SKRIPSI

IMPLEMENTASI DEEP LEARNING (CONVOLUTIONAL NEURAL NETWORK) UNTUK MENDETEKSI PEMAKAIAN MASKER SECARA REAL-TIME VIDEO STREAM

Disusun dan diajukan oleh

RAFLY AHMAD MUBIN

H071171304

PROGRAM STUDI SISTEM INFORMASI

DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

2021

IMPLEMENTASI DEEP LEARNING (CONVOLUTIONAL NEURAL NETWORK) UNTUK MENDETEKSI PEMAKAIAN MASKER SECARA REAL-TIME VIDEO STREAM

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin

RAFLY AHMAD MUBIN

H071171304

PROGRAM STUDI SISTEM INFORMASI DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

2021

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama	: Rafly Ahmad Mubin
NIM	: H071171304
Program Studi	: Sistem Informasi
Jenjang	: S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

Implementasi Deep Learning (Convolutional Neural Network) Untuk Mendeteksi Pemakaian Masker Secara Real-Time Video Stream

adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan belum pernah dipublikasikan dalam bentuk apapun.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 01 Oktober 2021

ng menyatakan,

Rafly Ahmad Mubin
NIM. H071171304

IMPLEMENTASI DEEP LEARNING (CONVOLUTIONAL NEURAL NETWORK) UNTUK MENDETEKSI PEMAKAIAN MASKER SECARA REAL-TIME VIDEO STREAM

Disusun dan diajukan oleh:

RAFLY AHMAD MUBIN

H071171304

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin pada tanggal 24 September 2021 dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui,

Pembimbing Utama	**Pembimbing Pertama**
Dr. Hendra, S.Si., M.Kom.	**Supri Bin Hj Amir, S.Si., M.Eng.**
NIP. 19760102 200212 1 001	NIP. 19880504 201903 1 012

Ketua Program Studi,

Dr. Muhammad Hasbi, M.Sc
NIP.196307201989031003

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama	: Rafly Ahmad Mubin
NIM	: H071171304
Program Studi	: Sistem Informasi
Judul Skripsi	: Implementasi Deep Learning (Convolutional Neural Network) Untuk Mendeteksi Pemakaian Masker Secara Real-Time Video Stream

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

		Tanda tangan
Ketua	: Dr. Hendra, S.Si., M.Kom.	(……)
Sekretaris	: Supri Bin Hj. Amir, S.Si., M.Eng.	(……)
Anggota	: Dr. Muhammad Hasbi, M.Sc.	(……)
Anggota	: A. Muh. Amil Siddik, S.Si., M.Si.	(……)

Ditetapkan di : Makassar

Tanggal : 24 September 2021

KATA PENGANTAR

Alhamdulillahi Robbil 'alamin, segala puji syukur dipanjatkan atas kehadirat Allah Subbhanahu Wa Ta'ala, atas segala karunia dan rahmat-Nya sehingga penulis dapat menyelesaikan pendidikan jenjang Strata 1 pada Program Studi Sistem Informasi, Universitas Hasanuddin. dengan judul "**Implementasi Deep Learning (Convolutional Neural Network) Untuk Mendeteksi Pemakaian Masker Secara Real-Time Video Stream**". Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom.) pada Program Studi Sistem Informasi Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

Pada kesempatan kali ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada kedua orang tua tercinta, Ayahanda **Ir. Irfan Ambas, Msc., ph.D.,** dan Ibunda **Andi Eviwati Budiman, SE.,** yang telah mendidik penulis dengan penuh kesabaran, cinta dan kasih sayang yang tidak pernah putus, terima kasih atas segala dukungan , nasihat, serta doa yang tidak henti-hentinya diberikan kepada penulis selama menjalani proses Pendidikan. Untuk saudara penulis **Siti Hutami Nurfatimah** dan **Siti Aisyah Nurfatimah** serta keluarga yang senantiasa memberikan dukungan dan doa. Terima kasih atas segala perhatian yang diberikan kepada penulis. Pada kesempatan ini penulis juga ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Rektor Universitas Hasanuddin, Ibu **Prof. Dr. Dwia Aries Tina Pulubuhu, M.A.** beserta jajarannya; Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, Bapak **Dr. Eng. Amiruddin, S.Si.** beserta jajarannya; Ketua Departemen Matematika, Bapak **Prof. Dr. Nurdin, S.Si., M.Si.** beserta jajarannya; Ketua Program Studi Sistem Informasi, Bapak **Dr. Muhammad Hasbi, M.Sc.** beserta jajarannya; serta Bapak/Ibu dosen Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Hasanuddin atas ilmu-ilmu dan bantuannya dalam berbagai bentuk.
2. Bapak **Dr. Hendra, S.Si., M.Kom.** sebagai pembimbing utama dan Bapak **Supri Bin Hj Amir, S.Si., M.Eng.** sebagai pembimbing pertama sekaligus pendamping akademik atas kesediaan, kesabaran, dan waktu yang telah

diluangkan untuk membimbing penulis selama proses penyusunan tugas akhir; Bapak **Dr. Muhammad Hasbi, M.Sc.** dan Bapak **A. Muh. Amil Siddik, S.Si., M.Si.** atas waktu dan kesediaannya sebagai penguji untuk tugas akhir. Terima kasih pula untuk ilmu-ilmu yang telah diberikan selama proses perkuliahan.

3. Kepada **Azzahra Mubarika** yang selama belakangan ini selalu menemani, membimbing, membantu dan memberikan dukungan kepada penulis dalam melaksanakan kuliah, tugas kuliah dan skripsi ini. Meskipun kamu telah melakukan banyak hal luar biasa bagi saya, saya ingin mengucapkan terima kasih hanya untuk satu di antaranya: atas kehadiranmu dalam hidupku.
4. Kepada soulmate penulis **Muh.Ramadhani Pratama S.Limpo** dan **Muhammad Amirullah Zulkiflie** terima kasih atas segala perhatiannya dan dukungannya kepada penulis selama ini. Tanpa inspirasi, dorongan, dan dukungan yang telah kalian berikan kepada saya, saya mungkin bukan apa-apa saat ini.
5. Kepada teman – teman grup **BRENG**, **KKN** dan **CHARM BERSAYAP** yang selalu membuat saya tertawa dan menyemangati ketika saya sedang jatuh. Terima kasih karena telah menjadi seseorang yang berharga dalam hidupku.
6. Seluruh **Angkatan 2017 SISTEM INFORMASI** terima kasih sudah hadir sejak maba sampai sekarang yang telah banyak membantu mengerjakan tugas, mengajari materi perkuliahan serta menghibur pada masa perkuliahan.
7. Saudara-saudari **MIPA 2017** tanpa terkecuali.
8. Kepada semua pihak yang tidak dapat disebutkan satu persatu, atas segala bentuk kontribusi, partisipasi, dan motivasi yang diberikan kepada penulis selama ini. Semoga yang telah kalian berikan dilipatgandakan oleh Allah SWT.

Makassar, 24 September 2021
Yang menyatakan,

Rafly Ahmad Mubin
NIM. H071171304

PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama	: Rafly Ahmad Mubin
NIM	: H071171304
Program Studi	: Sistem Informasi
Departemen	: Matematika
Fakultas	: Matematika dan Ilmu Pengetahuan Alam
Jenis karya	: Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

Implementasi Deep Learning (Convolutional Neural Network) Untuk Mendeteksi Pemakaian Masker Secara Real-Time Video Stream

beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada tanggal 24 September 2021

Yang menyatakan

(Rafly Ahmad Mubin)

ABSTRAK

Akhir 2019 dunia menyaksikan wabah COVID-19 (Coronavirus Disease-19) yang dianggap sebagai ancaman kesehatan masyarakat terbesar saat ini yang dapat menyebabkan gangguan sistem pernapasan, mulai dari gejala yang ringan seperti flu, hingga infeksi paru-paru, seperti pneumonia. salah satu solusi alternatif yang diberikan oleh pemerintah untuk menghentikan penyebaran virus COVID-19 yaitu menerapkan "New Normal", dimana setiap orang dapat kembali melakukan aktifitas sehari-hari dengan wajib menggunakan masker. Walaupun pemerintah telah memberikan aturan mengenai penggunaan masker, akan sangat sulit untuk memastikan orang-orang mengikuti aturan penggunaan masker yang sangat krusial ini. Oleh sebab itu, dibutuhkannya sebuah sistem otomatis yang dapat mendeteksi aturan penggunaan masker. Dengan berkembangnya teknologi, Computer Vision menyediakan alternatif yang dapat membantu melakukan deteksi manusia baik dalam keadaan bergerak maupun dalam keadaan diam, dengan menggabungkan kombinasi klasifikasi citra, objek deteksi, dan analisis video menggunakan metode Convolutional Neural Network (CNN) yang merupakan salah satu metode yang terdapat dalam Deep Learning yang banyak digunakan untuk menyelesaikan permasalahan berkaitan dengan object detection dan image classification karena memiliki tingkat akurasi yang relative tinggi dan memiliki hasil yang signifikan dalam pengenalan citra. Oleh karna itu hasil penelitian mendeteksi seseorang memakai masker atau tidak memakai masker menggunakan Convolutional Neural Network (CNN) dengan arsitektur MobileNetV2 memperoleh accuracy sebesar 99,61%. Dan untuk kedua perangkat yang digunakan yaitu kamera laptop dan kamera handphone memperoleh hasil yang sama dalam mendeteksi seseorang memakai masker atau tidak memakai masker dengan jarak yang berbeda – beda.

Kata Kunci : COVID-19, Masker, Convolutional Neural Network, MobileNetV2.

ABSTRACT

At the end of 2019, the world witnessed an outbreak of COVID-19 (Coronavirus Disease-19) which is considered the biggest public health threat today that can cause respiratory system disorders, ranging from mild symptoms such as flu, to lung infections, such as pneumonia. One of the alternative solutions provided by the government to stop the spread of the COVID-19 virus is implementing the "New Normal", where everyone can return to carrying out their daily activities by being obliged to wear a face mask. Even though the government has provided rules regarding the use of face masks, it will be very difficult to ensure that people follow this very crucial rule. Therefore, we need an automatic system that can detect a person wearing a face masks. With the development of technology, Computer Vision provides an alternative that can help detect humans both in motion and at rest, by combining a combination of image classification, object detection, and video analysis using the Convolutional Neural Network (CNN) method which is one of the existing methods in Deep Learning which is widely used to solve problems related to object detection and image classification because it has a relatively high level of accuracy and has significant results in image recognition. Therefore, the results of the research detecting someone wearing a mask or not wearing a mask using the Convolutional Neural Network (CNN) with the MobileNetV2 architecture obtained an accuracy of 99.61%. And for the two devices used, which is laptop camera and smartphone camera, obtained the same results in detecting someone wearing a mask or not wearing a mask at various distances.

Keywords : COVID-19, Face Mask, Convolutional Neural Network, MobileNetV2.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I
INTRODUCTION

1.1. Background

At the end of 2019, the world witnessed the outbreak of COVID-19 (Coronavirus Disease-19) which caused the suffering of millions of lives, businesses and everyday life, until the time this paper is created. COVID-19 (Coronavirus Disease-19) is a type of virus that is considered to be the biggest public health threat today. COVID-19 was first discovered in December 2019 in Wuhan, China (Huang et al, 2020). The COVID-19 virus can cause respiratory system disorders, ranging from mild symptoms such as flu, to lung infections, such as pneumonia (Liu et al, 2020). The virus is mostly spread to people who have close contact with an infected person through sneeze and cough. Based on current events, the COVID-19 virus has spread to 220 countries with a total of 179 million cases and more than 3 million people have lost their lives from it. This virus has become a threat to the whole world because it can disrupt and even hinder various sectors in daily life. Therefore, health organizations, medical experts and researchers around the world are looking for solutions to prevent the spread of the COVID-19 virus by trying to develop a vaccine for the virus.

At a time when the world is struggling against COVID-19, one of the alternative solutions provided by the government to stop the spread of the COVID-19 virus is to implement the "New Normal", where everyone can return to their daily activities without having to worry about being exposed to the virus by wearing a face mask as well as to maintain physical distancing. However, some people are having difficulty to adapt the new normal due the appearance of anxieties among individuals, especially for those who wants to continue their personal activities. According to Woods, about 65% of employees are anxious about returning to the office (Woods, 2020). Various studies have shown that the use of face masks reduce the risk of spreading the virus as well as provide a sense of protection (Verma et al, 2020).

A new study led by Christopher Leffler, M.D. from the VCU, which is currently in the process of peer review, pointed out when the face mask

requirements given by the government enacted a major factor. The study found out that the single most important factor is wearing a face mask in public. Figure 1.1 shows the COVID-19 death rate based on the country's policy of wearing a face mask.

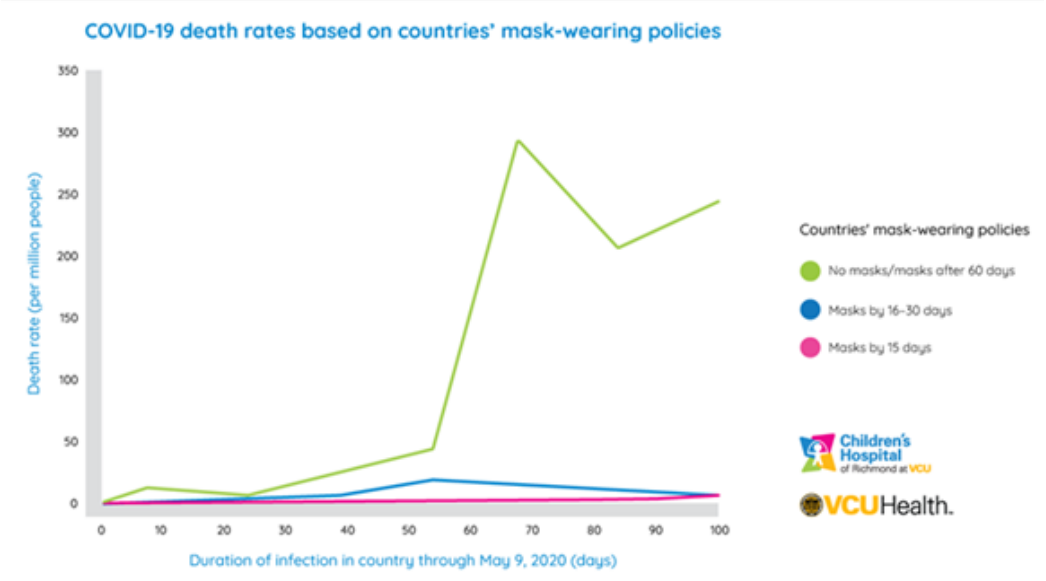

Figure 1.1 COVID-19 Death Rate Based on The Country's Policy

It can be seen in figure 1.1, countries' death rates per 1 million people after approximately three months varied based on how quickly that country adopted a policy of wearing a face mask. For countries that recommended to wear face masks within 15 and 30 days, the death rate from COVID-19 is far lower than for countries that waited longer (Marino, 2020).

For that reason, face masks have become commonly used today to prevent the spread of COVID-19 virus outbreak during 2020. Although the government has given the rules regarding the use of face masks, it will be very difficult to ensure that people follow these very crucial rules. Therefore, an automatic system that can detect people for using face masks is needed.

With the development of technology, Computer Vision provides an alternative that can help detect object both in motion and at rest, by combining image classification, object detection, and video analysis. The method to detect and recognize objects in an image, is a method of Convolutional Neural Network

(CNN). Research conducted by Imam Taufiq (2018) using the Convolutional Neural Network (CNN) method in detecting motor vehicle number plates resulted in 99% accuracy. Convolutional Neural Network (CNN) is one of the methods found in deep learning widely used to solve problems related to object detection and image classification. Convolutional Neural Network (CNN) is widely used in preliminary studies because it has a relatively high level of accuracy and has significant results in image recognition.

Based on the problems that have been described, the author decided to conduct a research to develop a system that can detect a person wearing a face mask or not wearing a face mask in the form of video stream using the Convolutional Neural Network method entitled "Implementation of Deep Learning (Convolutional Neural Network) to Detect Mask Usage Real-Time Video Stream."

1.2. Research Problem

Based on the background that has been explained, the formulation of the problems to be examined in this research are:

1. How is the implementation of Deep learning in detecting a person wearing a face mask or not wearing a face mask using the Convolutional Neural Network method?
2. What are the results of the performance analysis of the Convolutional Neural Network method using MobileNetV2 architecture in detecting a person wearing a face mask or not wearing a face mask?
3. What is the outcome of the system results in detecting someone wearing a mask or not wearing a mask using two different types of cameras?

1.3. Research Objectives

Based on the formulation of the problem that has been described, the purpose of this research are:

1. To find out how the implementation of Deep learning in detecting a person wearing a face mask or not wearing a face mask using the Convolutional Neural Network method?

2. To find out the results of the performance analysis of the Convolutional Neural Network method using MobileNetV2 architecture in detecting a person wearing a face mask or not wearing a face mask?
3. To find out the result of the system in detecting someone wearing a mask or not wearing a mask using two different types of cameras?

1.4. Benefits of Research

This research is expected to provide the following benefits:

1. For researchers, this research is expected to provide insight and develop capabilities in the field of research.
2. For educational institutions, it can be used as a reference in the development of research on related topics.
3. For governments and organizations that deal with the public, they can have a tool that automatically detects violations of not wearing a face mask to help reduce the spread of COVID-19.

1.5. Limitation of Research

The limitations of the problem in this research are:

1. The method used in this research is Convolutional Neural Network.
2. The Convolutional Neural Network architecture used in this research is the MobileNetV2 architecture which will be trained, analyzed and used for the face mask classifier in the system and a pretrained SSD architecture which only be used for the face detection in the system.
3. The MobileNetV2 architecture in this research trained using the facemask-dataset, that has a total of 3860 images and two classes, namely wearing a face mask totaling 1930 images and not wearing a face mask totaling 1930 images.
4. The device used in this research are webcam and smartphone camera.
5. Webcam resolution is 0.3 MP HD webcam.
6. Smartphone camera resolution is 12 MP, f/2.2, 23mm (wide), 1/3.6" SL 3D, (depth/biometrics sensor).

CHAPTER II
LITERATURE REVIEW

2.1. Preliminary Study

The use of preliminary study is vital as a reference to determine the relationship with this research, as well as to avoid plagiarism and duplication of other studies and can be used as a reference in future research.

The system development that uses Computer Vision has become very effective functionality to facilitate work in various fields, such as face detection, image recognition and pattern recognition. The preliminary study on Convolutional Neural Network has been done in a wide variety of objects. One of them is a research conducted by Ardian Yusuf Wicaksono, et al in 2017, regarding Modified Convolutional Neural Network architecture for batik motif image classification. This study used the Convolutional Neural Network method by developing the architecture and combining GoogleNet and Residual Networks or IncRes. This study used a different type of batik motif that has 11 classes with a total of 7112 image data which is divided into 6401 for training data and 711 for testing data. The results of this study obtained an accuracy of 70.84% with a time of 733 ms (milliseconds) (Wicaksono et al, 2017).

Research on "Deep Learning to Detect Motor Vehicle Number Signs Using Convolutional Neural Network Algorithms with Python and TensorFlow" is conducted by Imam Taufiq in 2018. In this study, the Convolutional Neural Network algorithm is used to classify and detect motorized vehicle plates in an image. In this study, there are 502 image datasets and used a ratio of 80% for training and 20% for testing. The training process requires more than 25,000 steps with a batch of 8 and when the batch used is 4, it requires 100,000 steps until the model being trained is able to detect the presence of TNKB and produce an accuracy of about 99% on an image of a motor vehicle plate (Taufiq, 2018).

Implementation of Convolutional Neural Network can also be developed in terms of architecture. In 2012, Alex Krizhevsky implemented the Convolutional Neural Network method using the AlexNet architecture tested with the ImageNet

dataset. This study used the ImageNet LSVRC-2010 dataset into 1000 classes. The architecture made in this study showed very significant results in the testing test with a test error of 17%. These results can be considered very good because the images used in the dataset are very large (Krizhevsky et al, 2012).

In 2016, Muhammad Zufar and Budi Setiyono conducted a research using the Convolutional Neural Network method implemented for real-time face recognition. The dataset used in this study is a face image consisting of two classes, namely an indoor face class (dark lighting conditions) and an outdoor face class (bright lighting conditions). The results of face recognition accuracy are 89% in 2 frames per second using the Convolutional Neural Network model construction to a depth of 7 layers with the results of the extraction of Extended Local Binary Pattern and radius 1 and neighbor 15. This study showed that the implementation of the Convolutional Neural Network model can be applied to the face recognition process automatically in real-time with high enough accuracy (Zufar, 2016).

The application of this Convolutional Neural Network method can also be implemented in the traffic sign recognition. As in the research conducted by S. Visalini in 2017 regarding the traffic sign recognition using the Convolutional Neural Network. The dataset used in this study is taken directly by geolocation using an android application. This study does not mention the number of datasets used, but the results from the level of accuracy given using Convolutional Neural Network to detect or recognize traffic signs are 85% - 90% (Visalini, 2017).

Based on preliminary studies that have been mentioned, the author utilizes these studies as a reference of ideas for this study. As a result, the emphasis of this research is on how the Convolutional Neural Network algorithm can detect and categorize a person wearing or not wearing a face mask.

2.2. Covid-19

Coronaviruses are a wide group of viruses that infect both people and animals. It causes respiratory tract infections in humans, ranging from the common cold to catastrophic illnesses like Middle East Respiratory Syndrome (MERS) and Severe Acute Respiratory Syndrome (SARS). A new type of coronavirus found in humans since an extraordinary event appeared in Wuhan City, Hubei Province, China in

December 2019, and was designated a pandemic by the World Health Organization (WHO) on March 11, 2020, later named as Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-COV2), and causes the disease of Coronavirus Disease-2019 (COVID-19). As of April 23, 2020, more than 2,000,000 cases of COVID-19 have been reported in more than 210 countries and regions such as Taiwan, Thailand, Vietnam, Malaysia, Nepal, Sri Lanka, Cambodia, Japan, Singapore, Saudi Arabia, South Korea, Philippines, India, Australia, Canada, Finland, France and Germany.

The first COVID-19 was reported in Indonesia on March 2, 2020 with two cases. Data on March 31, 2020 showed that there were 1,528 confirmed cases and 136 deaths. The COVID-19 mortality rate in Indonesia is 8.9%, this figure is the highest in Southeast Asia. As of March 30, 2020, there were 693,224 cases and 33,106 deaths worldwide. Europe and North America have become the pandemic center of the COVID-19, with cases and deaths already surpassing China. The United States ranks first with the most COVID-19 cases with the addition of 19,332 new cases on March 30, 2020, followed by Spain with 6,549 new cases. Italy has the highest mortality rate in the world at 11.3%.5, resulting in more than 195,755 deaths and more than 781,109 recoveries. As for the latest data as of August 18, 2020, there are 22,034,440 COVID-19 cases in the world, where America is still ranked first with 5,620,361 cases and Indonesia, which is 143,043 cases (Maulana, 2020).

2.3. Face Mask

Face masks, such as surgical masks and fabric masks, are used as a public and personal health control tool against the transmission of SARS-CoV-2 during the COVID-19 pandemic. Their usage is intended as both source control and personal protection in both community and healthcare settings to restrict viral transmission and prevent infection. Their function for source control is emphasized in community settings.

Health professionals and political leaders have advised wearing a face mask (or covers in some circumstances) to decrease the risk of infection (Woodruff et al, 2020). About 95% of the world's population were wearing a mask in public during a pandemic is encouraged or required.

In the case of the COVID-19 pandemic, governments advocate the use of face masks for the general public, primarily to prevent the spread of infection from infected people to others. Exhalation valve masks are not advised since they discharge the user's breath outwardly, and an infected wearer might spread viruses through the valve. A second function of face masks is to protect each user from contaminated surroundings, which may be accomplished by a variety of mask types.

Between the different types of face masks that have been recommended throughout the COVID-19 pandemic, with higher or lower effectivity, it is possible to include (Sanger, 2021):

a) Cloth face masks.
b) Surgical masks (medical masks).
c) Uncertified face-covering dust masks.
d) Certified face-covering masks, considered respirators, with certifications such as N95 and N99, and FFP.
e) Filtering respirators with certifications such as N95 and N99, and FFP.
f) Other respirators, including elastomeric respirators, some of which may also be considered filtering masks.

2.4. Digital Image

Digital image are images that can be processed by a computer (Sutoyo, 2019). A digital image can be represented by a matrix consisting of M columns and N rows where the intersections of the columns and rows are called pixels. Pixel is the smallest element of an image. Pixels have two parameters, namely coordinates and intensity or color.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,m-1) \\ f(1,0) & f(1,1) & & f(1,m-1) \\ & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1,M-1) \end{bmatrix} \quad (2.1)$$

Based on the matrix form as shown above, it can be seen that the f(x,y) pixel matrix has two parameters, namely coordinates and intensity or color. Systematically, a digital image can be written as a function of intensity f(x,y) where x (row) and y (column) are position coordinates and f(x,y) are the function value at

each point (x,y) which represents the magnitude the intensity of the image or the level of gray or color of the pixels at that point (Nalwan, 1997).

Several types of digital images that are often used are binary images, grayscale images, and color images (Sutoyo, 2009). Binary image is a digital image that has 2 colors, namely black and white. The black color is worth 1 and the white color is 0, the example of binary image is shown in figure 2.1 below.

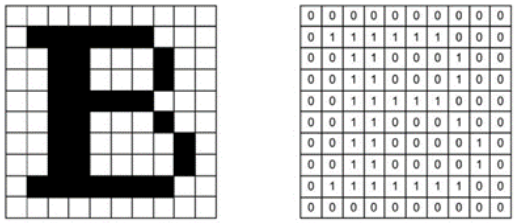

Figure 2.1 Example of Binary Image

In figure 2.1, is a binary image in the form of the letter B. This image representation has values 0 and 1. The black color in the image representation in the form of the letter B is worth 1, and the background of the letter B is 0.

A grayscale image is an image that has only one channel value in its pixel. In other words, the values of the red, green, and blue parts have the same color, namely black, gray, and white. This value is used to indicate the level of intensity. The gray level here is the color of gray with various levels from black, to close to white. Grayscale images have the possibility of black for the minimum value and white for the maximum value. Figure 2.2 below is an example of grayscale image.

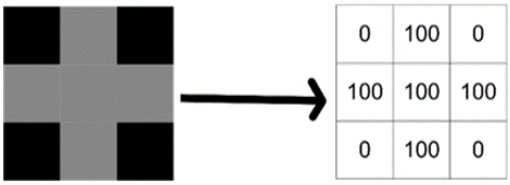

Figure 2.2 Examples of Grayscale Image

Figure 2.2 shows an image that has 2 colors, black and gray. The black color in Figure 2.2 is represented by a value of 0, while the gray color is represented by a value of 100.

In a color image, the pixel value representation does not only have one value, but three values. Each pixel represents a specific color which is a combination of three basic colors, namely red, green, and blue. This image format is often referred

to as an RGB (Red-Green-Blue) image. Each base color has its own intensity with a maximum value of 255, and the minimum color is white. Red has a minimum color of white and a maximum color of red, Green has a minimum color of white and a maximum of green, and Blue has a minimum color of white and a maximum of blue. Figure 2.3 below is an example of color image.

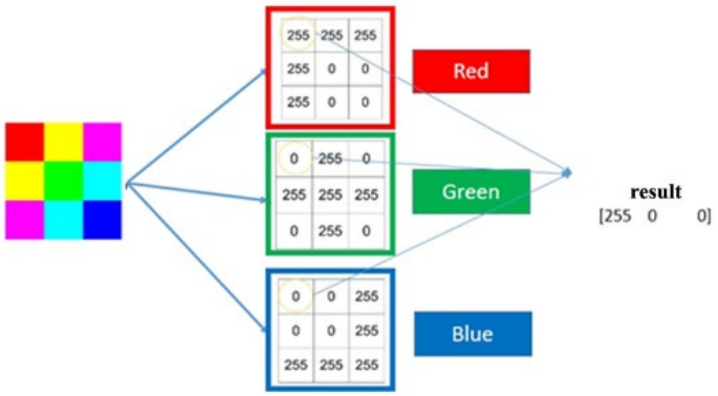

Figure 2.3 Example of Color Image

In figure 2.3, shown a color image with its representation. As explained above, in the color image representation, each pixel represents the basic colors, namely red, green, and blue so that the image representation in the row 1 column 1 pixels has a representation value of 255, 0, 0.

2.5. Digital Videos

Digital video is basically composed of a series of frames. The series of frames is displayed on the screen at a certain rate, depending on the given frame rate (in frames / second). If the frame rate is high enough, the human eye cannot capture images frame by frame, but rather captures them as a continuous series. Each frame is a digital image. Digital video is composed of several structures, namely (Bovik, 2005):

a) Video Sequence, starting with a sequence header, containing one or more groups of images, ending with an end-of sequence code.

b) Group of Pictures (GOP), Header and a series of one or more pictures intended to allow random access to be sequential.

c) Picture, the main coding unit of the video sequence and consists of three rectangular matrices representing the exposure value (Y) and two chromium values (Cb and Cr).
d) Slice, one or more contiguous macroblocks. The order of macroblocks in a slice is from left to right and top to bottom.
e) Macroblocks, lighting components and chromium components according to the order of blocks in the data stream.

The quality of a digital video is determined by the magnitude of the characteristic values described as follows (Manning, 1996):

a) Frame rate, is the number of frames displayed per second. The illusion of movement can be felt at a frame rate of 12 frames per second. The modern film industry uses 24 frames per second.
b) Frame dimensions, is the width and height of the video frame expressed in pixel size.
c) Pixel depth, is the number of bits per pixel. In some cases, it is possible to separate the bits dedicated to luminance from those used for chrominance. In other cases, all bits can be used as a reference to one of the various color palettes.

2.6. Video Streaming

Video streaming is an interesting field to explore because it is relatively new and relatively inexpensive with the cheapening of electronic equipment. One of the applications of video streaming is to monitor room conditions, video information will be sent through communication channels, including networks, telephone cables, ISDN channels or radio. Video information has a wide bandwidth (very many bytes per second transmitted), which is why video compression technology is very much needed to reduce bandwidth requirements before being sent over the communication channel. The only equipment that needs to be added is a simple video camera. Just a quick overview, a good uncompressed video channel will take up about 9 Mbps of bandwidth. With compression techniques that already exist today, we can save a video channel of about 30 Kbps. That means a not-so-fast

internet channel can actually be used to stream videos. Some things to consider when sending videos are (Angki & Subhan 2011):

a) If we use black and white video it will take up less bandwidth than if we do a conference using color video.
b) If we use a low frame per second (fps) video transmission speed, it will take up less bandwidth than a high frame per second (fps). A fairly good video is usually sent at a frame per second (fps) rate of around 30 fps. If sent uncompressed, a video at 30 fps will take up about 9Mbps of bandwidth, which is very large for the size of a data communication channel.

2.7. Resolution

Resolution or frame dimensions are the size of a frame in digital video. Resolution is expressed in pixels x pixels. The higher the resolution, the better the quality of the video, in the sense that in the same physical size, the higher resolution video will be more detailed. However, a high resolution will result in an increase in the number of bits required to store or transmit it. For example, in HD (High - Definition) format, the resolution is 1280 pixels x 720 pixels (Rossi, 2016).

2.8. Data Augmentation

Data augmentation is a method in image data processing that may change or modify pictures in such manner that the computer detects the modified image as a new image but people can still tell it is the same image (Mahmud et al, 2019). Cropping, padding, and horizontal flipping are popular in data augmentation techniques used to train massive neural networks. However, the augmentation type is employed in the majority of neural network training techniques (Sanjaya & Ayub, 2020).

2.9. Machine Learning

Arthur Samuel was the first to define Machine Learning in 1959. According to Arthur Samuel Machine Learning is a field of computer science that provides computer learning capabilities to find out or learn something from data. To achieve the best results, Machine Learning knowledge might use the same data but various methods and techniques. Machine Learning knowledge can involve the same data but different algorithms and approaches to get optimal results (Samuel, 1959).

2.10. Deep Learning

Deep Learning is part and development of Machine Learning. Deep Learning consists of several hidden layers which are also part of artificial intelligence. Deep Learning methodologies apply non-linear transformations and high-level model abstractions in large databases. The rapid development of Deep Learning architecture in various fields has made a significant contribution to artificial intelligence. This indicated by many algorithms in Deep Learning that are used in various applications. The layers in Deep Learning are shown in figure 2.4 (Vargas & Lourdes, 2017).

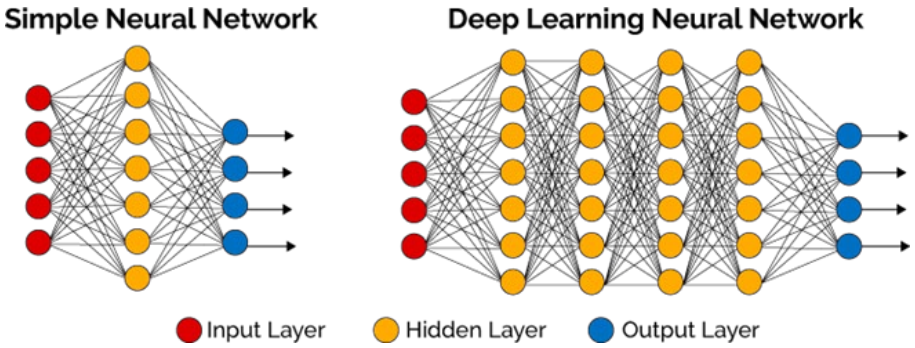

Figure 2.4 Deep Learning Layers

In figure 2.4 Shows a comparison of a simple and Deep Neural Network, highlighting the different number of hidden layers present in each process. The concept of Deep Learning provides many advantages and is growing in recent years. Deep learning is widely used in various real-life applications and can be done automatically by computers such as, in digital image processing, grayscale image coloring, biometrics and so on. Libraries in the python programming language that can be used for Deep Learning applications are the dlib library (Deep Learning library), TensorFlow, Keras, Theano, pytorch and others (Vargas & Lourdes, 2017).

Deep Learning also has many types of models, where each type has a different algorithm and function. The types of Deep Learning models and their uses are as follows (Herlambang, 2019):

1. Artificial Neural Network (ANN) : For regression (techniques to find out future data or predict future data) and classification (binary data predictions, categories and others).

2. Convolutional Neural Network (CNN) : Can be used for computer vision applications, face recognition, object detection, image recognition, visual recognition where it learns how machines (with algorithms) can recognize objects in the form of images or videos.
3. Recurrent Neural Network (RNN) : Can be applied for voice recognition (voice / speech recognition), continuous data analysis. The RNN process is carried out repeatedly by storing the results of calculations in the previous time step to be used in the next time step so that it can be considered as a collection of neural network layers.
4. Recursive Neural Tensor Network (RNTN): For text processing such as sentiment analysis, parsing, and entity recognition
5. Restricted Boltzmann Machine (RBM): To extract unlabeled data using an unsupervised approach and can automatically find patterns in the data by reconstructing the input.
6. Deep Belief Network (DBN): To study the feature set of high-level and more complex data gradually from the data distribution. DBN has a hidden layer where each layer is connected to each other but the units are not connected. DBN also represents a hierarchy of outputs from RBM.
7. Autoencoders: Used to reduce feature dimensions (dimensionality reduction). The autoencoder accepts unlabeled data for coding and then reconstructs the data as accurately as possible.

It can be observed that the Convolutional Neural Network technique is the most ideal deep learning model for creating a face mask detection system since it is most suitable for image identification and object detection according to its function.

2.11. Artificial Neural Network

Artificial Neural Network (ANN) or Multilayer Perceptron (MLP) is a data processing system by imitating the workings of the human nervous system. In 1943 Warren McCulloch and Walter Pitts explained how neural networks work with electronic network devices. Artificial Neural Network is a system consisting of many simple processing elements connected in parallel. Artificial Neural Network consists of a number of connected inputs and outputs, and each connection has its

own weight that can be changed to get the desired prediction results. In Figure 2.5 there are layers in an Artificial Neural Network which are described as follows (McCulloch & Pitts, 1943):

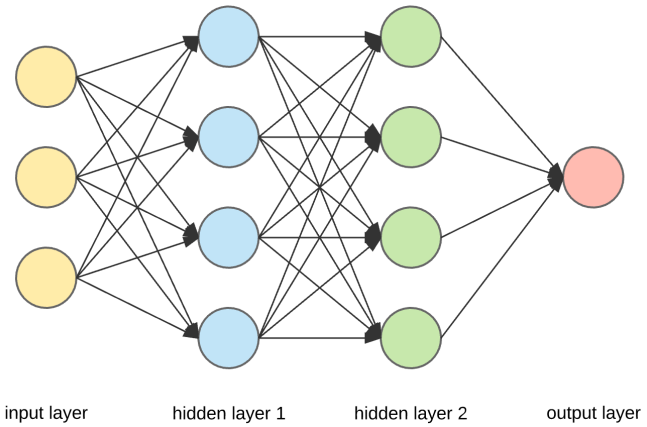

Figure 2.5 Artificial Neural Network

In figure 2.5 the Artificial Neural Network layers described as the following:

a) Input Layer

   is the layer that connects the data source to the processing network. In a sense, each input will represent the independent variables that affect the output.

b) Hidden Layer

   It is a propagation layer of input variables to get output results that are closer to what you want. A Multi-Layer Artificial Neural Network can have one or more hidden layers.

c) Output Layer

   is the output of Artificial Neural Network data processing. The output obtained depends on the weight, the number of hidden layers.

2.12. Convolutional Neural Network (CNN)

Convolutional Neural Network is one type of neural network and the development of Multi-Layer Perceptron (MLP). Convolutional Neural Networks are commonly used in the scope of visual recognition, on how machines can recognize objects in the form of images or videos. The architecture of the Convolutional Neural Network is said to be similar to the pattern of connections of neurons or nerve cells in the human brain. Convolutional Neural Network is

inspired by the Visual Cortex, the part of the brain responsible for processing information in visual form.

Convolutional Neural Network is widely used by researchers in analyzing an object, because this algorithm has been claimed as the best model in solving object recognition problems. It is based on how Deep Learning works, where the features of the dataset can be extracted automatically. This is quite different from the Machine Learning method in general where to obtain a feature from a dataset, the feature extraction method needs to be applied manually.

Convolutional Neural Network take pixels from image input, then converts them through Convolutional Layer, Rectified Linear Unit (ReLU), and Pooling Layers. At the end, it is combined in a Fully Connected Layer and SoftMax is activated to obtain the probability value of each class, where the input data is classified into the class with the highest probability.

As the name implies, the Convolutional Neural Network utilizes the Convolution process. By moving a Convolution kernel (filter) of a certain size to an image, the computer gets new representative information from the result of multiplying that part of the image with the kernel used. In addition, the following illustration of the Convolution kernel is shown in figure 2.6.

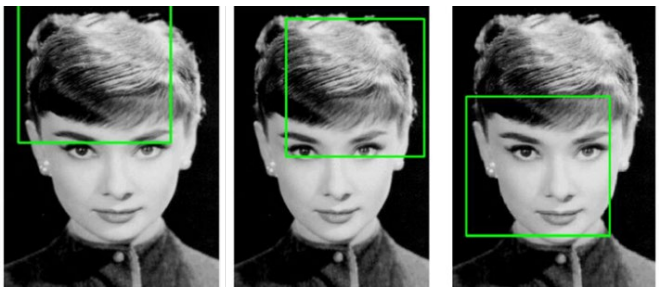

Figure 2.6 Kernel Illustration on an Image

In figure 2.6 is an illustration of a kernel on an image. The kernel will move from the top left, and continue to shift until it reaches the bottom right. The Convolutional Neural Network process will be shown in figure 2.7.

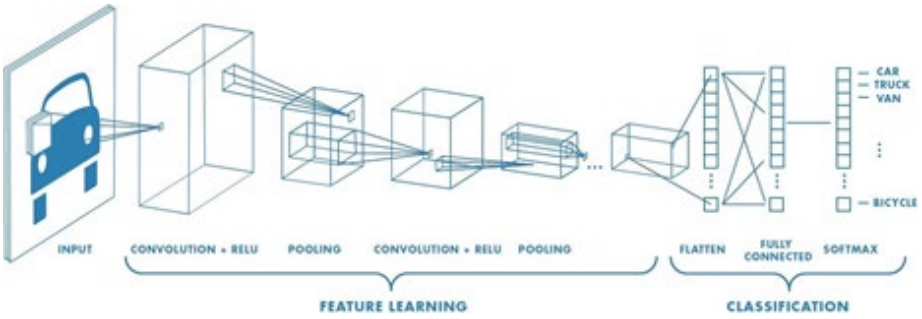

Figure 2.7 Convolutional Neural Network Process

Figure 2.7 shows a Convolutional Neural Network process, where the process that occurs in feature learning is performing the "encoding" of an image into features in the form of numbers that represent the image (feature extraction). Feature learning consists of two parts, namely the Convolutional Layer and the Pooling Layer. Whereas in classification, it is performing the "encoding" on the results of feature extraction on the input to be classified in a class. Classification consists of flatten, fully connected layer, and softmax (Sena, 2017).

2.12.1 Convolutional Layer

The Convolution Layer is a layer for the image manipulation process to produce a new image. Layers on Convolutional Neural Network work hierarchically, meaning that the output from the first layer will be used as input for the next layer. There are many parameters that can be changed to modify the properties of each layer, namely the kernel, stride, padding and others. Kernel is performed to extract objects from the image as input data. The kernel detects characters from the image such as mouth, nose and eyes. Kernel is done repeatedly to produce image data with better quality, eliminating noise and smoothing the image. Stride serves as a kernel controller that is applied to the input data and as a determinant of the number of kernel shifts. For example, if the stride value is 1, then the kernel in the Convolution layer will shift by 1 pixel horizontally and then vertically. The padding parameter is the addition of the pixel size to a certain value around the input data, this is done because the process in the kernel reduces the size of the original image matrix so that padding is needed to maintain the actual matrix size. The following is a discussion of the Convolution layer process using a RGB image shown in figure 2.8 (Herlambang, 2019).

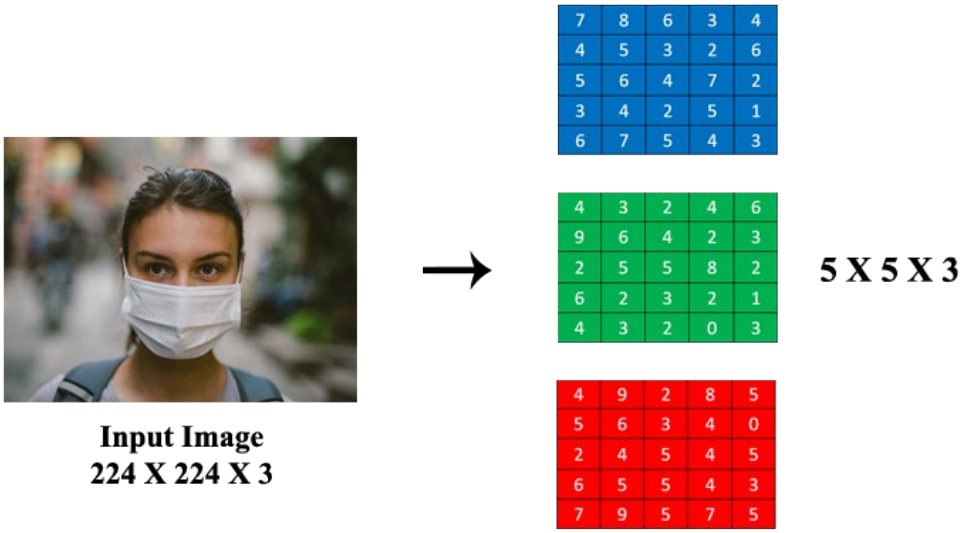

Figure 2.8 Data Image Input

Figure 2.8 above is an example of an input image that represents a dataset. The input image is an RGB image and has a pixel size of 224 x 224. Before entering the convolution stage, the image will be converted into pixel weights or can be said to be a matrix. Each pixel or matrix has a value, this value is referred to as a pixel value. Where the pixel value has a range of 0 – 255. The picture above is an example of an image that has a pixel size of 224 x 224 x 3 but because the calculation is too much, the process of simplifying the pixel/matrix size with a size of 5 x 5 x 3 as an example of the convolution process on the Convolution Layer is carried out, and the calculation could be simpler. Number 3 above means the image has 3 color channels, represents red, green, blue or commonly called RGB.

The element involved in carrying out the convolution operation in the first part of a Convolutional Layer is called the Kernel/Filter. The convolution process makes use of what is referred to as a kernel/filter. Like images, kernel have a certain height, width, and thickness. This kernel is initialized with a certain value and the value of this filter is the parameter that will be updated in the learning process. The convolutional prosses for each channel with the kernel, can be seen in figure 2.9 (Suyanto & Mandala, 2019).

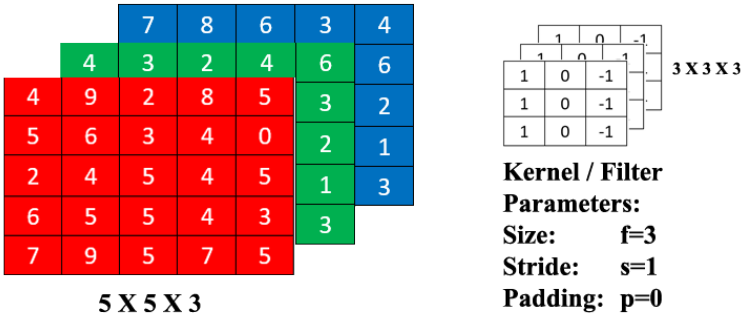

Figure 2.9 Convolution Process Using Kernel

Based on figure 2.9 above the convolution operation uses a 3-dimensional kernel that's going to be 3x3x3 kernel, so the kernel itself will also have three layers corresponding to red, green and blue channels. Each kernel then calculates for each channel red, green and blue. As a result, the calculated channel with the kernel will be combined to produce the feature maps in the Convolution Layer.

The next step is how the convolutional process calculation uses the stride parameter with the kernel. The Convolution process using the stride parameter shown in figure 2.10

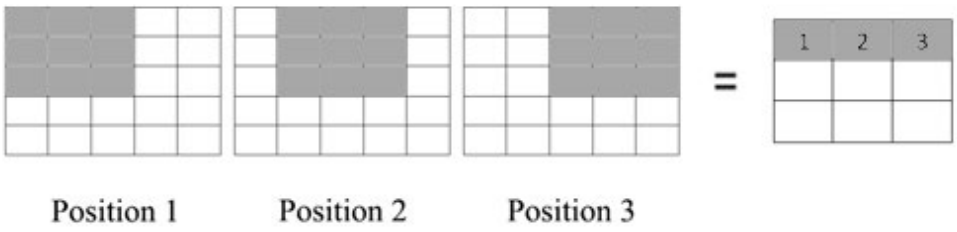

Figure 2.10 Stride Shift by One Pixel

From figure 2.10 above Stride is a parameter that controls the number of kernel shifts. if the stride value is 1, then the convolutional kernel will shift by 1 pixel horizontally and vertically. The smaller the stride, the more detailed the information obtained. After the description above, the next process is to calculate the output for each channel using stride = 1 with the kernel. The result shown in figure 2.11

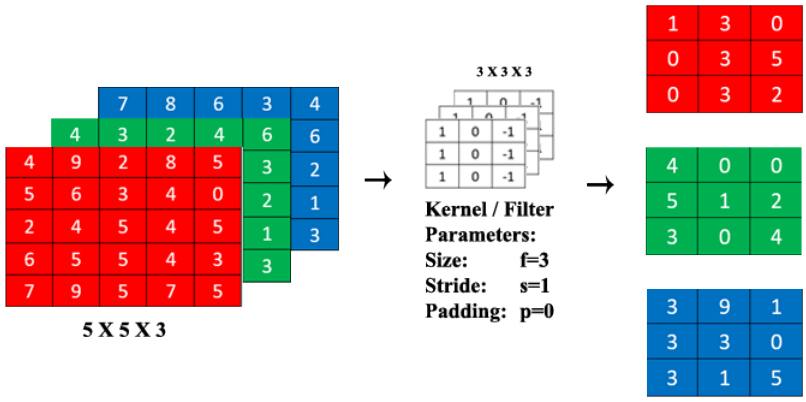

Figure 2.11 Output of Each Channel

Figure 2.11 above is the result of the output for each channel which will be combined to produce a future map. The output from each channel is summed with the bias to give a squashed one-depth channel for Convoluted Feature Map shown in figure 2.12.

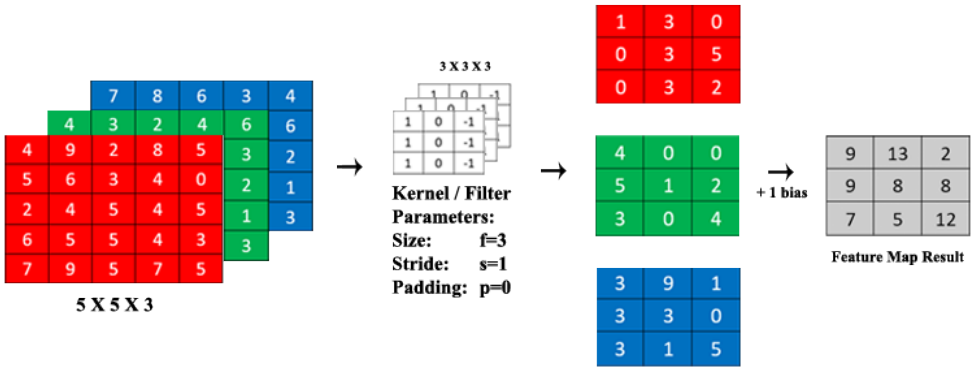

Figure 2.12 Feature Maps Result

It can be seen in figure 2.12 above, the process that the initial 5×5 image is reduced to a smaller 3×3 size. It shows that if you process large and high-resolution images, the number of pixels will reach thousands. The kernel of the image processing becomes faster because the pixel data processed is also getting smaller.

2.12.2 Activation Function

Before entering into the pooling process, it utilizes the activation function before you begin the pooling process. The activation function aims to determine whether a neuron should be active or not based on the weighted sum of the inputs. The following formula of activation functions:

a) Rectification Linear Unit (ReLU):

ReLu is an activation function in Artificial Neural Network that is currently widely used, following the formula for ReLu:

$$f(x) = max(0, x) \tag{2.2}$$

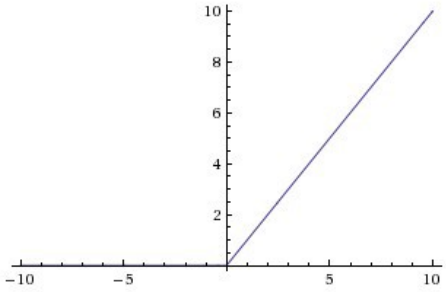

Figure 2.13 ReLU

Figure 2.13 above describe ReLU only creates a limit on the number zero, meaning that if x 0 then x = 0 and if x > 0 then x = x. If the input is greater than 0, the output is the same as the input. ReLu functions are more like neurons like in the human body. ReLu activation is basically the simplest non-linear activation function. If you get a positive input, the derivative is only 1, in other words, the activation threshold is only at zero. Research showed that ReLu results in faster training for large networks.

b) Softmax

Softmax is a very interesting activation function because it not only maps our output to a [0,1] range but also maps each output in such a way that the total sum is 1. The output of Softmax is therefore a probability distribution. (Vedaldi & Lenc, 2015):

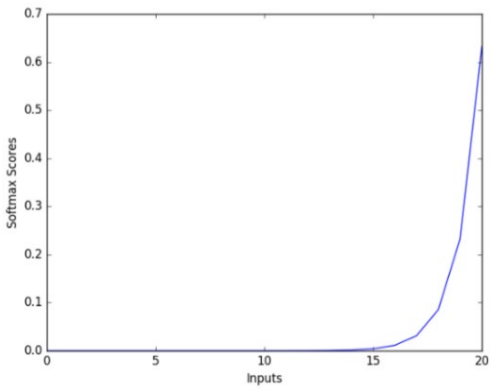

Figure 2.14 Softmax

Mathematically from Figure 2.14 Softmax is the following function where z is vector of inputs to output layer and j indexes the the output units from 1,2, 3 …. k:

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \text{ for } j = 1, \ldots, K \tag{2.3}$$

In conclusion, Softmax is used for multi-classification in logistic regression model (multivariate) whereas Sigmoid is used for binary classification in logistic regression model.

2.12.3 Pooling Layer

Pooling Layer or subsampling is a reduction in the size of the matrix of the Convolution layer. There are two types of pooling that are often used, namely average pooling and max pooling. The value taken in average pooling is the average value while in max pooling is the maximum value. The Pooling layer process is shown in figure 2.15 (Zhi et al, 2016).

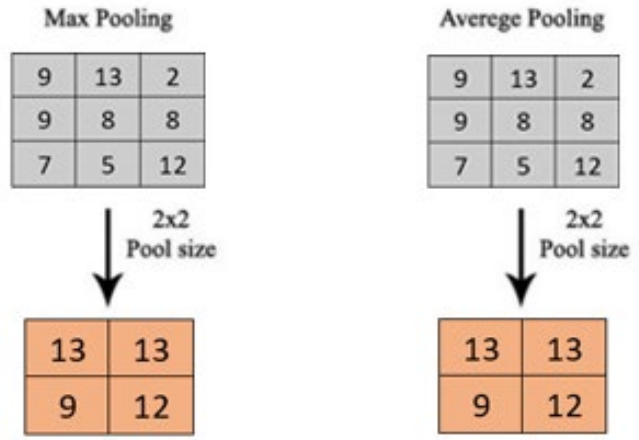

Figure 2.15 Pooling Layer

In Figure 2.15 is the result of pooling layer, on the left is max pooling where the result is the highest value of numbers in each stride, and on the right is average pooling by dividing the output of the Convolutional layer into several grids based on the determination of the number of strides and the type of pooling used.

2.12.4 Flattening

Flatten is a simpler stage when compared to Convolution and pooling layers. The flattening stage works by changing the matrix in the pooling layer to only one

column (a single vector). Where this vector will later become part of the input layer in the fully connected layer or Artificial Neural Network. The flattening process is shown in figure 2.16.

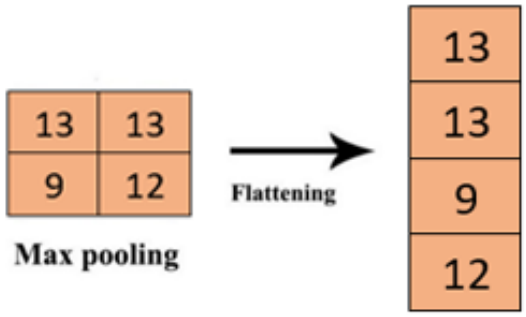

Figure 2.16 Flattening Process

In figure 2.16 flattening is a process of converting the data into a 1-dimensional array for inputting it to the next layer. We flatten the output of the Convolutional layers to create a single long feature vector. After the flattening process is carried out, the results are entered into the input layer of an Artificial Neural Network. All results from the pooling layer (pooled feature maps) will be 1 vector only. So, if a pooling layer is a 2×2 matrix, then it will be 1 vector with 4 rows. If there are 8 pooling layers with the same size (2×2), then there will be 1 vector with 32 lines as input for Artificial Neural Network (Herlambang, 2019).

2.12.5 Fully Connected Layer

In this layer, the results of the flattening process are incorporated into the intact Artificial Neural Networks structure. As usual, Artificial Neural Network consists of 3 parts, namely the input layer, hidden layer, and output layer. Thing to note is in the Deep Learning model of Artificial Neural Network that each node does not have to be connected to the nodes in front of it. In the context of Convolutional Neural Network, all nodes of the Artificial Neural Network must be connected with nodes in front and behind it, therefore it is called Fully connected. An illustration of a fully connected layer is shown in figure 2.17 (Herlambang, 2019).

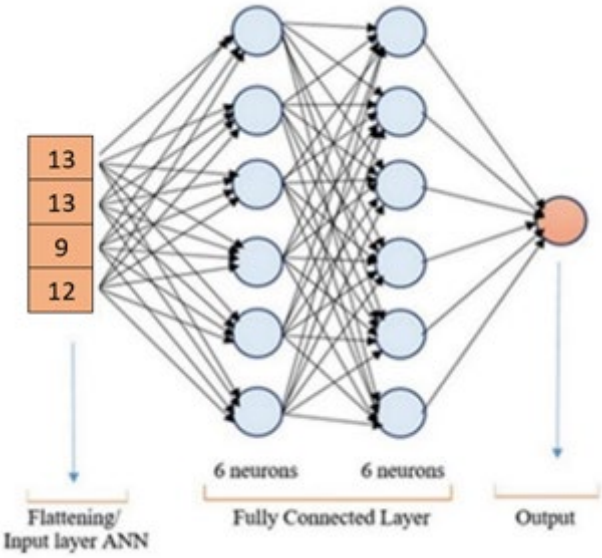

Figure 2.17 Fully Connected Layer

Figure 2.17 is a fully connected layer, where the input value is obtained from the results of flattening. The fully connected layer is also known as the hidden layer, where the number of hidden layers is not limited or free to determine, and the example above has 2 layers of hidden layers. The number of neurons / nodes / perceptions in each layer of the hidden layer is also not limited or free to determine it, but in this case, it must be adjusted to the computer hardware. The larger the number of hidden layers and neurons, the computational process will be longer and heavier.

2.12.6 Dropout Regularization

Dropout is one of the techniques used to avoid overfitting in the model. In this method, the activation of randomly selected neurons in the neural network is taken as zero during training. The selected neurons are changed in each training iteration. The learning process becomes more reliable and overfitting is reduced by this method. The term “Dropout” refers to the disconnection of neurons in a neural network. By removing the neural temporarily removes it from the neural network, along with all of its input and output connections, as shown in figure 2.18 the selection of units dropped randomly (Srivastava et al, 2014).

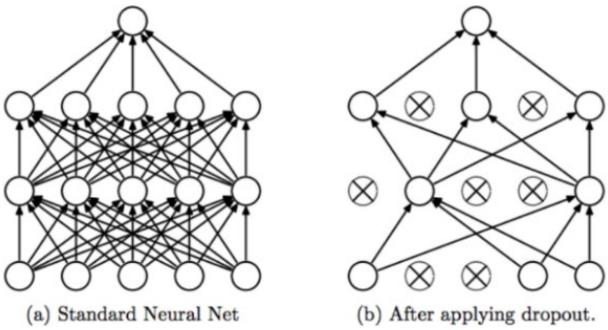

Figure 2.18 Normal Neural Network and After Dropout

In the figure 2.18 above, is an ordinary neural network with 2 hidden layers and the neural network has applied the dropout regularization technique where there are several activation neurons that are no longer used.

2.12.7 MobileNet

MobileNets, is one of the Convolutional Neural Networks (CNN) architectures that can be used to overcome the need for excess computing resources. As the name suggests, Mobile, researchers from Google created a CNN architecture that can be used for mobile phones. There are two different version of MobileNet that is MobileNetV1 and MobileNetV2. The MobileNetV2 is mostly an updated version of V1 that makes it even more efficient and powerful in terms of performance. The following discussion below is more detailed explanation of the MobileNets

a) MobileNetV1

The MobileNetV1 model is based on depthwise separable convolutions which is a form of factorized convolutions which factorize a standard convolution into a depthwise convolution and a 1×1 convolution called a pointwise convolution. For MobileNetsV1 the depthwise convolution applies a single filter to each input channel. The pointwise convolution then applies a 1×1 convolution to combine the outputs the depthwise convolution. A standard convolution both filters and combines inputs into a new set of outputs in one step. The depthwise separable convolution splits this into two layers, a separate layer for filtering and a separate layer for combining. This factorization has the effect of drastically reducing computation and model size. MobileNetV1 divides convolution into

depthwise convolution and pointwise convolution as shown in figure 2.19. The architecture of MobileNetV1 itself is shown in figure 2.20.

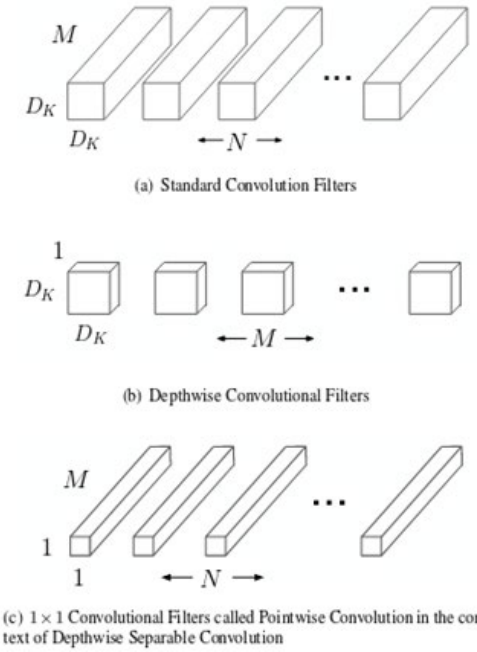

Figure 2.19 Depthwise Separable Convolutions

Figure 2.19 The standard convolutional filters in (a) are replaced by two layers: depthwise convolution in (b) and pointwise convolution in (c) to build a depthwise separable filter.

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Figure 2.20 MobileNet Architecture

Original architecture of MobileNet as shown in figure 2.20 above. The architecture is based on Depthwise Separable Convolutions. (Howard et al, 2017)

b) MobileNetV2

Just like MobilenetV1, MobileNetV2 still uses depthwise and pointwise convolution. MobileNetV2 adds two new features that is linear bottlenecks, and shortcut connections between bottlenecks. The bottlenecks of the MobileNetV2 encode the intermediate inputs and outputs while the inner layer encapsulates the model's ability to transform from lower-level concepts such as pixels to higher level descriptors such as image categories. With traditional residual connections, shortcuts enable faster training and better accuracy. The basic structure of this architecture is shown in figure 2.21 and figure 2.22.

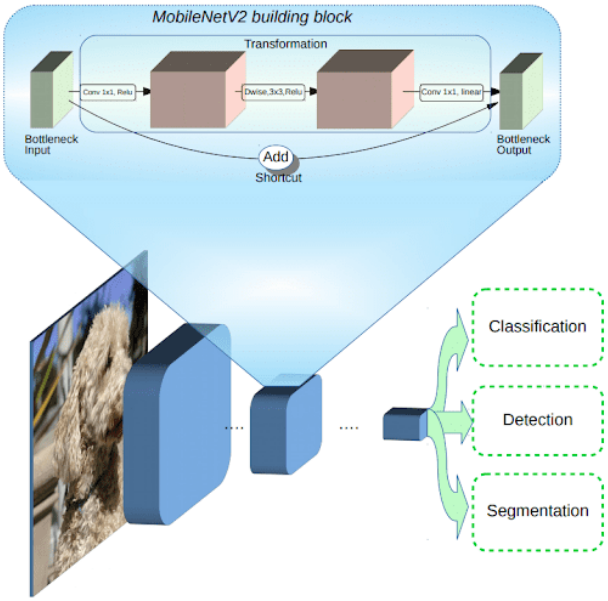

Figure 2.21 The Basic Structure of The MobileNetV2 Architecture

In figure 2.21 above shows that the basic building block is a bottleneck depth-separable convolution with residuals. The architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers. The researchers have tailored the

architecture to different performance points, by using the input image resolution and width multiplier as tunable hyperparameters, that can be adjusted depending on desired accuracy or performance trade-offs. The primary network (width multiplier 1, 224 × 224), has a computational cost of 300 million multiply-adds and uses 3.4 million parameters. The network computational cost ranges from 7 multiply-adds to 585M MAdds, while the model size varies between 1.7M and 6.9M parameters.

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Figure 2.22 The MobileNetV2 Architecture

Now that we understand the building block of MobileNetV2 we can take a look at the entire architecture. In the figure 2.22 you can see how the bottleneck blocks are arranged. t stands for expansion rate of the channels. As you can see, they used a factor of 6 opposed to the 4 in our example. c represents the number of input channels and n how often the block is repeated. Lastly s tells us whether the first repetition of a block used a stride of 2 for the down sampling process. All in all, it's a very simple and common assembly of convolutional blocks (Sandler et al, 2018).

2.13. Shot MultiBox Detector (SSD)

Single Shot MultiBox Detector is a deep learning model used to detect objects in an image or from a video source. Single Shot Detector is a simple approach to solve the problem but it is very effective till now. SSD has two components and they are the Backbone Model and the SSD Head. Backbone Model is a pre-trained image classification network as a feature extractor. Usually, the fully connected classification layer is removed from the model. SSD Head is another set of

convolutional layers added to this backbone and the outputs are interpreted as the bounding boxes and classes of objects in the spatial location of the final layer's activations. The following figure 2.23 is the Shot MultiBox Detector (SSD) architecture

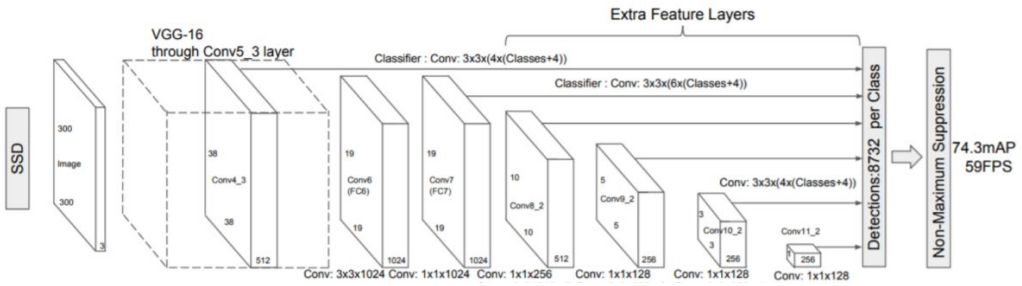

Figure 2.23 Architecture of Single Shot MultiBox detector

Figure 2.23 above is an example of a SSD's architecture builds on the venerable VGG-16 architecture, but discards the fully connected layers. The reason VGG-16 was used as the base network is because of its strong performance in high quality image classification tasks and its popularity for problems where transfer learning helps in improving results. Instead of the original VGG fully connected layers, a set of auxiliary convolutional layers (from conv6 onwards) were added, thus enabling to extract features at multiple scales and progressively decrease the size of the input to each subsequent layer (Forson, 2017).

2.14. Batch Size and Epoch Single

Batch size is the number of data samples that are distributed to the neural network, while epoch is when the entire dataset has gone through the training process on the neural network until it is returned to the beginning and repeat the process, because one epoch is too large to be fed into the computer, therefore it is necessary to divide it into small units (batches).

2.15. Learning Rate

Learning rate, is one of the training parameters to calculate the weight correction value during the training process. This learning rate value is in the range 0 to 1. The greater the learning rate value, the faster the training process will run. The greater the learning rate, the less accurate the neural network will be, on the other hand, if the learning rate is getting smaller, the network accuracy will be

greater or increase with the consequence that the training process will take longer (Lu, 2017).

2.16. Object Detection

Object detection determines the presence of an object and its scope and location in an image. This can be treated as the introduction of a second-class object, where one class represents an object class and another class represents a non-object class. Object detection can be further divided into soft detection and hard detection. Soft detection only detects the presence of objects while hard detection detects the presence of objects and the location of the object (Jalled & Voronkov, 2016).

2.17. OpenCV

OpenCV is an open-source computer vision library. The library is written in C and C++ and runs under Linux, Windows and Mac OS X. There is active development on interfaces for Python, Ruby, Matlab, and other languages. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. OpenCV is written in optimized C and can take advantage of multicore processors. If you desire further automatic optimization on Intel architectures, you can buy Intel's Integrated Performance Primitives (IPP) libraries, which consist of low-level optimized routines in many different algorithmic areas. OpenCV automatically uses the appropriate IPP library at runtime if that library is installed. One of OpenCV's goals is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. The OpenCV library contains over 500 functions that span many areas in vision, including factory product inspection, medical imaging, security, user interface, camera calibration, stereo vision, and robotics. Because computer vision and machine learning often go hand-in-hand, OpenCV also contains a full, general purpose Machine Learning Library (MLL). This sub library is focused on statistical pattern recognition and clustering. The MLL is highly useful for the vision tasks that are at the core of OpenCV's mission, but it is general enough to be used for any Machine Learning problem (Bradski & Kaehler, 2008).

2.18. Confusion Matrix

The Confusion Matrix is a visualization tool commonly used in supervised learning. Each column in the matrix is an example of a prediction class, while each row represents an event in the actual class (Gorunescu, 2011). The confusion matrix is also useful for analyzing how well the classifier recognizes tuples from different classes. TP and TN provide information when the classifier is correct, while FP and FN tell when the classifier is wrong (Han et al, 2011).

Table 2.1 Confusion Matrix

Prediction	Actual	
	Positive	Negative
TRUE	TP	FN
FALSE	FP	TN

Where :

TP = True Positive TN = True Negative

FP = False Positive FN = False Negative

Based on table 2.1 True Positive is the number of positive records classified as positive, false positive is the number of negative records classified as positive, false negative is the number of positive records classified as negative, true negative is the number of negative records classified as negative (Andriani, 2013).

The evaluation will be carried out using the F-Measure parameter which consists of calculating Precision, and Recall. Recall, precision and F-measure are methods of measuring effectiveness in the classification process. Recall and precision are two criteria used to evaluate the effectiveness of information retrieval system performance (Hayuningtyas, 2017).

a) Precision

Precision or Confidence shows that the proportion of positive case predictions that are actually positive, but analogically can be called True Positive Accuracy (TPA), being a measure of the accuracy predicted positives is different from the real positive discovery rate (TPR) (Powers, 2020). Precision is defined as follows:

$$Precision\ (positive) = \frac{TP}{TP + FP} \quad (2.4)$$

Inverse precision is a comparison of the negative prediction cases that are actually negative, and can also be called True Negative Accuracy (TNA) (Powers, 2020):

$$Inverse\ Precision\ (negative) = \frac{TN}{TN + FN} \tag{2.5}$$

b) Recall

Recall or Sensitivity indicates that the proportion of positive real cases that are correctly predicted to be positive. This feature is used to describe how many relevant cases are drawn from positive predictions. In this context it is called True Positive Rate (TPR) (Powers, 2020). Recall is defined as follows:

$$Recall\ (positive) = \frac{TP}{TP + FN} \tag{2.6}$$

Inverse recall is a comparison of real negative cases that are correctly predicted to be negative, and is also known as True Negative Rate (TNR) (Powers, 2020).

$$Recall\ (negative) = \frac{TN}{FP + TN} \tag{2.7}$$

c) Accuracy

is calculated as the number of all correct predictions divided by the total number of the dataset.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.8}$$

d) F1 Score

F1 score is the average of weighted precision and recall (Sokolova & Lapalme, 2009).

$$F1(positive) = 2 * \frac{precision(p) * recall(p)}{precision(p) + recall(p)}$$
$$F1(negative) = 2 * \frac{precision(n) * recall(n)}{precision(n) + recall(n)} \tag{2.9}$$

Where:

P = Positive

N = Negative