2020.9314511

Pedregosa, F., Michel, V., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., … Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. Retrieved April 5, 2022 from https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf?ref=https://githubhelp.com

Prince, S. J. D. (2013). *Computer Vision: Models, Learning, and Inference*. *The Lancet Neurology* (Vol. 12). Retrieved April 6, 2022 from https://tnfarmlink.org/sites/default/files/pdf-computer-vision-models-learning-and-inference-dr-simon-j-d-prince-pdf-download-free-book-a9aed6e.pdf

Rahman, M. M., Manik, M. M. H., Islam, M. M., Mahmud, S., & Kim, J. H. (2020). An automated system to limit COVID-19 using facial mask detection in smart city network. In *IEMTRONICS 2020 - International IOT, Electronics and Mechatronics*

Conference, Proceedings. Institute of Electrical and Electronics Engineers Inc. Retrieved from https://doi.org/10.1109/IEMTRONICS51293.2020.9216386

Raschka, S. (2015). *Python machine learning*. Packt publishing. Retrieved April 5, 2022 from https://raw.githubusercontent.com/rasbt/python-machine-learning-book/master/docs/equations/pymle-equations.pdf

Runia, T. F. H. (2015). High-Speed Object Detection: Design, Study and Implementation of a Detection Framework using Channel Features and Boosting . Retrieved April 6, 2022, from http://resolver.tudelft.nl/uuid:166773b8-a748-43dc-9cd3-64f9753c0044

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 4510–4520. Retrieved April 20, 2022 from https://doi.org/10.48550/arxiv.1801.04381

Sanjaya, S. A., & Rakhmawan, S. A. (2020). Face Mask Detection Using MobileNetV2 in the Era of COVID-19 Pandemic. *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy, ICDABI 2020*. Retrieved April 20, 2022 from https://doi.org/10.1109/ICDABI51230.2020.9325631

Saravanan, T. M., Karthiha, K., Kavinkumar, R., Gokul, S., & Mishra, J. P. (2022). A novel machine learning scheme for face mask detection using pretrained convolutional neural network. *Materials Today: Proceedings*. Retrieved April 20, 2022 from https://doi.org/10.1016/J.MATPR.2022.01.165

Setyanto, A., Kusrini, K., Sasongko, T. B., Permana, A. B., & Saputra, A. P. (2021). Efficient Deep Learning Architecture for Facemask Detection. *ICOIACT 2021 - 4th International Conference on Information and Communications Technology: The Role of AI in Health and Social Revolution in Turbulence Era*, 119–124. Retrieved July 3, 2022 from https://doi.org/10.1109/ICOIACT53268.2021.9564011

Sharma, U. B. (2014). A survey on face detection methods and feature extraction techniques of face recognition. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, 3(3). Retrieved from www.ijettcs.org

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. Retrieved April 20, 2022 from https://doi.org/10.48550/arxiv.1409.1556

Solem, J. E. (2012). *Programming computer vision with Python*. O'Reilly. Retrieved April 6, 2022 from https://www.academia.edu/download/59009952/ProgrammingComputerVision_CCdraft20190423-34911-1rty4uh.pdf

Su, X., Gao, M., Ren, J., Li, Y., Dong, M., & Liu, X. (2022). Face mask detection and classification via deep transfer learning. *Multimedia Tools and Applications*, 81(3), 4475–4494. Retrieved July 2, 2022 from https://doi.org/10.1007/S11042-021-11772-5

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *International conference on machine learning* (pp. 1139–1147). PMLR.

Szeliski, R. (2010). *Computer Vision: Algorithms and Applications* (2nd ed.). Cham: Springer International Publishing. Retrieved April 6, 2022 from https://doi.org/10.1007/978-3-030-34372-9

Tan, M., & Le, Q. v. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *36th International Conference on Machine Learning, ICML 2019*, 2019-June, 10691–10700. Retrieved April 20, 2022 from https://doi.org/10.48550/arxiv.1905.11946

Tiwari, A., Kumar, A., & Saraswat, G. M. (2013). Feature extraction for object recognition and image classification. *International Journal 112 of Engineering Research & Technology*, 2, 0180–2278.

Treiber, M. A. (2010). *An Introduction to Object Recognition*. London: Springer London. Retrieved April 6, 2022 from https://doi.org/10.1007/978-1-84996-235-3

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Retrieved April 5, 2022 from https://www.researchgate.net/profile/Michael-Jones-66/publication/3940582_Rapid_Object_Detection_using_a_Boosted_Cascade_of_Simple_Features/links/0f31753b419c639337000000/Rapid-Object-Detection-using-a-Boosted-Cascade-of-Simple-Features.pdf

Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48, 144–156. Retrieved April 6, 2022 from https://doi.org/10.1016/J.JMSY.2018.01.003

Watt, J., Borhani, R., & Katsaggelos, A. (2020). *Machine learning refined: Foundations, algorithms, and applications*. Retrieved April 5, 2022 from https://dl.acm.org/doi/10.5555/3126125

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).

Zou, Z., Shi, Z., Guo, Y., & Ye, J. (2019). Object Detection in 20 Years: A Survey. Retrieved April 5, 2022 from https://arxiv.org/pdf/1905.05055.pdf%C2%A0%EF%BC%88PS