People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdes

Institute of Electrical and Electronic Engineering

Department of Electronics

Final Year Project Report Presented in Partial Fulfilment of
The Requirements for the Degree of

MASTER

In **Electronics**

Option: **Computer Engineering**

Title:

Skin Cancer and Covid19 Classification using Machine and Deep Learning

Presented by:

- **Feghoul Amine**
- **Ferarha Djamal Eddine**

Supervisor:

- **Dr. CHERIFI Dalila**

Registration Number:2019/2020

DEDICATIONS

I would love to dedicate this work

To

My parents that has made me the man I am today

To

My brothers and Sister who never hesitated to help

To

My dear friends and all people I have ever known

To

The most special teacher I had the honor to be student of

"Dr. CHERIFI Dalila"

Amine

DEDICATIONS

I would like to thank God for helping me through all the way. I dedicate this work To my family and many friends. A special feeling of gratitude to my loving parents, my Mother and my Father whose words of encouragement and push for tenacity ring in my ears. I also dedicate this work to my friends and my family who have supported me throughout the process. I will always appreciate all they have done for me.

Djamal Eddine

ACKNOWLEDGMENTS

First and foremost, we would like to express our sincere gratitude and appreciation to our supervisor **Dr. CHERIFI Dalila** for her continuous help and orientation throughout the different stages of this project. Also, we would like to thank all people that helped us all the way to build our skillset and realizing this project.

Special thanks to all **professors** and **teachers** that have guided us and have given us all the needed information to build up a rich background for our future engineering life.

ABSTRACT

On one hand, skin cancer is one of the most known cancer in the world, the early detection plays a major role in the ability to remove this kind of tumors. On another hand, Covid-19 is the most dangerous corona virus that has spread around the world in 2020.

One of the fastest and most useful ways to achieve early detection is to use machine learning and deep learning classifiers. To get a good performance, the accuracy of the classifier should be high so the patients may have a clear idea about their state.

For this purpose, there are many hyper parameters that can be changed in order to improve the performance of the artificial models that are used for the identification of such illnesses.

In this project we have applied some classification algorithms on two applications which are Covid-19 identification and multiclass skin cancer classification.

In the first application, we have applied the classification algorithms on the Covid-19 data set and we have got good performance concerning the random Forest and SVM classifiers and acceptable accuracy by the CNN models due to the lack of data samples.

In the second application, we have adapted the same models to be applied on the skin cancer dataset. In this part, CNN models have overpassed the other algorithms in the performance.

After that we have compared the results of the two applications and we have suggested some methods in order to improve the performance of these classification algorithms.

Table of contents

List of figures

List of tables

List of abbreviations

ACC	.	Accuracy
ANN		Artificial Neural Network
SVM		Support Vector Machine
RF		Random Forest
Sens		Sensitivity
Speci		specificity

INTRODUCTION

Classification is one of the major fields of artificial intelligence, classification is used in almost all applications in different fields starting from security field, medical field to self-driving cars. In other words classification is the most used type of artificial intelligence especially in computer vision. There are two types of classification: binary classification and multiclass classification.

Binary classification is when the output can be only of two values, while multiclass classification the output may take more than two values. In this project we are aiming to implement both types of classification binary through the Covid-19 presence identification, and multiclass in the classification of skin cancers.

Skin cancers are cancers that arise from the skin. They are due to the development of abnormal cells that have the ability to invade or spread to other parts of the body. There are three main types of skin cancers: basal-cell skin cancer (BCC), squamous-cell skin cancer (SCC) and melanoma. The first two, along with a number of less common skin cancers, are known as nonmelanoma skin cancer (NMSC). Basal-cell cancer grows slowly and can damage the tissue around it but is unlikely to spread to distant areas or result in death. It often appears as a painless raised area of skin that may be shiny with small blood vessels running over it or may present as a raised area with an ulcer. Squamous-cell skin cancer is more likely to spread. It usually presents as a hard lump with a scaly top but may also form an ulcer. Melanoma is the deadliest of skin cancers. Rates of diagnosis for the disease have increased dramatically over the past three decades, outpacing almost all other cancers. Today, it is one of the most common cancers found among young adults in the World.

In 2020, the world is facing one the most complicated viruses in the 21s century which is Covid-19. Coronavirus disease 2019 (COVID-19) is an infectious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). It was first identified in December 2019 in Wuhan, Hubei, China, and has resulted in an ongoing pandemic. The first confirmed case has been traced back to 17 November 2019 in Hubei. As of 4 August 2020, more than 18.4 million cases have been reported across 188 countries and territories, resulting in more than 697,000 deaths. More than 11 million people have recovered[1].

Common symptoms include fever, cough, fatigue, shortness of breath, and loss of smell and taste. While the majority of cases result in mild symptoms, some progress to acute respiratory distress syndrome (ARDS) possibly precipitated by cytokine storm, multi-organ failure, septic shock, and blood clots. The time from exposure to onset of symptoms is typically around five days, but may range from two to fourteen days.

The virus is primarily spread between people during close contact, most often via small droplets produced by coughing, sneezing, and talking. The droplets usually fall to the ground or onto surfaces rather than travelling through air over long distances. However, the transmission may also occur through smaller droplets that are able to stay suspended in the air for longer period of time in enclosed spaces, as typical for airborne diseases. Less commonly, people may become infected by touching a contaminated surface and then touching their face. It is most contagious during the first three days after the onset of symptoms, although spread is possible before symptoms appear, and from people who do not show symptoms. The standard method of diagnosis is by real-time reverse transcription polymerase chain reaction from a nasopharyngeal swab. Chest CT imaging may also be helpful for diagnosis in individuals where there is a high suspicion of infection based on symptoms and risk factors; however, guidelines do not recommend using CT imaging for routine screening.

The objective of this project is to develop and implement classification approaches using machine learning (SVM and Random Forest classifier) and deep learning models (CNNs) in order to classify the skin tumors images into skin cancer types and to identify the presence of covid-19 based on CT scans images.

This report consists of three chapters. The **first chapter** aims to give a brief description about the abnormal cases we are going to deal with for both diseases Skin cancer and covid-19. The **second chapter** explains the used machine learning Algorithms and deep learning Classifier we are going to use in our experimental part .We have used three classifiers, which are, Support Vector Machine, Random Forest Classifier and Artificial Neural Network. Ultimately, **chapter three** presents the experimental results obtained from applying the algorithms described in chapter two on the datasets we have chosen. We end up our report with a brief conclusion, in which we suggest some future work to be done by coming students.

CHAPTER I

Review about Skin Cancer and Covid-19

I.1. Skin Cancer:

Skin cancer is the out-of-control growth of abnormal cells in the epidermis, the outermost skin layer, caused by unrepaired DNA damage that triggers mutations. These mutations lead the skin cells to multiply rapidly and form malignant tumors. The main types of skin cancer are basal cell carcinoma (BCC), squamous cell carcinoma (SCC), melanoma and Merkel cell carcinoma (MCC). The two main causes of skin cancer are the sun's harmful ultraviolet (UV) rays and the use of UV tanning machines. The good news is that if skin cancer is caught early, the dermatologist can treat it with little or no scarring and high odds of eliminating it. Often, the doctor may even detect the growth at a precancerous stage, before it has become a full-blown skin cancer or penetrated below the surface of the skin.

I.1.1.Types of Skin Cancer:

There are many types of the skin cancer, but our work is focused only on the following types:

1. **Actinic keratosis:**

Actinic keratosis is a rough, scaly patch on the skin. It is most commonly found on the face, lips, ears, back of hands and forearms, scalp or neck. Also known as a solar keratosis, an actinic keratosis enlarges slowly and usually causes no signs or symptoms other than a patch or small spot on the skin. These patches take years of exposure to the sun to develop, usually first appearing in people over 40. The percentage of actinic keratosis lesions that can become skin cancer is very small. Minimizing the sun exposure and protecting the skin from ultraviolet (UV) rays can reduce the risk of actinic keratoses [2].

Figure I.1: *Actinic keratosis [2]*.

A. Symptoms:

The signs and symptoms of an actinic keratosis include:

- Dry or squamous patch of skin, usually less than 2.5 centimeters in diameter
- Flat to slightly raised patch or bump on the top layer of skin.
- In some cases, a hard, wartlike surface.
- The color maybe pink, red or brown.
- Itching or burning in the affected area

B. Causes:

The most known cause of actinic keratosis is the frequent or intense exposure to UV rays from the sun or tanning beds.

C. Complications:

Generally, actinic keratosis can be removed and cleared up from the skin if they are treated early before they develop to skin cancer. In some untreated cases, these skin spots may progress to squamous cell carcinoma which is a not life threating cancer if detected and treated early.

D. Risk factors:

Anyone can develop actinic keratoses. But the chance to develop this kind of skin patches is higher for people that:

- Are older than 40
- Live in a sunny places.
- Have been exposed to frequent of intense to sun rays.
- Have red or blond hair and blue or light-colored eyes.
- Have a faced actinic keratosis or skin cancer before.
- Have a weak immune system.

2. Basal Cell Carcinoma:

The most common and the most frequently occurring form of skin cancer is Basal cell carcinoma (BCC). Each year, In the U.S. alone, more than 4 million cases are diagnosed to be BCC. This type of cancer arises from abnormal, uncontrolled growth of basal cells. The growth of BCCS is slowly which makes most of them curable and cause minimal damage if they are treated early [3].

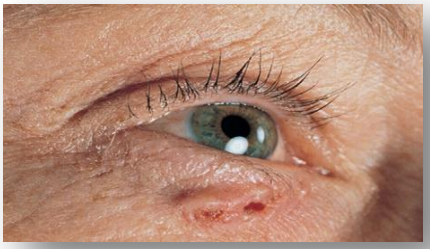

Figure I.2: *Basal cell carcinoma[3]*

A. Symptoms:

Usually, BCC develops on the parts of body that are exposed to sun, especially the head and the neck. Less often, basal cell carcinoma can develop on parts of the body that are usually protected from the sun, such as the genitals. Basal cell carcinoma appears as a change in the skin, such as a growth or a sore that won't heal. These changes in the skin (lesions) usually have one of the following characteristics:

- A lesion with dark spots with a slightly raised, translucent border.
- A flat, scaly, reddish patch with a raised edge is more common on the back or chest. These patches can grow quite large over time.
- A white, waxy, scar-like lesion without a clearly defined border, called morpheaform basal cell carcinoma, is the least common.

B. Causes:

The main reason of Basal cell carcinoma is the occurrence of mutation in the DNA in basal cells. Basal cells are found in underneath the outermost layer of skin. These cells produce new skin cells. When new cells are produced, they push the older cells toward the surface. Where they die. The process of creating new skin cells is controlled by a basal cell's DNA. The mutation that has occurred on the DNA tells basal cells to multiply rapidly and continuously when it would normally die. Eventually the accumulating abnormal cells may form a cancerous tumor the lesion

that appears on the skin. Much of the DNA damage in basal cells is resulted from the UV radiation found in the sunlight and in commercial tanning lamps and tanning beds.

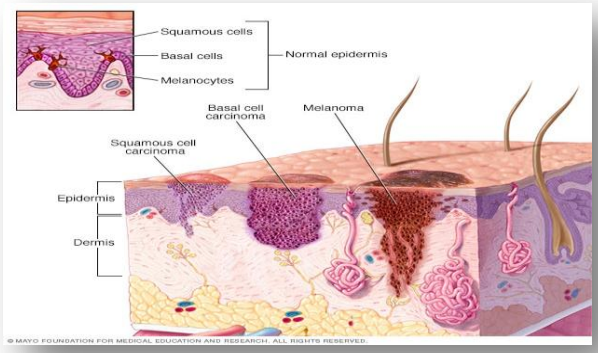

Figure I.3: *The layer where skin cancer develops[3]*

C. Complications:

Complications of basal cell carcinoma can include:

- There is a very high probability for recurrence**.**
- Having history with of basal cell carcinoma may increase the chance of developing other types of skin cancers.
- Cancer that spreads beyond the skin**.** Very rarely, basal cell carcinoma can spread (metastasize) to nearby lymph nodes and other areas of the body, such as the bones and lungs.

D. Risk factors:

Factors that increase the risk of basal cell carcinoma include:

- **Chronic sun exposure.** A lot of time spent in the sun or in commercial tanning beds increases the risk of basal cell carcinoma. The threat is greater if the patient lives in a sunny or high-altitude location, both of which expose him to more UV radiation. Severe sunburns also increase the risk.
- **Radiation therapy.** Radiation therapy to treat acne or other skin conditions may increase the risk of basal cell carcinoma at previous treatment sites on the skin.
- **Fair skin.** The risk of basal cell carcinoma is higher among people who freckle or burn easily or who have very light skin, red or blond hair, or light-colored eyes.

- **Increasing age.** Because basal cell carcinoma often takes decades to develop, the majority of basal cell carcinomas occur in older adults. But it can also affect younger adults and is becoming more common in people in their 20s and 30s.

3. Seborrheic keratosis

A seborrheic keratosis is a very common noncancerous skin growth. People are most likely to get when they get older. Usually, they are of a dark color brown or black or they can be light tan. These growths look waxy, Squamous and slightly raised. They appear on different parts of the body such as the head, neck, chest or back. Seborrheic keratoses are harmless and not contagious. Most of time they don't need treatment, but they maybe removed if they become irritated by clothing or if they don't look good [4]:

Figure.1.4 : Close-up of seborrheic keratosis[4]

Figure.1.5 : Seborrheic keratosis on the back[4]

A seborrheic keratosis usually looks like a waxy or wartlike growth. It typically appears on the face, chest, shoulders or back. They might develop as single growth, but multiple growths are more common. A seborrheic keratosis:

- Color ranges from light tan to brown or black
- Is flat or slightly raised with a scaly surface.
- Ranges in size from very small to more than 2.5 centimeters across.
- May itch.

B. Cause:

Doctors couldn't figure out the reason behind seborrheic keratosis, but some families tends more to have them. So most probably that the genes play a role in the growth of seborrheic keratosis.

C. Risk factors:

Generally, people are more likely to develop seborrheic keratoses if they are over age 50. They also more likely to have them if they have a family history of the condition.

4. Dermato fibroma:

Dermatofibromas are small harmless growths that appear on the skin and can grow anywhere in the body. They are also called nodules, they are most commonly found on the arms, lower legs and on the upper back. Dermatofibromas are seen in adults but are rare in children and more frequent in women than in men. Also they are more in common in people with weak immune systems. Dermatofibromas are small in diameter and the may vary in color, the color may change over the years. Many people say that Dermatofibromas feel like small stone underneath or above the skin. Most of Dermatofibromas cause no pain, but they might be itchy or they may cause irritation on the site of the growth. Dermatofibromas may also be called benign fibrous histiocytomas [5].

A. Symptoms and complications :

Dermatofibromas tend to grow slowly. The growths typically have some defining characteristics that can help identify them. Key markers of a dermatofibroma are:

- Appearance : a round bump that is mostly under the skin.
- Size : the normal range is about the size of the tip of a ballpoint pen to a pea, and it usually remains stable.
- The color can be pink, gray or light brown in different degrees, it may change over time.
- dermatofibroma most likely grow in the legs and sometimes on the arms, trunk and there is very small percentage to be on other parts of the body.
- Usually, dermatofibroma are harmless and painless, but occasionally may be itchy, tender, painful, or feel inflamed.

When pinched, a dermatofibroma will not push towards the surface of the skin. Instead, it will dimple inward on itself, which can help tell the difference between a dermatofibroma and another type of growth. It is usual for only one growth to appear on the body. However, multiple dermatofibromas are more likely to occur in people with weakened immune systems. Skin growths can be alarming, but dermatofibromas are harmless and do not develop into cancerous growths.

B. Causes and Risk Factors:

Dermatofibromas are an accumulation of extra cells within the deeper layers of the skin. The exact cause of these growths is unknown. They may be caused by an adverse reaction to a small injury, such as a bug bite, splinter, or puncture wound.

Age may be another risk factor, as the growths appear mostly in adults. People with suppressed immune systems may be more likely to experience dermatofibromas, and may have more than one growth. Multiple dermatofibromas are especially common in people with systemic lupus.

5. Vascular lesions:

Vascular birthmarks are common, idiopathic clusters of blood vessel growths found in infants and children. They may not be of noticeable size after months or years from birth. These birthmarks are categorized in two groups: tumors and malformations. Vascular tumors are neoplasms, involving cellular spread of vessels. Vascular malformations, on the other hand, do not involve the spread of the blood vessels and are stable throughout one's lifetime; they are further classified according to the type of vessels found in the birthmark. Examples of vascular tumors include [6]:

- Hemangiomas
- Tufted angioma
- Kaposiform hemangioendothelioma
- Pyogenic granuloma

Examples of vascular malformations include:

- Capillary
- Venous
- Lymphatic
- Arterial
- Arteriovenous
- Complex/combined

Oftentimes, these vascular birthmarks are stand-alone lesions, without association with other findings. However, vascular birthmarks may be found in a larger constellation of physical findings, comprising a vascular malformation syndrome such as Sturge-Weber syndrome or Klippel-Trenaunay syndrome. This is one reason why it is important for vascular birthmarks to be evaluated by a pediatric dermatologist.

Figure I.6: *Some types of vascular lesions[6]*

6. Melanocytic nevus:

A **melanocytic nevus** is a type of melanocytic tumor that contains nevus cells. Some people call the term mole on "melanocytic nevus". The majority of moles appear before the age of 20s, with about one in every 100 babies being born with moles. Acquired moles are a form of benign neoplasm, while congenital moles, or congenital nevi, are considered a minor malformation or hamartoma and may be at a higher risk for melanoma. A mole can be either subdermal (under the skin) or a pigmented growth on the skin, formed mostly of a type of cell known as a melanocyte. The high concentration of the body's pigmenting agent, melanin, is responsible for their dark color [7].

A. Symptoms:

Benign moles are usually:

- Brown, tan, pink or black (especially on dark-colored skin).
- Circular or oval and are usually small (commonly between 1–3 mm)

Some moles produce dark, coarse hair. Common mole hair removal procedures include plucking, cosmetic waxing, electrolysis, threading and cauterization.

7. Melanoma:

Melanoma is the most serious and dangerous type of skin cancer. It develops in the melanotic cells that are responsible on producing melanin that gives the color of the skin. Melanoma can be in the eyes also and rarely inside the body such as the noise or the throat. The main reason of melanoma is not clear yet to the doctors but the exposure to UV radiation increase the risk of having this kind of tumors. Limiting the exposure to UV radiation may reduce the risk of melanoma. Melanoma can be treated if it is detected in early stages so it is so important to know the warning signs of skin cancer [8].

Figure I.7: *Melanoma [8]*

A. Symptoms:

Melanoma is most likely to be developed in areas that had much exposure to UV radiation such as head, neck, back legs, but this doesn't that melanoma can't be on other parts of the body. Melanomas can also occur in areas that don't receive much sun exposure, such as the soles of the feet, palms of hands and fingernail beds. These hidden melanomas are more common in people with darker skin. The first melanoma signs and symptoms often are:

- A change in an existing mole
- The development of a new pigmented or unusual-looking growth on the skin

Melanoma doesn't always begin as a mole. It can also occur on normal-appearing skin.

B. Risk factors:

Factors that may increase the risk of melanoma include:

- Having a fair skin means having less protection from UV radiation damage due to lack of melanin in the skin. White people with blond or red hair are more likely to develop Melanoma than people with darker complexion including Hispanic and black people.
- Blistering sunburns can increase the risk of melanoma.
- Excessive ultraviolet (UV) light exposure from the sun and from tanning lights and beds, can increase the risk of skin cancer, including melanoma.
- People that live closer to the earth's equator, where the sun's rays are more direct, experience higher amounts of UV radiation than do those living farther north or south. In addition, if the patient lives at a high elevation, he is exposed to more UV radiation.
- Having many moles or unusual moles. Having more than 50 ordinary moles on the body indicates an increased risk of melanoma. Also, having an unusual type of mole increases the risk of melanoma.

- A family history of melanoma. If a close relative such as a parent, child or sibling has had melanoma, it has greater chance to develop a melanoma, too.
- Weakened immune system. People with weakened immune systems have an increased risk of melanoma and other skin cancers. the immune system may be impaired if a medicine to suppress the immune system are taken, such as after an organ transplant, or in case of a disease that impairs the immune system, such as AIDS.

I.1.2. Skin Cancer Prevention:

UV radiation from the sun isn't just dangerous, it's also sneaky. Not only it can cause premature aging and skin cancer, it reaches you even when you're trying to avoid it penetrating clouds and glass, and bouncing off of snow, water and sand. What's more, sun damage accumulates over the years, from prolonged outdoor exposure to simple activities like walking the dog, going from the car to the store and bringing in the mail.

That's why preventing skin cancer by protecting yourself completely requires a comprehensive approach. **The Skin Cancer Foundation** recommends to:

- **Seek the shade**, especially between 10 AM and 4 PM.
- **Don't get sunburned.**
- **Avoid tanning**, and never use UV tanning beds.
- **Cover up** with clothing, including a broad-brimmed hat and UV blocking sunglasses.
- **Use a broad spectrum (UVA/UVB) sunscreen** with an SPF of 15 or higher every day. For extended outdoor activity, use a water-resistant, broad- spectrum (UVA/UVB) sunscreen with an SPF of 30 or higher.
- **Apply 1 ounce (2 tablespoons) of sunscreen** to the entire body 30 minutes before going outside. Reapply every two hours or after swimming or excessive sweating. Find sunscreen by searching our Recommended Products.
- **Keep newborns out of the sun**. Use sunscreen on babies over the age of six months.
- **Examine your skin** head-to-toe every month.
- **See a dermatologist** at least once a year for a professional skin exam.
- **Get all the details:** *the Daily Sun Protection Guide*.

I.2. Coronaviruses:

Coronaviruses are viruses that can cause respiratory tract infections ,this infection can range from mild to lethal. while more lethal varieties can cause SARS, MERS, and COVID-19, Mild illnesses can derive some cases of the common cold (which is also caused by other viruses, predominantly rhinoviruses). Symptoms in other species vary: in cows and pigs they cause diarrhea , while in chickens, they cause an upper respiratory tract disease. Coronaviruses are related RNA viruses that cause diseases in all species which include mammals ,birds and humans. There are as yet no vaccines or antiviral drugs to prevent or treat human coronavirus infections. They have characteristic club-shaped spikes that project from their surface, which in electron micrographs create an image reminiscent of the solar corona, from which their name derives.

I.2.1 Coronaviruses description:

1. Where the name of coronaviruse comes from?

The name Coronavirus comes from the crown-like spikes on their surface,"coronavirus" means “crown” derived from Latin name “corona” due to the characteristic appearance of virions (the infective form of the virus) by electron microscopy. Coronaviruses have a fringe of large, bulbous surface projections creating an image reminiscent of the solar corona or halo. which are proteins on the surface of the virus. It first named by June Almeida and David Tyrrell who first observed and studied human coronaviruses. [9]
Coronavirus Evolution Scientists first identified a human coronavirus in 1965. It caused a common cold. Later that decade, researchers found a group of similar human and animal viruses and named them after their crown-like appearance. Seven coronaviruses can infect humans. The one that causes SARS emerged in southern China in 2002 and quickly spread to 28 other countries. More than 8,000 people were infected by July 2003, and 774 died. A small outbreak in 2004 involved only four more cases. This coronavirus causes fever, headache, and respiratory problems such as cough and shortness of breath. MERS started in Saudi Arabia in 2012. Almost all of the nearly 2,500 cases have been in people who live in or travel to the Middle East [10]. This coronavirus is less contagious than its SARS cousin but more deadly, killing 858 people. It has the same respiratory symptoms but can also cause kidney failure. Coronaviruses didn’t just pop up recently. They’re a large family of viruses that have been around for a long time. Many of them can make people ill with sniffles or coughing. Before the SARS-CoV-2 outbreak, coronaviruses were thought to cause only mild respiratory infections in people.

2. Coronavirus under electron micrographs:

Coronaviruses in electron micrographs appears as a distinct pair of electron-dense shells (shells that are relatively opaque to the electron beam used to scan the virus particle) [11]. The viral envelope consists of a lipid bilayer, in which the membrane (M), envelope (E) and spike (S) structural proteins are anchored. The average diameter of the virus particles is around 125 nm (0.125 μm). The diameter of the envelope is 85 nm and the spikes are 20 nm long. The S2 subunit forms the stem which anchors the spike in the viral envelope and on protease activation enables fusion. The E and M protein are important in forming the viral envelope and maintaining its structural shape. Inside the envelope, there is the nucleocapsid, which are bound to the positive-sense single-stranded RNA genome in a continuous beads-on-a-string type conformation. The lipid bilayer envelope, membrane proteins, and nucleocapsid protect the virus when it is outside the host cell.

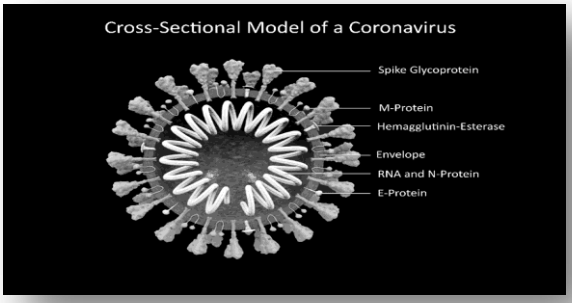

Figure I.8: *Cross-sectional model of a coronavirus[12]*

3. Steps of the virus infections:

Cell entry: Depending on the host cell protease available, infection begins when the viral spike protein attaches to its complementary host cell receptor. After attachment, a protease of the host cell cleaves and activates the receptor-attached spike protein, cleavage and activation allows the virus to enter the host cell by endocytosis or direct fusion of the viral envelop with the host membrane.

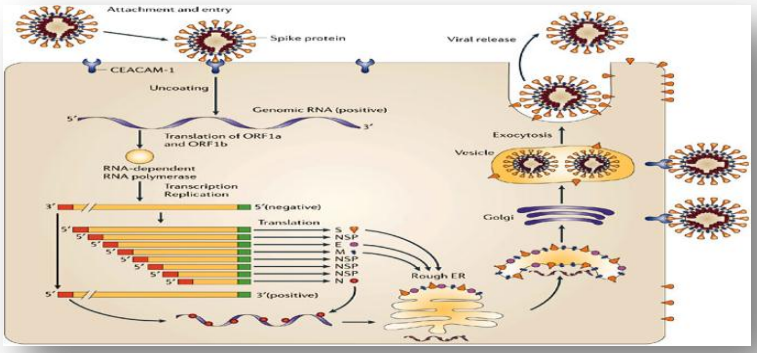

Figure I.9: The life cycle of a coronavirus[13]

4. Replication of RNA:

A number of the nonstructural proteins coalesce to form a multi-protein replicase-transcriptase complex. The main replicase-transcriptase protein is the RNA-dependent RNA polymerase (RdRp). It is directly involved in the replication and transcription of RNA from an RNA strand. The other nonstructural proteins in the complex assist in the replication and transcription process. The exoribonuclease nonstructural protein, for instance, provides extra fidelity to replication by providing a proofreading function that the RNA-dependent RNA polymerase lacks[14].

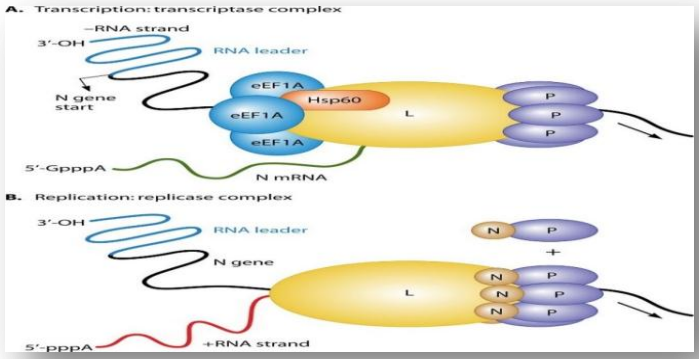

Figure 1.10: *Replicas-transcriptase complex[14]*.

5. RNA of the virus:

Viruses belonging to the family Coronaviridae are unique among RNA viruses because of the unusually large size of their genome, which is of messenger- or positive- or plus-sense. It is ~30,000 bases or 2–3 times larger than the genomes of most other RNA viruses. Coronaviruses belong to the order Nidovirales, the other three families being the Arteriviridae. Among the Nidovirales, coronaviruses (and toroviruses) are unique in their possession of a helical nucleocapsid, which is unusual for plus-stranded but not minus-stranded RNA viruses; plus-stranded RNA-containing plant viruses in the Closteroviridae and in the

Tobamovirus genus also possess helical capsids. Coronaviruses are very successful and have infected many species of animals, including bats, birds (poultry) and mammals, such as humans and livestock.

The genome of coronaviruses is depicted in Figure I.10 Its length varies from ~27.5 to ~31 kb among the various species of coronaviruses, depending on the virus, which are each preceded by a short repeated sequence called the transcription regulating sequence (TRS) immediately upstream of the initiating AUG for that ORF. A TRS is also found about 65 nts from the 5'-end of the genome. The sequence at the 5' end of the genome, up to this first TRS, is called the leader sequence (Figure I.10). The organization of multiple genes was first observed with IBV when its genome was sequenced, which was a feat of manual sequencing skill. After MHV and other coronaviruses were sequenced and shown to have a similar size and organization, equine arteritis virus (EAV, the type member of the Arteriviridae) was sequenced and found to have a similar organization of genes but with half the number of bases as coronaviruses[15].

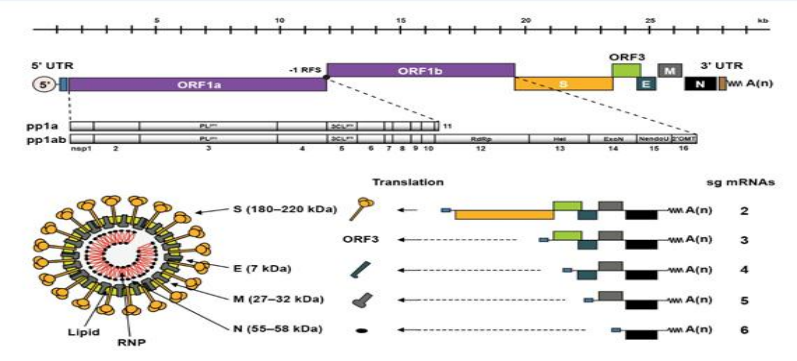

Figure I.11: *Schematic representation of the genome organization and functional domains of covid-19 [16]*

6. Assembly and release:

RNA translation occurs inside the endoplasmic reticulum. The viral structural proteins S, E, and M move along the secretory pathway into the Golgi intermediate compartment. The replicated positive-sense genomic RNA becomes the genome of the progeny viruses. The mRNAs are gene transcripts of the last third of the virus genome after the initial overlapping reading frame. These mRNAs are translated by the host's ribosomes into the structural proteins and a number of accessory proteins. There, the M proteins direct most protein-protein interactions required for assembly of viruses following its binding to the nucleocapsid. Progeny viruses are then released from the host cell by exocytosis through secretory vesicles. Once released the viruses can infect other host cells.

7. Transmission:

Coronavirus are transmitted from one host to another host, depending on the coronavirus species, by either an aerosol, fomite, or fecal-oral route. The interaction of the coronavirus spike protein with its complementary cell receptor is central in determining the tissue tropism, infectivity, and species range of the released virus. Coronaviruses mainly target epithelial cells. Human coronaviruses infect the epithelial cells of the respiratory tract, while animal coronaviruses generally infect the epithelial cells of the digestive tract. SARS coronavirus, for example, infects via an aerosol route, the human epithelial cells of the lungs by binding to the angiotensin-converting enzyme 2 (ACE2) receptor.

8. History:

Arthur Schalk and M.C. Hawn discovered in 1931 a new respiratory infection of chickens in North Dakota. The infection of new-born chicks was characterized by gasping and listlessness. The chicks' mortality rate was 40–90%. Fred Beaudette and Charles Hudson six years later successfully isolated and cultivated the infectious bronchitis virus which caused the disease. In the 1940s, two more animal coronaviruses, mouse hepatitis virus (MHV) and transmissible gastroenteritis virus (TGEV), were isolated. It was not realized at the time that these three different viruses were related .

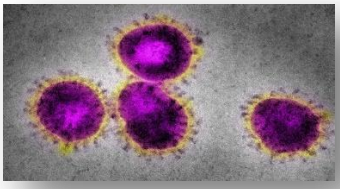

Figure I.12: *Electron micrograph of Coronavirus[17]*

E.C. Kendall, Malcom Byone, and David Tyrrell working at the Common Cold Unit of the British Medical Research Council in 1960 isolated from a boy a novel common cold virus B814 this considered as the first Human coronaviruses infection. The virus was not able to be cultivated using standard techniques which had successfully cultivated rhinoviruses, adenoviruses and other known common cold viruses. In 1965, Tyrrell and Byone successfully cultivated the novel virus by serially passing it through organ culture of human embryonic trachea. The new cultivating method was introduced to the lab by Bertil Hoorn. A research group at the National Institute of Health the same year was able to isolate another member of this new group of viruses using organ culture and named the virus strain OC43 (OC for organ culture). Like B814, 229E, and IBV, the novel cold virus OC43 had distinctive club-like spikes when observed with the electron microscope. It is not known which present human coronavirus it was. Other human coronaviruses have since been identified, including SARS-CoV in 2003, HCoV NL63 in 2004,

HCoV HKU1 in 2005, MERS-CoV in 2012, and SARS-CoV-2 in 2019. There have also been a large number of animal coronaviruses identified since the 1960s.

9. The host's species of the virus:

Bats and birds, as warm-blooded flying vertebrates, are an ideal natural reservoir for the coronavirus gene pool. It is estimated that this type of coronaviruses existed as recently as 8000 BCE, although some models place the common ancestor as far back as 55 million years or more, implying long term coevolution with bat and avian species. The large number and global range of bat and avian species that host viruses has enabled extensive evolution and dissemination of coronaviruses. Many human coronaviruses have their origin in bats. The human coronavirus NL63 shared a common ancestor with a bat coronavirus (ARCoV.2) between 1190 and 1449 CE. The human coronavirus 229E shared a common ancestor with a bat coronavirus (GhanaGrp1 Bt CoV) between 1686 and 1800 CE. More recently, alpaca coronavirus and human coronavirus 229E diverged sometime before 1960. MERS-CoV emerged in humans from bats through the intermediate host of camels. Phylogentically, mouse hepatitis virus (Murine coronavirus), which infects the mouse's liver and central nervous system, is related to human coronavirus OC43 and bovine coronavirus. Human coronavirus HKU1, like the aforementioned viruses, also has its origins in rodents. MERS-CoV, although related to several bat coronavirus species, appears to have diverged from these several centuries ago. The most closely related bat coronavirus and SARS-CoV diverged in 1986. Later in the 1890s, human coronavirus OC43 diverged from bovine coronavirus after another cross-species spillover event. It is speculated that the flu pandemic of 1890 may have been caused by this spillover event, and not by the influenza virus, because of the related timing, neurological symptoms, and unknown causative agent of the pandemic. Besides causing respiratory infections, human coronavirus OC43 is also suspected of playing a role in neurological diseases. In the 1950s, the human coronavirus OC43 began to diverge into its present genotypes

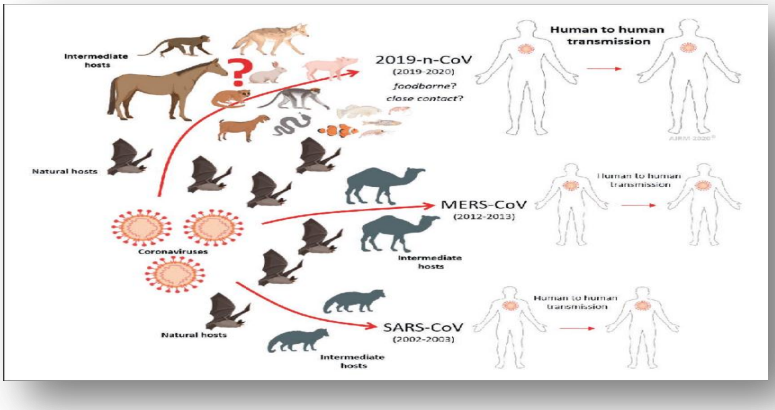

***Figure I.13**: Origins of human Coronaviruses with possible intermediate hosts[18]*

10. Coronaviruses on human body:

Coronaviruses vary significantly in risk factor. Some can kill more than 30% of those infected, such as MERS-CoV, and some are relatively harmless, such as the common cold. Coronaviruses can cause colds with major symptoms, such as fever, and a sore throat from swollen adenoids. Coronaviruses can cause pneumonia (either direct viral pneumonia or secondary bacterial pneumonia) and bronchitis (either direct viral bronchitis or secondary bacterial bronchitis). The human coronavirus discovered in 2003, SARS-CoV, which causes severe acute respiratory syndrome (SARS), has a unique pathogenesis because it causes both upper and lower respiratory tract infections. Six species of human coronaviruses are known, with one species subdivided into two different strains, making seven strains of human coronaviruses altogether.

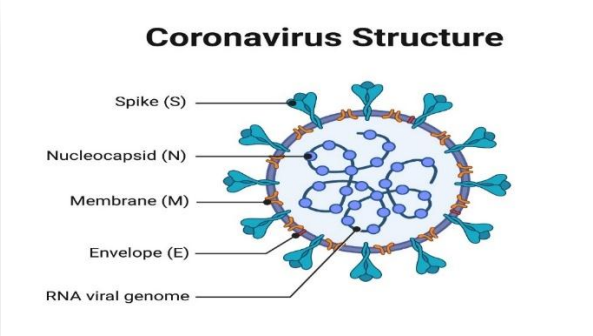

Figure I.14: *Illustration of SARSr-CoV structure[19]*

Four human coronaviruses produce symptoms that are generally mild:

- Human coronavirus OC43.
- Human coronavirus HKU1.
- Human coronavirus 229E.
- Human coronavirus NL63.

Three human coronaviruses produce symptoms that are potentially severe:

- Middle East respiratory syndrome-related coronavirus (MERS-CoV).
- Severe acute respiratory syndrome coronavirus (SARS-CoV).
- Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2).

In 2003, following the outbreak of severe acute respiratory syndrome (SARS) which had begun the prior year in Asia, and secondary cases elsewhere in the world, the World Health Organization (WHO) issued a press release stating that a novel coronavirus identified by a number of laboratories was the causative agent for SARS. The virus was officially named the SARS coronavirus (SARS-CoV). More than 8,000 people were infected, about ten percent of whom died.

I.2.2. Coronavirus disease 2019 (COVID-19):

On 31 December 2019, a pneumonia outbreak was reported in Wuhan, China., the outbreak was traced to a novel strain of coronavirus, which was given the interim name 2019-nCoV by the World Health Organization (WHO), later renamed SARS-CoV-2 by the International Committee on Taxonomy of Viruses. As of 3 July 2020, there have been at least 521,355 confirmed deaths and more than 10,874,146confirmed cases in the COVID-19 pandemic. The Wuhan strain has been identified as a new strain of Betacorona virus from group 2B with approximately 70% genetic similarity to the SARS-CoV. The virus has a 96% similarity to a bat coronavirus, so it is widely suspected to originate from bats as well. The pandemic has resulted in travel restrictions and nationwide lockdowns in many countries.

1. The virus in animals:

Coronaviruses can infect animals including swine, cattle, horses, camels, cats, dogs, rodents, birds and bats ,it have been recognized as causing pathological conditions in veterinary medicine since the 1930s. research efforts have been focused on elucidating the viral pathogenesis of these animal coronaviruses, especially by virologists interested in veterinary and zoonotic diseases.

The majority of animal related coronaviruses infect the intestinal tract and are transmitted by a fecal-oral route.

2. Farm animals:

Coronaviruses infect domesticated birds. this type of coronavirus, causes avian infectious bronchitis. The virus is of concern to the poultry industry because of the high mortality from infection, its rapid spread, and affect on production. The virus affects both meat production and egg production and causes substantial economic loss. In chickens, infectious bronchitis virus targets not only the respiratory tract but also the urogenital tract. The virus can spread to different organs throughout the chicken. The virus is transmitted by aersol and food contaminated by feces. Different vaccines against IBV exist and have helped to limit the spread of the virus and its variants. Infectious bronchitis virus is one of a number of strains of the species Avian coronavirus. Another strain of avian coronavirus is turkey coronavirus (TCV) which causes enteritis in turkeys.

Coronaviruses also affect other branches of animal husbandry such as pig farming and the cattle raising. Swine acute diarrhea syndrome coronavirus (SADS-CoV), which is related to bat coronavirus HKU2, causes diarrhea in pigs. Porcine epidemic diarrhea virus (PEDV) is a coronavirus that has recently emerged and similarly causes diarrhea in pigs. Transmissible gastroenteritis virus (TGEV), which is a member of the species Alpha coronavirus 1, is another coronavirus that causes diarrhea in young pigs. In the cattle industry bovine coronavirus (BCV), which is a member of the species Betacoronavirus 1 and related to HCoV-OC43, is responsible for severe profuse enteritis in young calves.

3. Domestic pets:

Cats, dogs, and ferrets can be infected by Coronaviruses, There are two forms of feline coronavirus that infect cats, 1. Feline enteric coronavirus is a pathogen of minor clinical significance, but spontaneous mutation of this virus can result in feline infectious peritonitis (FIP), a disease with high mortality. For the dogs There are two different coronaviruses. Canine coronavirus (CCoV), which is a member of the species Alphacoronavirus 1, causes mild gastrointestinal disease. Canine respiratory coronavirus (CRCoV), which is a member of the species Betacoronavirus 1, cause respiratory disease. Similarly, For the ferrets there are two types of coronavirus. Ferret enteric coronavirus causes a gastrointestinal syndrome known as epizootic catarrhal enteritis (ECE), and a more lethal systemic version of the virus (like FIP in cats) known as ferret systemic coronavirus (FSC).

4. How to protect yourself:

A number of vaccines using different methods are under development for different human coronaviruses, but at this moment there are no vaccines or antiviral drugs to prevent or treat human coronavirus infections. Treatment is only supportive. A number of antivirial targets have been identified such as viral proteases, polymerases, and entry proteins. Drugs are in development which target these proteins and the different steps of viral replication.The same thing goes for animal coronaviruses which there are no antiviral drugs to treat. Vaccines are available for IBV, TGEV, and Canine CoV, although their effectiveness is limited.

I.3. Summary:

In this chapter we have covered an overview of the various types of the skin cancer that we will be dealing with in the third chapter. Also, we have seen definition, description and history of the second disease that we will be classifying which is Covid-19.

CHAPTER II

Classification Algorithms using Machine and Deep Learning.

II.1. Introduction:

Machine Learning is a sub-area of artificial intelligence, whereby the term refers to the ability of (information and technology) IT systems to independently find solutions to problems by recognizing patterns in databases. In other words: Machine Learning enables IT systems to recognize patterns on the basis of existing algorithms and data sets and to develop adequate solution concepts. Therefore, in Machine Learning, artificial knowledge is generated on the basis of experience. In order to enable the software to independently generate solutions, the prior action of people is necessary. For example, the required algorithms and data must be fed into the systems in advance and the respective analysis rules for the recognition of patterns in the data stock must be defined. Once these two steps have been completed, the system can perform the following tasks by Machine Learning:

- Finding, extracting and summarizing relevant data
- Making predictions based on the analysis data
- Calculating probabilities for specific results
- Adapting to certain developments autonomously
- Optimizing processes based on recognized patterns

II.1.1. Machine Learning: How it works

In a way, Machine Learning works in a similar way to human learning. For example, if a child is shown images with specific objects on them, they can learn to identify and differentiate between them. Machine Learning works in the same way: Through data input and certain commands, the computer is enabled to "learn" to identify certain objects (persons, objects, etc.) and to distinguish between them. For this purpose, the software is supplied with data and trained. For instance, the programmer can tell the system that a particular object is a human being ("human") and another object is not a human being ("no human"). The software receives continuous feedback from the programmer. These feedback signals are used by the algorithm to adapt and optimize the model. With each new data set fed into the system, the model is further optimized so that it can clearly distinguish between "humans" and "non-humans" in the end. But Machine Learning means much more than just distinguishing between two classes. Using the KUKA table tennis robot as an example, you can see how a machine scans the complex tendencies and the playing style of its opponent, adapts to them and even makes a world champion sweat this way.

II.1.2. Advantages of Machine Learning

Machine Learning undoubtedly helps people to work more creatively and efficiently. Basically, you too can delegate quite complex or monotonous work to the computer through Machine Learning - starting with scanning, saving and filing paper documents such as invoices up to organizing and editing images. In addition to these rather simple tasks, self-learning machines can also perform complex tasks. These include, for example, the recognition of error patterns. This is a major advantage, especially in areas such as the manufacturing industry: the industry relies on continuous and error-free production. While even experts often cannot be sure where and by which correlation a production error in a plant fleet arises, Machine Learning offers the possibility to identify the error early - this saves downtimes and money. Self-learning programs are now also used in the medical field. In the future, after "consuming" huge amounts of data (medical publications, studies, etc.), apps will be able to warn a in case his doctor wants to prescribe a drug that he cannot tolerate. This "knowledge" also means that the app can propose alternative options which for example also take into account the genetic requirements of the respective patient.

II.1.3. Methods used in Machine Learning

In Machine Learning, statistical and mathematical methods are used to learn from data sets. Dozens of different methods exist for this, whereby a general distinction can be made between two systems, namely symbolic approaches on the one hand and sub-symbolic approaches on the other. While symbolic systems are, for example, propositional systems in which the knowledge content, i.e. the induced rules and the examples are explicitly represented, sub-symbolic systems are artificial neuronal networks. These work on the principle of the human brain, whereby the knowledge contents are implicitly represented.

II.1.4. Types of Machine Learning

Basically, algorithms play an important role in Machine Learning: On the one hand, they are responsible for recognizing patterns and on the other hand, they can generate solutions. M.L Algorithms can be divided into different categories:

1. **Supervised learning:** In the course of monitored learning, example models are defined in advance. In order to ensure an adequate allocation of the information to the respective model groups of the algorithms, these then have to be specified. In other words, the system learns on the basis of given input and output pairs. In the course of monitored learning, a programmer, who acts as a kind of teacher, provides the appropriate values for a particular input. The aim is to train the system in the context of successive calculations with different inputs and outputs and to establish connections.

2. **Unsupervised learning:** In unsupervised learning, artificial intelligence learns without predefined target values and without rewards. It is mainly used for learning segmentation (clustering). The machine tries to structure and sort the data entered according to certain characteristics. For example, a machine could (very simply) learn that coins of different colors can be sorted according to the characteristic "color" in order to structure them.
3. **Partially supervised learning:** Partially supervised learning is a combination of supervised and unsupervised learning.
4. **Encouraging learning:** Reinforcing learning - just like Skinner's classic conditioning - is based on rewards and punishments. The algorithm is taught by a positive or negative interaction which reaction to a certain situation should take place.
5. **Active learning:** Within the framework of active learning, an algorithm is given the opportunity to query results for specific input data on the basis of pre-defined questions that are considered significant. Usually, the algorithm itself selects questions with high relevance.

   In general, the data basis can be either offline or online, depending on the corresponding system. In addition, it can be available only once or repeatedly for Machine Learning. Another distinguishing feature is either the staggered development of the input and output pairs or their simultaneous presence. On the basis of this aspect, a distinction is made between so-called sequential learning and so-called batch learning.

II.1.5. Some Machine Learning algorithms:

Here is the list of commonly used machine learning algorithms. These algorithms can be applied to almost any data problem:

1) Linear Regression
2) Logistic Regression
3) Decision Tree
4) SVM
5) Naive Bayes
6) kNN
7) K-Means
8) Random Forest
9) Dimensionality Reduction Algorithms
10) Gradient Boosting algorithms:
    a) XGBoost
    b) LightGBM
    c) CatBoos
    d) GBM

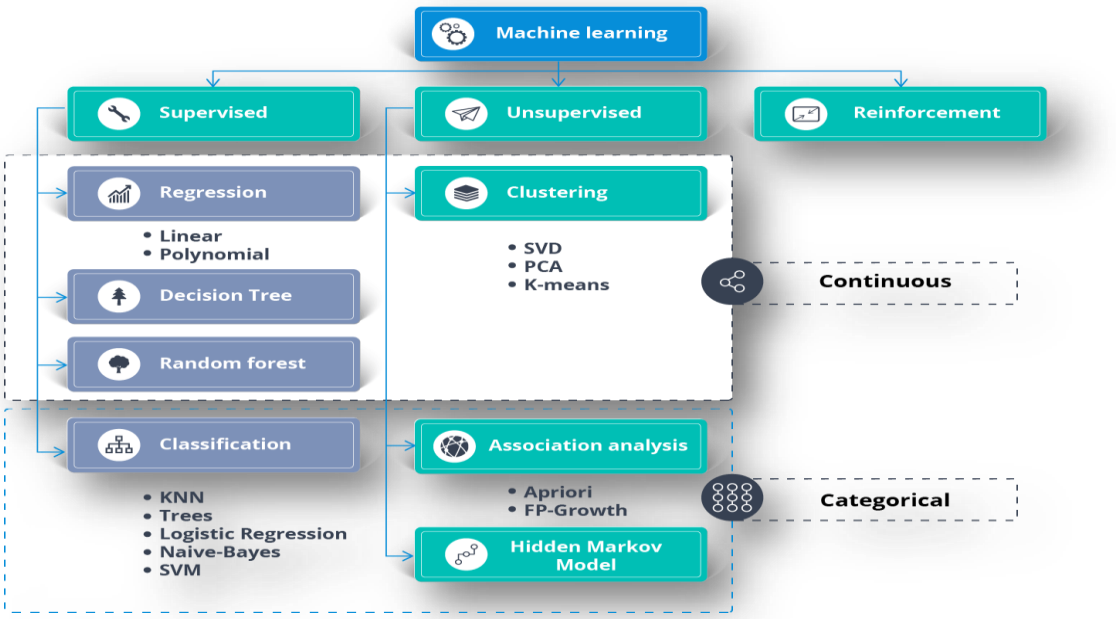

Figure II.1: *Some machine learning based on type and use cases.* [20]

II.1.6. Machine Learning and its most popular applications

Machine Learning is applied at Netflix and Amazon as well as for Facebook's face recognition. For you as a user, Machine Learning is for example reflected in the possibility of tagging people on uploaded images. In fact, Facebook has the largest face database in the world. The data fed by users into the social network is used by Facebook to optimize and train Machine Learning systems in terms of visual recognition. Another application of Machine Learning that is now firmly integrated into everyday life is the automatic detection of spam that is integrated into almost all e-mail programs. Within the scope of spam detection, the data contained in the e-mails is analyzed and categorized. The "spam" and "non-spam" patterns are used in this respect. If an e-mail is recognized as junk mail, the program learns to identify spam mails even more efficiently. Other areas of application for Machine Learning include search engine ranking, combating cybercrime and preventing computer attacks.

II. 2. Machine Learning Algorithms

In this part, we are going to cover the algorithms used in the implementation of the experiments of chapter 3.

II. 2.1. Support-Vector Machines:

Classifying data is a common task in machine learning. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. In the case of support-vector machines, a data point is viewed as a p-dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a $(p-1)$ −dimensional hyperplane where p is the dimension of the dataset. This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. So we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier it defines is known as a maximum-margin classifier; or equivalently, the perceptron of optimal stability.

support-vector machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. SVM algorithm is a popular machine learning tool that offers solutions for both classification and regression problems. Developed at AT&T Bell Laboratories by Vapnik with colleagues (Boser et al., 1992, Guyon et al., 1993, Vapnik et al., 1997), it presents one of the most robust prediction methods, based on the statistical learning framework or VC theory proposed by Vapnik and Chervonekis (1974) and Vapnik (1982, 1995). Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall.

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N the number of features) that distinctly classifies the data points.

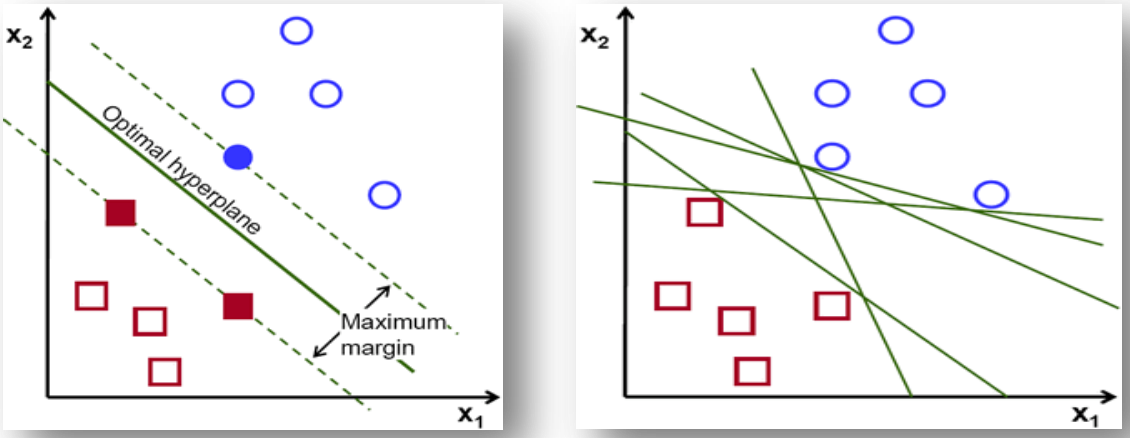

***Figure II.2**: Possible hyperplanes.* [21]

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

1. Hyperplanes and Support Vectors:

Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds

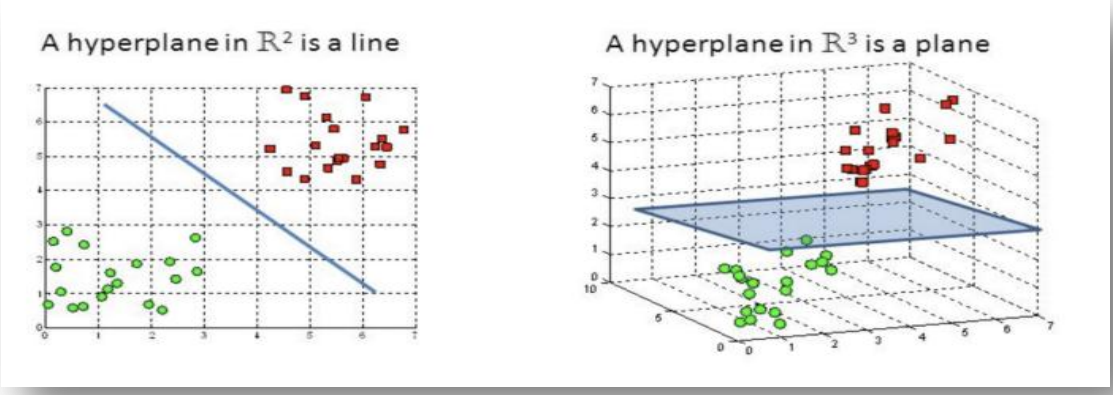

***Figure II.3:** Hyperplanes in 2D and 3D feature space*. [21]

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

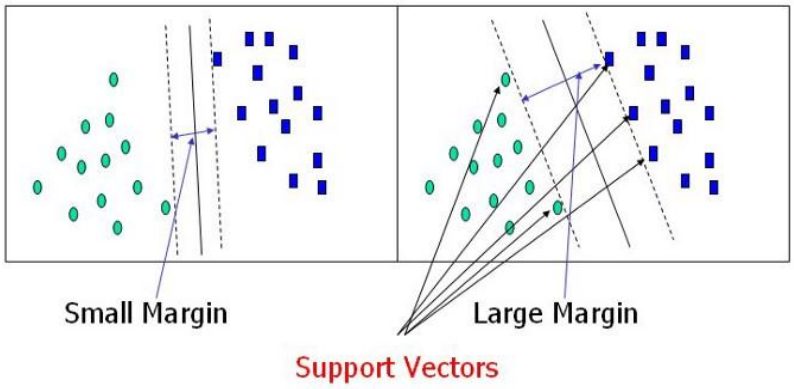

Figure II.4: *Support Vectors.* [22]

1. Large Margin Intuition

In logistic regression, we take the output of the linear function and squash the value within the range of [0,1] using the sigmoid function. If the squashed value is greater than a threshold value (0.5) we assign it a label 1, else we assign it a label 0. In SVM, we take the output of the linear function and if that output is greater than 1, we identify it with one class and if the output is -1, we identify is with another class. Since the threshold values are changed to 1 and -1 in SVM, we obtain this reinforcement range of values ([-1,1]) which acts as margin.

2. Hard-margin:

If the training data is linearly separable, we can select two parallel hyperplanes that separate the two classes of data, so that the distance between them is as large as possible. The region bounded by these two hyperplanes is called the "margin", and the maximum-margin hyperplane is the hyperplane that lies halfway between them. With a normalized or standardized dataset, these hyperplanes can be described by the equations $\vec{w} \cdot \vec{x} - b = 1$ (anything on or above this boundary is of one class, with label 1)
$\vec{w} \cdot \vec{x} - b = -1$ (anything on or below this boundary is of the other class, with label −1). Where ($\vec{w}$ is the normal vector of the desired hyperplane, $\vec{x}$ is a vector that needs to be classified and b is a scaler. Both b and $\vec{w}$ are to be defined by the algorithm).

Geometrically, the distance between these two hyperplanes is$\frac{2}{||\vec{w}||}$, so to maximize the distance between the planes we want to minimize$||\vec{w}||$. The distance is computed using the distance from a point to a plane

equation 13]. We also have to prevent data points from falling into the margin, we add the following constraint: for each i either

$$\vec{w}.\vec{x}_i - b \geq 1, if\ y_i = 1 \quad \text{(II. 1)}$$

(Where $\vec{x}_i$ Is a sample from the dataset)

Or

$$\vec{w}.\vec{x}_i - b \leq -1, if\ y_i = -1 \quad \text{(II. 2)}$$

These constraints state that each data point must lie on the correct side of the margin.

This can be rewritten as

$$y_i(\vec{w}.\overrightarrow{x_i} - b) \geq 1, for\ all\ 1 \leq i \leq n. \quad \text{(II. 3)}$$

We can put this together to get the optimization problem:

Minimize $||\vec{w}||$ subject to $y_i(\vec{w}.\overrightarrow{x_i} - b) \geq 1, for\ all\ 1 \leq i \leq n.$

The $\vec{w}$ and b that solve this problem determine our classifier,$\vec{x} \mapsto sgn(\vec{w}.\vec{x} - b)$.

An important consequence of this geometric description is that the max-margin hyperplane is completely determined by those $\vec{x}_i$ that lie nearest to it. These $\vec{x}_i$ are called support vectors.

3. Soft-Margin:

To extend SVM to cases in which the data are not linearly separable, we introduce the hinge loss function [23],

$$max\ (\ 0\ ,1\ -\ y_i(\vec{w}.\overrightarrow{x_i} - b)\ ) \quad \text{(II. 4)}$$

Note that y_i is the i th target (i.e., in this case, 1 or −1), and $\vec{w}.\overrightarrow{x_i} - b$ is the i-th output.

This function is zero if the constraint in (1) is satisfied, in other words, if $\overrightarrow{x_i}$ lies on the correct side of the margin. For data on the wrong side of the margin, the function's value is proportional to the distance from the margin.

4. Multiclass SVM

Multiclass SVM aims to assign labels to instances by using support-vector machines, where the labels are drawn from a finite set of several elements. The dominant approach for doing so is to reduce the single multiclass problem into multiple binary classification problems. Common methods for such reduction include: Building binary classifiers that distinguish between one of the labels and the rest (one-versus-all) or between every pair of classes (one-versus-one). Classification of new instances for the one-versus-all case is done by a winner-takes-all strategy, in which the classifier with the highest-output function assigns the class (it is important that the output functions be calibrated to produce comparable scores). For the one-versus-one approach, classification is done by a max-wins voting strategy, in which every classifier assigns the instance to one of the two classes, then the vote for the assigned class is increased by one vote,

and finally the class with the most votes determines the instance classification. Crammer and Singer proposed a multiclass SVM method which casts the multiclass classification problem into a single optimization problem, rather than decomposing it into multiple binary classification problems.

II.2.2 Random Forests (Decision Trees):

1. The Random Forest definition:

RF is based on decision trees. In machine learning decision trees is a technique for creating predictive models. They are called decision trees because the prediction follows several branches of "if… then…" decision splits - similar to the branches of a tree. If we imagine that we start with a sample, which we want to predict a class for, we would start at the bottom of a tree and travel up the trunk until we come to the first split-off branch. This split can be thought of as a feature in machine learning, let's say it would be "age"; we would now make a decision about which branch to follow: "if our sample has an age bigger than 30, continue along the left branch, else continue along the right branch". This we would do until we come to the next branch and repeat the same decision process until there are no more branches before us. This endpoint is called a leaf and in decision trees would represent the final result: a predicted class or value. At each branch, the feature thresholds that best split the (remaining) samples locally is found.

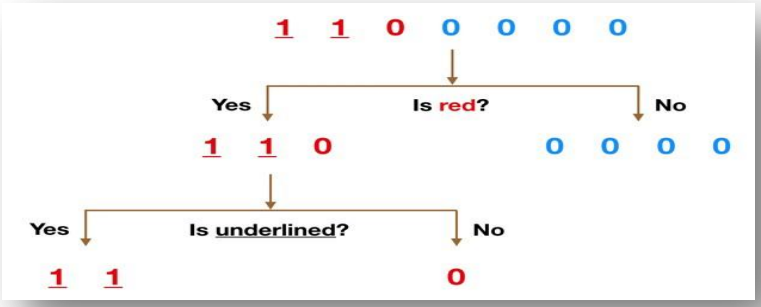

Figure II.5: *Simple Decision Tree Example.* **[24]**

It is probably much easier to understand how a decision tree works through an example. Our dataset consists of the numbers at the top of the precious figure. We have two 1s and five 0s (1s and 0s are our classes) and desire to separate the classes using their features. The features are color (red vs. blue) and whether the observation is underlined or not. So how can we do this?

Color seems like a pretty obvious feature to split by as all but one of the 0s are blue. So we can use the question, "Is it red?" to split our first node. You can think of a node in a tree as the point where the path splits into two observations that meet the criteria go down the Yes branch and ones that don't go down the

No branch. The No branch (the blues) is all 0s now so we are done there, but our Yes branch can still be split further. Now we can use the second feature and ask, "Is it underlined?" to make a second split. The two 1s that are underlined go down the Yes subbranch and the 0 that is not underlined goes down the right subbranch and we are all done. Our decision tree was able to use the two features to split up the data perfectly.

2. The Random Forest Classifier

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction

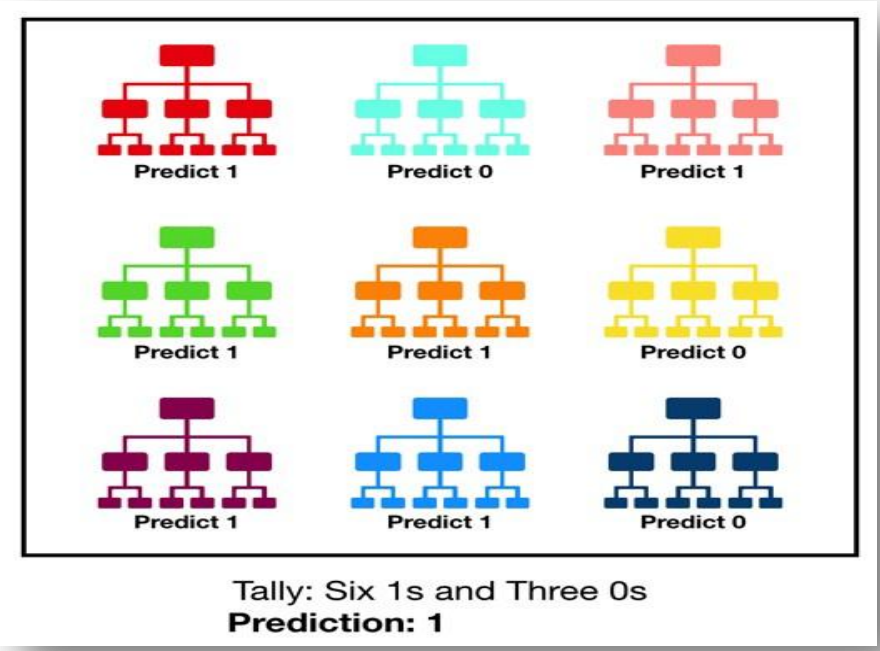

Figure II.6: *Visualization of a Random Forest Model Making a Prediction*. [25]

The fundamental concept behind random forest is a simple but powerful one. In data science speak, the reason that the random forest model works so well because a large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models. The low correlation between models is the key. Just like how investments with low correlations (like stocks and bonds) come together to form a portfolio that is greater than the sum of its parts, uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions**.** The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don't constantly all err in the same direction). While some trees may be wrong, many other trees will be right, so

as a group the trees are able to move in the correct direction. So the prerequisites for random forest to perform well are:

- There needs to be some actual signal in our features so that models built using those features do better than random guessing.
- The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other.
- Bagging (Bootstrap Aggregation).

Decisions trees are very sensitive to the data they are trained on small changes to the training set can result in significantly different tree structures**.** Random forest takes advantage of this by allowing each individual tree to randomly sample from the dataset with replacement, resulting in different trees. This process is known as bagging.

Notice that with bagging we are not subsetting the training data into smaller chunks and training each tree on a different chunk. Rather, if we have a sample of size N, we are still feeding each tree a training set of size N (unless specified otherwise). But instead of the original training data, we take a random sample of size N with replacement. For example, if our training data was [1, 2, 3, 4, 5, 6] then we might give one of our trees the following list [1, 2, 2, 3, 6, 6]. Notice that both lists are of length six and that "2" and "6" are both repeated in the randomly selected training data we give to our tree (because we sample with replacement).

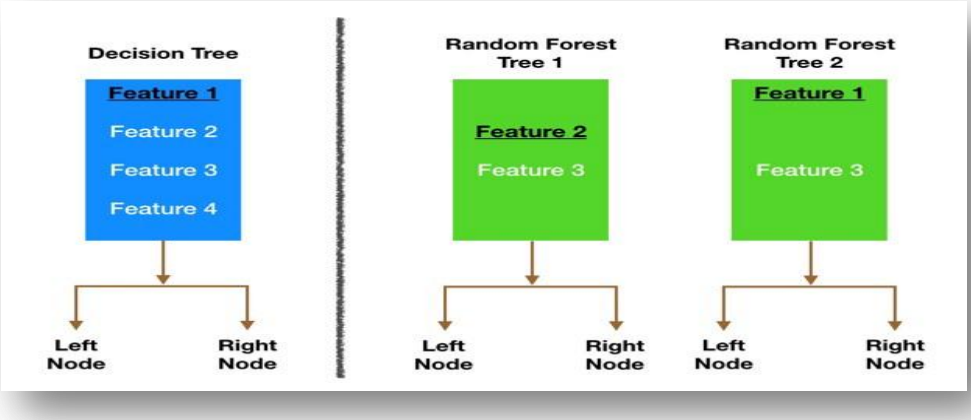

Figure II.7: *Node splitting in a random forest model is based on a random subset of features for each tree. [26]*

3. Feature Randomness:

In a normal decision tree, when it is time to split a node, we consider every possible feature and pick the one that produces the most separation between the observations in the left node vs. those in the right node. In contrast, each tree in a random forest can pick only from a random subset of features. This forces even more variation amongst the trees in the model and ultimately results in lower correlation across trees and more diversification.

Let's go through a visual example in the picture in the pervious page, the traditional decision tree (in blue) can select from all four features when deciding how to split the node. It decides to go with Feature 1 (black and underlined) as it splits the data into groups that are as separated as possible.

Now let's take a look at our random forest. We will just examine two of the forest's trees in this example. When we check out random forest Tree 1, we find that it can only consider Features 2 and 3 (selected randomly) for its node splitting decision. We know from our traditional decision tree (in blue) that Feature 1 is the best feature for splitting, but Tree 1 cannot see Feature 1 so it is forced to go with Feature 2 (black and underlined). Tree 2, on the other hand, can only see Features 1 and 3 so it is able to pick Feature 1.So in our random forest, we end up with trees that are not only trained on different sets of data (thanks to bagging) but also use different features to make decisions. this creates uncorrelated trees that buffer and protect each other from their errors.

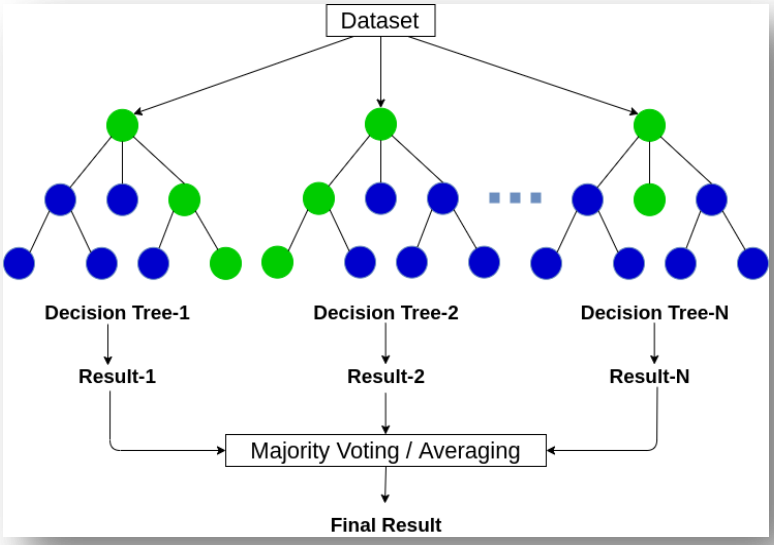

***Figure II.8**: Simplified random forest algorithm.* **[27]**

II.3. Deep Learning Algorithms:

II.3.1. Convolutional neural network:

A convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics. They have applications in image and video recognition, recommender systems, image classification, medical image analysis, natural language processing and financial time series. CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "fully-connectedness" of these networks makes them prone to overfitting data. Typical ways of regularization include adding some form of magnitude measurement of weights to the loss function. CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns. Therefore, on the scale of connectedness and complexity, CNNs are on the lower extreme. Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

3. Definition:

The name “convolutional neural network” indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of a linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

4. Components of a Convolutional Neural Network :

A. **Input layer:** this layer takes the input image on its original form “RGB” or gray scale this layer doesn’t change anything on the image. The image passes directly to the next layer.

B. **Convolutional Layer:** this layer applies the convolution operation on the input image by sliding a kernel (filter) on the image and computing the dot product with the region overlapped with the kernel.

C. Pooling layer: this layer is similar to the convolutional layer it slides a kernel on the resulted image from the convolutional layer and down sample it by taking the maximum or the average value of the region overlapped by the kernel.

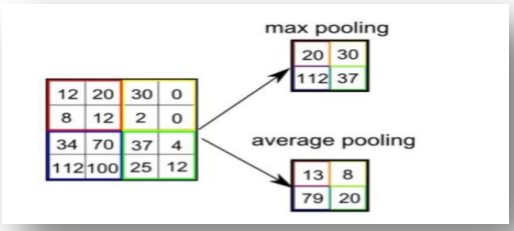

Figure II.9: *4x4 Explanation of Max and Average Pooling. [28]*

D. Fully Connected Layer :

In a typical CNN, fully-connected layers are usually placed toward the end of the architecture. Adding a Fully-Connected layer is a cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. Usually the output of the fully-connected layer uses "softmax" or "sigmoid" activation functions:

- **Softmax:** this activation function is usually used in the case of multiclass problems. It produces a discrete probability for each neuron in the output layer. The softmax formula can be given like:

$$f_i(\vec{a}) = \frac{e^{a_i}}{\sum_k e^{a_k}} \qquad \text{(II. 5)}$$

- **Sigmoid:** this activation function bounds the output of the neurons between 1 and 0 this function is used in the case of binary classification because of its binary output. sigmoid formula is given like :

$$f(x) = \frac{1}{1+e^{-(x)}} \qquad \text{(II. 6)}$$

5. CNN training:

The training process of CNNs is depending on the back propagation algorithm very similar to artificial neural networks. The back propagation algorithm starts from the output layer. At first step, it initializes all the weights and the kernels of the fully connected layer and the convolutional layers randomly. The next step is doing a prediction using the random weights and computing the loss function, after that, the derivative of the loss function is computed with respect to the input in

order to find the error for the weights, we use the error in order to update the weights. Another thing to compute is the error that should be propagated the pervious layer, to compute the propagated error the algorithm do the derivation of the loss function with respect to the weights of the output layer. This process is repeated on each layer for all samples of the dataset. This is the main algorithm used in training of deep learning models.

6. Hyper parameters:

The training algorithm we have mentioned previously may face some problems like running to overfitting or falling to local minimum there are many problems that needs some optimization in order to get good models.

E. Optimizer:

The optimizer is an algorithm that has huge effect on the training algorithm while it decides how and when to update the weights or not, how compute the derivatives and some many others features. But the most important feature is that optimizer guarantee that the loss function will approach its global minimum. The most used optimizer is the Adam optimizer.

F. Dropout:

Overfitting is one of the major problems that faces deep learning models, overfitting is the case where the model has high performance in the training set but it is not good at other samples. Dropout is a way that help the model to not fall in overfitting by randomly select some weights and stop them from updating in some training iterations.

G.Batch size:

The batch size is number of samples that are processed by the model without updating the weights.

H.Batch normalization:

Batch normalization is the operation of making the output of a layer in the same range in order to prevent from dominant features, this operation also helps in avoiding overfitting.

II.4. Summary:

In this chapter we have seen, a general introduction about the machine learning field its types use cases and some applications, after that we have described some of the algorithms we will be using in the next chapter that are (Support vector Machine and Random Forest Classifiers). At the end we have seen another type of artificial intelligence which is the deep learning Convolutional neural networks specifically and we have discussed its characteristics, structure and the hyper parameters that controls the performance of the artificial models.

CHAPTER III

Experiments & Results:

III.1 Introduction:

This chapter describes and compares the performance of the three classification methods. We applied it on Covid-19 Dataset as a first application of binary classification and on the Skin cancer for multi-class classification as second application. We have implemented both applications using three classification methods which are SVM, Random Forest and Convolutional Neural Networks. The implementation process passes through training phase using the training dataset, after that comes the test phase to evaluate the obtained classifiers on a validation set that contains some test images. We have used the python language in order to create the algorithms of our classifiers. In addition to that, we have changed some parameters in order to see their effects on performance of our models.

III.2 Evaluation metrics:

When evaluating clinical test in medical field, there are four common main evaluative metrics that are sensitivity, specificity, accuracy and loss function.

- **Accuracy:** The accuracy is represented as the number of items correctly identified as either truly positive or truly negative out of the total number of items [29].

(III.1)

$$\frac{TruePositive(TP) + TrueNegative(TN)}{TruePositive(TP) + TrueNegative(TN) + FalsePositive(FP) + FalseNegative(FN)}$$

- **Recall (also called Sensitivity or True Positive Rate):** Recall is represented as the number of items correctly identified as positive out of the total actual positive [29].
- **Specificity (True Negative Rate):** Specificity is represented as the number of items correctly identified as negative out of the total actual negatives [29].

$$\text{Specificity} = \frac{TruePositive}{TruePositive + FalseNegative} \quad \text{(III.2)}$$

$$Sensitivity = \frac{TrueNegative}{TrueNegative + FalsePositive} \quad \text{(III.3)}$$

In multi-class problems, since each confusion matrix pools all observations labeled with a class other than target class as the negative class, this approach leads to an increase in the number of true negatives, especially if there are many classes. This problem effects the specificity of the model. For multiclass application we will use the following metrics:

- **The micro average:**

The micro average has its name from the fact that it pools the performance over the smallest possible unit (i.e. over all samples) [29]:

$$\mathrm{P}_{micro} = \frac{\sum_{i=1}^{|G|} TP_i}{\sum_{i=1}^{|G|} TP_i + FP_i} \quad \text{(III.4)}$$

$$\mathrm{R}_{micro} = \frac{\sum_{i=1}^{|G|} TP_i}{\sum_{i=1}^{|G|} TP_i + FN_i} \quad \text{(III.5)}$$

(Where G is the number of classes, T is true, F is false, P is positive and N is negative)

The micro-averaged precision, $\mathrm{P}_{\mathrm{micro}}$ and recall $\mathrm{R}_{\mathrm{micro}}$, give rise to the micro F1-score [29]:

$$\mathrm{F1}_{micro} = 2\frac{\mathrm{P}_{micro} \cdot R_{micro}}{\mathrm{P}_{micro} + \mathrm{R}_{micro}} \quad \text{(III.6)}$$

- **The macro average:**

The macro average has its name from the fact that it averages over larger groups, namely over the performance for individual classes rather than observations [29]:

The macro-averaged precision and recall give rise to the macro F1-score:

$$P_{macro} = \frac{1}{|G|}\sum_{i=1}^{|G|}\frac{TP_i}{TP_i + FP_i} \quad \text{(III.7)}$$

$$R_{macro} = \frac{1}{|G|}\sum_{i=1}^{|G|}\frac{TP_i}{TPi + FNi} \quad \text{(III.8)}$$

$$\mathrm{F1}_{macro} = 2\frac{\mathrm{P}_{macro} \cdot R_{macro}}{\mathrm{P}_{macro} + \mathrm{R}_{macro}} \quad \text{(III.9)}$$

If $\mathrm{F1}_{\mathrm{macro}}$ has a large value, this indicates that a classifier performs well for each individual class. The macro-average is therefore more suitable for data with an imbalanced class distribution.

Loss Function:

There are so many loss functions that can be used, but we only focus on the following functions:

- **Mean Squared Error (MSE):**

MSE is a measure of the average of the squares between the actual observations and those predicted as described by [30]:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2 \quad \text{(III.10)}$$

In the case of neural networks, the predicted values $\hat{y}_i$ are the outputs of the final layer and the true values y_i the output of the modelled function, N being the number of training samples.

- **Binary Cross Entropy**:

Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label, which is given as follow [31]:

$$CrossEntropy\ -\frac{1}{N}\sum_{1}^{N} Y_i log\hat{Y}_i - (1 - Y_i)\log(1 - \hat{Y}_i) \quad \text{(III.11)}$$

- **Categorical Cross Entropy:**

Categorical cross-entropy is a loss function that is used in multi-class classification tasks. The categorical cross-entropy loss function calculates the loss of an example by computing the following sum [32]:

$$Categorical\ Cross\ Entropy = -\sum_{i=1}^{outpui\ Size} Y_i . log\hat{Y}_i \quad \text{(III.12)}$$

III.3 Tools:

All of our programming codes were written in python with Keras library for the implementation of deep learning models and scikit learn library for machine learning algorithms. Also we have used Kaggle platform in order to train our machine and deep learning models.

1. **Python:** is a high-level, interpreted, interactive and object-oriented functional scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages [33].

   A. Python is interpreted: Python is processed at runtime by the interpreter. We do not need to compile our program before executing it. This is similar to PERL and PHP.

   B. Python is Interactive: we can actually sit at a Python prompt and interact with the interpreter directly to write our programs.

   C. Python is Object-Oriented: Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

   D. Python is a Beginner's Language: Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

2. **TensorFlow:** is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications [34].

3. **Keras:** is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and maintainer is François Chollet, a Google engineer. Chollet also is the author of the XCeption deep neural network model [35].

4. **Scikit-learn**: is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, *k*-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy [36].

5. **Kaggle**: is a subsidiary of Google LLC, is an online community of data scientists and machine learning practitioners. Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists and machine learning

engineers, and enter competitions to solve data science challenges. Kaggle got its start in 2010 by offering machine learning competitions and now also offers a public data platform, a cloud-based workbench for data science, and Artificial Intelligence education. Its key personnel were Anthony Goldbloom and Jeremy Howard. Nicholas Gruen was founding chair succeeded by Max Levchin [37]. Equity was raised in 2011 valuing the company at $25 million. On 8 March 2017, Google announced that they were acquiring Kaggle.

III.4 Application1-Binary Classification: Covid-19 identification

1. **Procedure:** first, we have started by implementing the machine learning models (SVM and RF) using sklearn library and we have fitted them to our dataset. After that, we have tested our machine learning models on validation subset, using the test results we evaluated our models. At second step, we have implemented deep learning models (CNNs) with Keras library using tensorflow backend, we compiled our model and we have trained it on training dataset. After the training is finished, we have tested it on the validation subset. After that, we have augmented our deep learning model, by adding more layers to it and we tested the new models as well.
2. **Data handling:** Before we build our models, we need to setup the data that we are going to use properly. After creating the notebook on Kaggle, we added the training dataset to it. The data we are going to use composed of two folders Covid-19 and normal folders [38 - 42] to get the images that are inside these images we have used the 'OS' library in order to surf through the operating system folders, and get the all the paths of the images. After that, we have used the OpenCv library to load the images from the data folders, while we are doing this step, we make another copy of the data in order to flatten the images, so we will able to use them on the machine learning algorithms as vectors.

3. **Experiment:**

A. Covid-19 identification based on SVM:

First, we need to import the Scikitlearn library that contains the machine learning algorithms that we are going to use.

```
from sklearn.svm import SVC
```

In the previous piece of code, we have imported the class that we are going to use to build our machine learning models (Kernel SVM in the “SVC”), after that, we have built our SVM classifier as follow:

```
modelSVM = SVC(C = 3,gamma = 0.05,kernel = 'poly')
modelSVM.fit(xtrain,ytrain)
```

```
SVC(C=3, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.05, kernel='poly',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

After using the cross validation split test class to split the dataset into training and validation sets. We have used the previous code in order to build our SVM model. In the first line the parameter C is the regularization parameter and gamma is the Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.

In the third line, we have fitted our model to the training set using the fit method. When the training is finished, we have done some predictions on the validation set to test the performance and the accuracy of the SVM model on our dataset using the confusion matrix.

```
predictionSVM = modelSVM.predict( xtest)
accuarcySVM = modelSVM.score(xtest,ytest)
print(accuarcySVM)
print(predictionSVM[0])
```

In this code, we have used the score method to measure the accuracy of the SVM. The next step is computing the sensitivity and the specificity using the confusion matrix.

After that we test our model by predict the test data and computing the accuracy.

```
predictionSVM = modelSVM.predict(xtest)
accuarcySVM = modelSVM.score(xtest,ytest)
print("SVM accuracy with poly Kernel:" ,str(accuarcySVM) )
```

```
SVM accuracy with poly Kernel: 0.9834710743801653
```

Figure III.1: *poly SVM accuracy for covid-19 dataset*

```
cm = confusion_matrix(label,predicted_class_indices)
print(cm)
sensitivitySVM = cm[0,0]/(cm[0,0]+cm[1,0])
print('Sensitivity of poly SVM: ', sensitivitySVM )
specificitySVM = cm[1,1]/(cm[0,1]+cm[1,1])
print('Specificity of poly SVM: ', specificitySVM)
```

```
[[64  1]
 [ 1 55]]
Sensitivity of poly SVM:  0.9486153846153847
Specificity of poly SVM:  0.9821428571428571
```

Figure III.2: *Poly SVM confusion matrix, Sensitivity and Specificity for covid-19 dataset*

Now we have changed the kernel to the default Rbf kerenl and we have kept the same value of gamma. To change the kernel to RBF we just delet the kernel argument from the SVC class. After that we have fitted our model to the same data as before and we have tested it's accuracy, sensitivity and specificity.

```
predictionSVM = modelSVM.predict(xtest)
accuarcySVM = modelSVM.score(xtest,ytest)
print("SVM accuracy with RBF Kernel:" ,str(accuarcySVM) )
```

```
SVM accuracy with RBF Kernel: 0.884297521
```

Figure III.3: *RBF SVM accuracy for covid-19 dataset*

```
SVMcm = confusion_matrix(ytest,predictionSVM)
print('Confusion Matrix : \n', SVMcm)

sensitivitySVM = SVMcm[0,0]/(SVMcm[0,0]+SVMcm[1,0])
print('Sensitivity for RBF kernel SVM is  : ', sensitivitySVM )

specificitySVM = SVMcm[1,1]/(SVMcm[0,1]+SVMcm[1,1])
print('Specificity for RBF kernel SVM is : ', specificitySVM)
```

```
Confusion Matrix :
 [[58  7]
 [ 7 49]]
Sensitivity for RBF kernel SVM is  :  0.8923076923076924
Specificity for RBF kernel SVM is :  0.875
```

Figure III.4: *RBF SVM confusion matrix, Sensitivity and Specificity for covid-19 dataset*

We can see clearly that the poly kernel SVM surpassed the RBF kernel SVM in this application.

B. Covide-19 identification based on Random Forest:

First, we import all needed modules (Random forest classifier form scikitlearn package). At this part we have used the same data we have used at the SVM, so no data handling is needed. We implement our random forest model with 1000 estimator, we also initiated the model to use all available cores of the CPU by assigning -1 to the number of jobs.

```
modelRandomForest = RandomForestClassifier(n_estimators= 1000,n_jobs=-1)
modelRandomForest.fit(xtrain,ytrain)
```

After training the random forest model we have tested the model and we have computed the evaluation metrics using the following codes:

```
prediction = modelRandomForest.predict( xtest)
accuarcy = modelRandomForest.score(xtest,ytest)
```

In the previous code we have done some predictions on the test set in the first line and in the second line we have used the score method which gives the accuracy of the model.

The accuracy of the model is giving in the following figure:

```
predictionRMF = modelRandomForest.predict( xtest)
accuarcyRMF = modelRandomForest.score(xtest,ytest)
print( "the accuracy of the random forest model is  " + str(accuarcyRMF))
```

```
the accuracy of the random forest model is  0.9752066115702479
```

Figure III.5: *Random forest accuracy for covid-19 dataset*

The following picture gives the confusion matrix and the sensitivity and the specificity of the random forest model:

```
RMFcm = confusion_matrix(ytest,predictionRMF)
print('Confusion Matrix : \n', RMFcm)

sensitivityRMF = RMFcm[0,0]/(RMFcm[0,0]+RMFcm[1,0])
print('Sensitivity for Random forest is  : ', sensitivityRMF )

specificityRMF = RMFcm[1,1]/(RMFcm[0,1]+RMFcm[1,1])
print('Specificity for Random forest is : ', specificityRMF)
```

```
Confusion Matrix :
 [[63  2]
 [ 1 55]]
Sensitivity for Random forest is  :  0.984375
Specificity for Random forest is :  0.9649122807017544
```

Figure III.6: *Random forest confusion matrix, Sensitivity and Specificity for covid-19 dataset*

For covid-19 dataset all of the results we have obtained from the previous model are ordered in the following table:

Table III.1: *SVM and RF results for covid-19*

	Accuracy	Sensitivity	Specificity
Poly SVM	98%	0.98	0.98
RBF SVM	88%	0.89	0.87
Random Forest	97%	0.98	0.96

C. Covide-19 identification based on CNN:

First, we need to import the Keras library and the classes that are needed to implement the main parts of the Convolutional Neural Networks.

```
from tensorflow.keras.layers import Flatten,Dense,Dropout,BatchNormalization,Conv2D,MaxPooling2
D
from tensorflow.keras.models import Sequential
```

In the first line, we have imported Conv2D, MaxPooling2D, Batch Normalization, Dense and Flatten classes from Keras layers package which uses tensorFlow backend. After that we have implemented the main of CNN parts using the Keras API:

- First, we have to create a sequential class object to initialize a tensor flow computational graph:
- **Convolution Step:** After creating the sequential object, we have to add operations that we want to

```
model=Sequential()
```

perform to our tensors and the first operation is the convolution. In this line of code we have taken the sequential object which control the computational graph and we add to it a convolutional operation with 64 kernels that are of size (2, 2). We have specified that the size if the input tensor is (224, 224,3), which indicates the resolution of the image, the activation function that we have used is the “relu”.

```
model3.add(Conv2D(64,(3,3),input_shape=(224, 224, 3),activation='relu',name = "conv1",padding ="same"))
```

- **Pooling step:** The same as we did with the convolutional layer we added MaxPool2D object to the computational graph the Sequential add method. Here we are adding Max pooling layer with kernel of size (2, 2) and the default stride which is equal to the pool size.

```
model.add(MaxPooling2D(pool_size=(2, 2)))
```

- After adding max pooling layer, we added batch normalization layer.

```
model3.add(BatchNormalization())
```

The batch normalization layer is mainly used for feature scaling and it is also helpful for regularization and avoiding over fitting. As we have added Batch Normalization layer to perform regularization and the feature scaling, we have added dropout layer to avoid over fitting by randomly selecting weights and stop them from getting updated temporary based on probability argument passed when creating the layer object.

```
model.add(Dropout(0.3))
```

At this point, we have created the feature extraction part of the CNN model, we can make the model deeper and deeper by repeating the previous steps as much as we want. After finishing the feature extraction, we have to change the shape of our tensor from 2D to 1D in order to be passed through the fully connected network all we have to do is to add Flatten Layer.

```
model.add(Flatten())
```

In this line of code, we are adding Flatten that is going to convert the shape of the output of the computational graph to 1D tensor. We have converted our tensor to the proper shape and now it is ready to be classified through the fully connected layer, we just have to add that.

- In the first line we have added a Dense layer that contains 256 neurons with “relu” activation function and l2 regularizing algorithm as kernel_regulizer. At the last line, we add output layer with 7 neurons each for a class from the classes in the dataset with “softmax” activation for the categorical classification.

```
model.add(Dense(256,activation='relu',kernel_regularizer=regularizers.l2(0.01)))
model.add(Dropout(0.3))
model.add(Dense(1,activation='sigmoid'))
```

After building our model, we are not ready yet to train it, we have to compile it and assign the suitable optimizer that we are going to use, in this case we are using “Adam” optimizer.

```
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy',Recall()])
```

On the previous code, we have used the “Adam” optimizer, in order to choose the stochastic gradient descent algorithm for the training. For the second argument, we have the categorical cross entropy as loss function, because we have multiple classes in our dataset. In addition, we have used the “accuracy and Recall” metrics to track the performance of our model. After that, we have the early stopping operation; in the training process of our model, we have used the early stopping method, in order to stop the model from training if it has reached out the global minimum of the loss function before the last epoch.

```
early=EarlyStopping(monitor='accuracy',patience=4,mode='auto')
reduce_lr = ReduceLROnPlateau(monitor='accuracy', factor=0.5, patience=2, verbose=
1,cooldown=0, mode='auto',min_delta=0.0001, min_lr=0)
```

In the previous piece of code, we have the first line that adding the early stopping operation with patience argument of 4, which means the training will wait for 4 batches if the loss is the same it stops the training process. In the second line we have the reduce learning rate operation that update the learning rate through the training. The monitor argument indicate what quantity to be monitored we have the accuracy in our case, the second argument is the patience it’s integer that indicates the number of epochs to update the learning rate from the last it has been updated and we have used 2 epochs to be wait for. The factor argument is the factor by which the learning is reduced. Verbose is an integer that may take two value 1or 0 and it is used in order to display update message or no. Cooldown: number of epochs to wait before resuming normal operation after the learning rate has been reduced. Min_delta is threshold for measuring the new optimum, to only focus on significant changes. min_lr is the minimum value the learning rate may have. At this point, we have finished all the setup of our model and we can fit it to the data set, but before we do that, we wanted to pass the dataset through the data generator to increase the number of images in the training set.

The summary of our 4 conv layers model was like follow:

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1 (Conv2D)               (None, 224, 224, 64)      1792
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 112, 112, 64)      0
_________________________________________________________________
batch_normalization_1 (Batch (None, 112, 112, 64)      256
_________________________________________________________________
conv2 (Conv2D)               (None, 112, 112, 128)     73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 56, 56, 128)       0
_________________________________________________________________
conv3 (Conv2D)               (None, 56, 56, 128)       147584
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 28, 28, 128)       0
_________________________________________________________________
batch_normalization_2 (Batch (None, 28, 28, 128)       512
_________________________________________________________________
conv4 (Conv2D)               (None, 28, 28, 128)       147584
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 14, 14, 128)       0
_________________________________________________________________
batch_normalization_3 (Batch (None, 14, 14, 128)       512
_________________________________________________________________
flatten_1 (Flatten)          (None, 25088)             0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               6422784
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 257
=================================================================
Total params: 6,795,137
Trainable params: 6,794,497
Non-trainable params: 640
_________________________________________________________________
```

Figure III.7: *4 conv layers model summary*

```
data_generator=ImageDataGenerator(rotation_range=20, # rotate the image 20 degrees
                                  width_shift_range=0.10, # Shift the pic width by a
max of 5%
                                  height_shift_range=0.10, # Shift the pic height by
a max of 5%
                                  rescale=1/255, # Rescale the image by normalzing it.
                                  shear_range=0.1, # Shear means cutting away part of
the image (max 10%)
                                  zoom_range=0.1, # Zoom in by 10% max
                                  horizontal_flip=True,
                                  vertical_flip=True,
                                  fill_mode='nearest')
```

In the previous code we have initialized the image data generator in order to be used in the data augmentation step.After that we fitted our model to the generated dataset as follow:

```
#Training our CNN
model.fit(trainx,testx,epochs=50,batch_size=90,validation_data=(trainy, testy),cal
lbacks=[early,reduce_lr])
```

To fit our model, we have used the fit method on our model, also we have passed a list of arguments to our fit function to work properly and they are:

- Trainx and it refers to the training set we have obtained from the data generator.
- Testx and this refer to the labels corresponding to the training data.
- Epochs indicates the number of how many iterations through the dataset the training will pass
- Batch_size since we are using mini batch stochastic gradient decent the batch size is the of samples number the model should iterate through before updating the weights.
- Validation_data is the data on which we are going to test the performance of our model.
- Callbacks is the argument that calls the early stopping and reducing the learning rate operations. After the training is finished, we check the accuracy and the validation accuracy from the history property of the model.

```
#Visualizing Training and Validation Accuracy
p.figure(figsize=(15,5))
loss=pd.DataFrame(model.history.history)
loss=loss[['accuracy','val_accuracy']]
loss.plot()
```

In the code above, loss is the panda's variable holding the history of the model, but we just want to plot the accuracy and the validation accuracy, in the third line we saved only the two columns that we need. Last, we have plotted the history as graphs. The next step is to try some predictions with our model and check the confusion matrix in order to be able to calculate the evaluation metrics.

➢ Experiment 1: CNN with 4 convolutional layers

In this experiment, we have trained the model that we have created earlier on the data that has been generated by the image data generator, after the training is finished the model history was like follow:

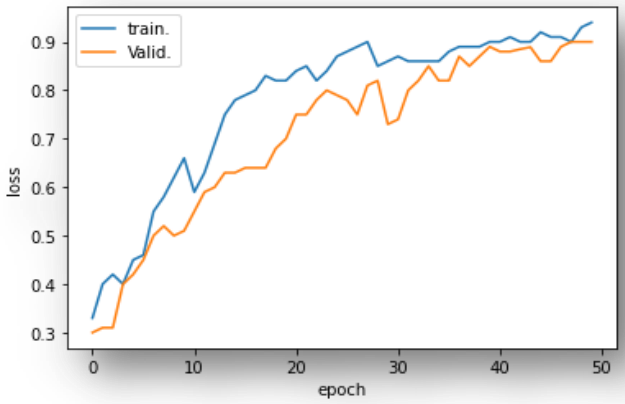

Figure III.8: *The Acc. of 4 conv layers model for Covid-19 identification*

After doing some predictions on the validation set the confusion matrix that we have obtained is given in the following figure :

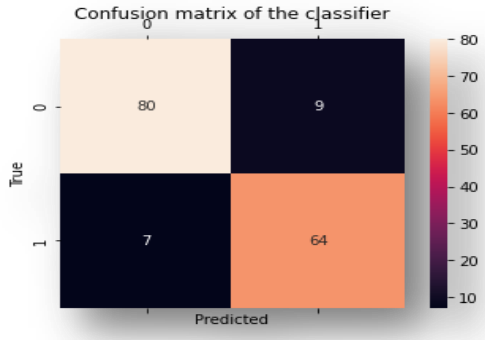

Figure III.9: *Confusion matrix of the 4 conv layers model*

After getting the confusion matrix we have used its values to compute the evaluative metrics (sensitivity and specificity) and the results are summarized in the table below:

Table III.2: *Results of 4 conv layers model for Covid-19 identification*

	Accuracy	Sensitivity	Specificity
4 conv layers	90%	91%	88%

➢ **Experiment 2: CNN with 5 convolutional Layers**

In this Experiment, we kept the same algorithm just we have added a convolutional layer with the same features. In addition to that we have used increased the dropout probability to avoid overfitting. After the training was finished the have plotted the accuracy history and the results were like follow:

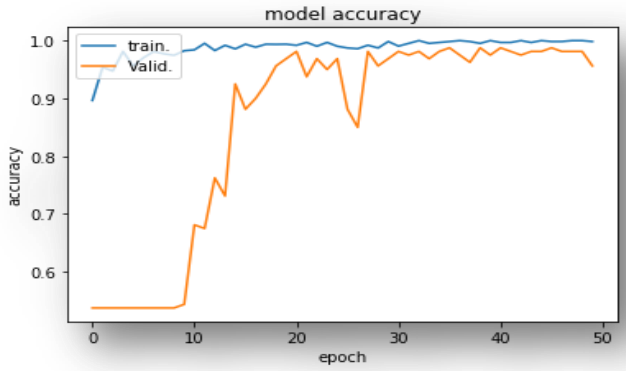

Figure III.10: *The Accuracy of 5 conv layers model for Covid-19 identification*

After testing the model with the validation data we have got the following confusion matrix and based on that we have computed the evaluation metrics:

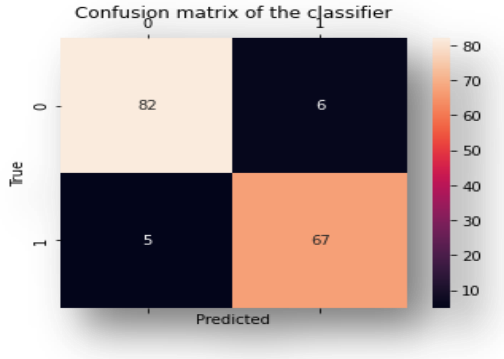

Figure III.11: *Confusion matrix of the 5 conv layers model*

The evaluative metrics are calculated based on the confusion matrix and the results are in the following table:

Table III.3: *Results of 5 conv layers model for Covid-19 identification*

	Accuracy	Sensitivity	Specificity
5 conv layers	**93%**	**94%**	**91%**

➢ Experiment 3: CNN with 6 Convolutional Layers

In this step, we repeat the same process we did in the previous experiment with 5 convolutional layers, we add another convolutional layer and we increase the dropout probability to avoid overfitting. The results were like follow:

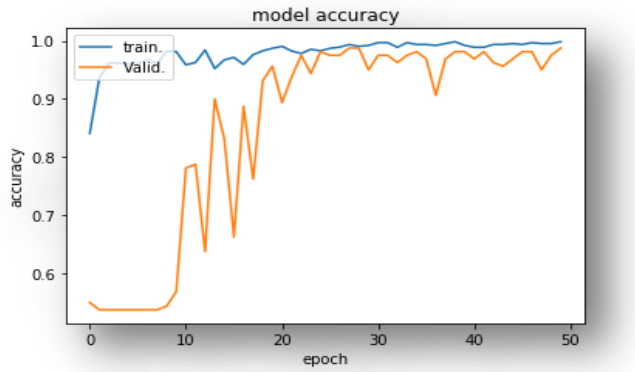

Figure III.12: *The Accuracy of 6 Conv layers model for Covid-19 identification*

The confusion matrix associated with this model is given bellow:

4

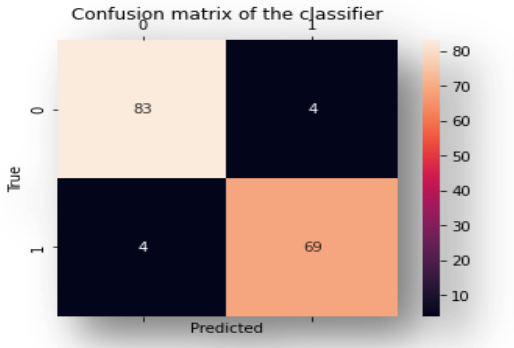

Figure III.13: *Confusion matrix of the 6 conv layers model*

The evaluation metrics for 6 convolutional layers model are given in the table below :

Table III.4: Results of 6 conv layers model for covid-19 identification

	Accuracy	Sensitivity	Specificity
6 conv layers	95%	95%	94%

III.5 Application 2: Skin Cancer- Multiclass Classification

1. **Procedure:** at this application we brought the models that we have built at the first application and we have done some changes on them. After that we have trained the models on the Skin cancer dataset. When the training is finished, we have tesedt the models on the validation subset as we have done previously in the first application.
2. **Data handling:** Before to start building any model we should have lean the way to use the dataset and how to extract the images, first we have created our python notebook to the Kaggle account and we have linked the dataset we wanted to use from Kaggle. This set consists of 2357 images of malignant and benign oncological diseases, which were formed from The International Skin Imaging Collaboration (ISIC). All images were sorted according to the classification taken with ISIC, and all subsets were divided into the same number of images, with the exception of melanomas and moles, whose images are slightly dominant. The data set contains the following diseases:

    - actinic keratosis
    - basal cell carcinoma
    - dermatofibroma
    - melanoma
    - nevus
    - pigmented benign keratosis
    - seborrheic keratosis
    - squamous cell carcinoma
    - vascular lesion

We can see that our dataset contains 9 categories of images but the two categories of tumors are of the same type of two other categories the y have the same medications and diagnosis so we will merge them and we will get 7 classes at the end. We will use openCv and the'os' libraries in order to extract the images from the directories we flatten them in order to use them with SVM and Random Forest classifiers. For CNN we will image data generator in order to augment the original data that we have extracted from the dataset [35].

3. Experiments:

A. Skin Cancer classification based on SVM: In this step we have used the SVM models that we used in the previous application one with 'poly' kernel and the other using 'RBF' kernel. We just fitted them to the skin cancer dataset. After that we compute the evaluative metrics and we obtained the following results:

```
print ("the accuracy for the SVM is "+ str(accuarcySVM))
```

```
the accuracy for the SVM is 0.7539936102236422
```

Figure III.14: RBF SVM acc for Skin cancer dataset

The overall performance of the model has given the following results:

Table III.5: *RBF SVM results for Skin cancer dataset*

	Accuracy	F1 micro	F1 macro
RBF SVM	**75%**	**74%**	**69%**

We can see that $f1_{macro}$ score is little bit less $f1_{micro}$ and that's due to the fact that some classes have poor performance (precision: 40% and recall: 67%) and it contribute with 1/7 in the result.

After we have fitted our poly SVM model we have got the following accuracy:

```
predictionSVM = modelSVM.predict( xtest)
accuarcySVM = modelSVM.score(xtest,ytest)
print(accuarcySVM)
print(predictionSVM[0])
```

```
0.6721246006389776
```

Figure III.15: *poly SVM accuracy for Skin cancer dataset*

Because we got a low accuracy, we didn't compute the evaluation metrics, but clearly, they will be low.

B. Skin Cancer classification based on Random Forest algorithm:

As we did with SVM we started by importing the Random Forest Classifier class from sklearn library:

```
from sklearn.ensemble import RandomForestClassifier
```

After that, we jumped directly to create our Random Forest model by creating an object of the class we have imported previously, and then we fitted the newly created object to the sale data set that we have implemented SVM on.

```
modelRandomForest = RandomForestClassifier(n_estimators= 1000,n_jobs=-1)
modelRandomForest.fit(xtrain,ytrain)
```

In this code we have created a random forest model named model RandomForest with n_estimators equal to 1000, this means that our random forest is constructed by 1000 decision trees. n_jobs = -1 means that when we are training our model all the cores of the CPU we are using will participate in the training. After the training has finished wee test our model the same way we did with SVM to compute the accuracy and the sensitivity of this model:

```
predictionRMF = modelRandomForest.predict( xtest)
accuarcyRMF = modelRandomForest.score(xtest,ytest)
print(accuarcyRMF)
print(predictionRMF[0])
```

After that, we have called the confusion matrix on the predictions obtained by the random forest, to check the sensitivity and the specificity of the trained random forest model:

The results for the random forest model were as follow:

- For the accuracy we didn't get a high value, we got 73% accuracy:

```
the of the random forest model is  0.7332268370607029
```

Figure III.16: Random Forest accuracy for Skin cancer dataset

- We have tried to increase the number of estimators to 1500 and more but there was no big difference in the accuracy so we continue with this model. The F1 micro and the F1 macro scores were as follow:

Table III.6: Random forest evaluation metrics results.

	Accuracy	F1 micro	F1 macro
Random Forest	73%	72%	68%

The F1 macro of the Random forest was somehow low, but the F1 micro was a little higher.

Table III.7: SVM and Random Forest results for Skin cancer dataset

	Accuracy	F1 micro	F1 macro
RBF SVM	75%	0.74	0.69
Random Forest	73%	0.72	0. 68

We can see that the F1 macro is less than F1 micro and that is because the performance on some classes is good but on the other classes the performance is poor.

C. Skin Cancer classification based on CNN:

In this part we have made some changes on the models that we have used earlier for Covid-19 classification in order to make it compatible with the data set we have. We have changed the input shape to match the images that we going to train them on and we have changed the output shape to the number of the classes we have in the dataset. Also, we have changed the loss function to categorical cross entropy because we have multiclass dataset as we have said earlier.

➢ **Experiment 1: CNN with 4 convolutional layers:**

At this step we trained our 4 conv layers model we have from the first application on the Skin cancer dataset we made the proper changes and its performance measure has given this result:

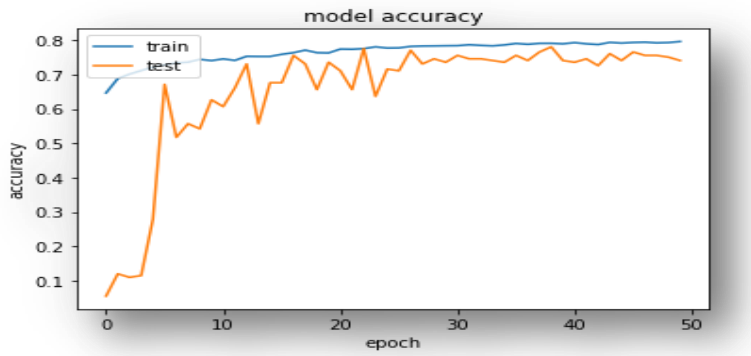

Figure III.17: *The Acc4 conv layers model for Skin cancer classification*

After testing the model on the validation set, we have applied the confusion matrix on the results the validation predictions: All the obtained results are arranged in the following table:

Table III.8: *Results of 4 conv layers model for Skin cancer classification*

	Accuracy	**F1 micro**	**F1 micro**
4 conv layers	77%	0.76	0.72

➢ **Experiment 2 CNN with 5 convolutional Layers:** In this Experiment, we kept the same algorithm just we have added a convolutional layer with the same features. In addition to that we have used increased the dropout probability to avoid overfitting.

The accuracy history of the model is in the following figure:

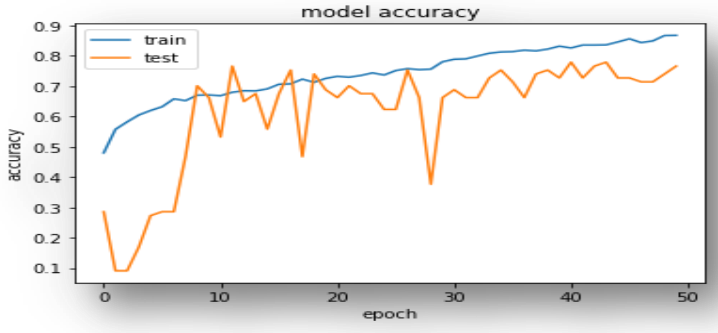

Figure III.18: *The accuracy of 5 conv layers model for Skin cancer classification*

After calculating the evaluative metrics, we have organized them in the table given underneath:

Table III. 9: *Results of 5 conv layers model for Skin cancer classification*

	accuracy	F1 micro	F1 macro
5 conv layers	**80%**	**0.77**	**0.74**

- **Experiment 3: CNN with 6 convolutional Layers:**

In this step, we repeat the same process we did in the previous experiment with 5 convolutional layers, we add another convolutional layer and we increase the dropout probability to avoid overfitting. The results were like follow*:*

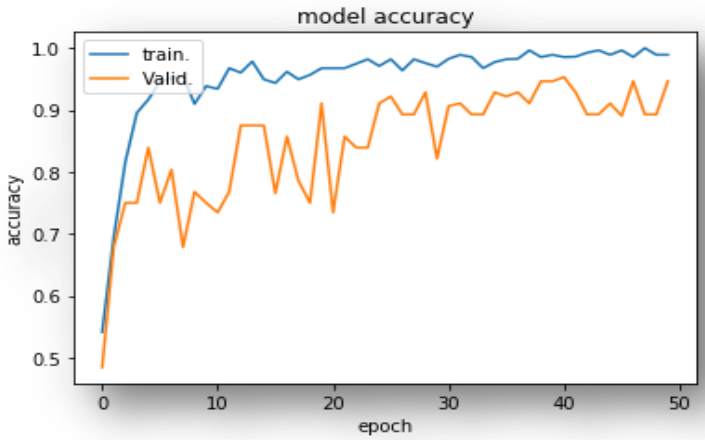

Figure III.19: *The accuracy of 6 conv layers model for Skin cancer classification*

The results of the 6 conv layers models are in the following table:

Table III. 10: *Results of 6 conv layers model for Skin cancer classification*

	Accuracy	Sensitivity	Specificity
6 conv layers	**90%**	**0.87**	**0.83**

III.6 Discussion:

1. **Application1:** In first application, the results of Poly SVM and Random forest algorithms are quite high with compare to deep learning in terms of specificity and sensitivity, but this doesn't mean that machine learning algorithms are better than CNN when it comes to binary classification. These results were like this only because the dataset of the covid-19 CT is not large enough so the deep learning models couldn't get the decision boundary between the classes as correct as possible, we could have tried to increase the epochs number but that the training has been stopped by the early stopping callback due the stability of the accuracy. In another hand RBF SVM didn't have that high results and that's due to the distribution of the dataset, Poly kernel could fit the data much better than the RBF. Also, in deep learning models increasing the depth of the models has increased the performance of the models from 90% accuracy to 95%. If we want to improve the performance of the deep learning models we should transfer the pre-trained models of Keras such as ResNet or ImageNet models using transfer learning approach.
2. **Application2:** In the second Application, we can see that the results of machine learning algorithms (SVM and RF) are small with compare to the Deep learning models in terms of accuracy and F1 Scores. Generally, in multiclass classification problems $F1_{macro}$ score is smaller than the $F1_{micro}$ score and that's due to the high number of probabilities. Deep learning models are much stronger than machine learning models in the multiclass problems with large amount of data. But in our application, we could notice that increasing the number of convolutional layers had a huge effect on the performance, it has raised the accuracy from 77% to 90 %. Increasing the number of convolutional layers to 7 may result a better performance but we stopped at 6 layers, because 6 layers has been the most efficient deep learning model in the first application. Also in this application we can improve the performance by using the transfer learning, there is another approach that can be used to improve the performance is the decomposition-composition, in this approach we divide the class into sub classes we apply the classification to the sub classes than we recompose the original classes based on some criteria. For machine learning models we may improve the performance by applying some data preprocessing such as applying some image processing filters to the dataset images.

GENERAL CONCLUSION

The main contributions of this work can be summarized as: Firstly, classifying chest CT scan images to covid-19 and normal classes using some machine learning models (SVM and Random Forest classifiers) and using Convolutional Neural Networks. Secondly, applying the same developed algorithms in order to classify images of skin cancers into its different classes. Finally, experimental results were done in order to demonstrate the capabilities and robustness of each classifier.

Accordingly, in the first application, we have first collected the data then implemented the classification methods on the Covid-19 dataset. We have built an SVM classifier using two kernels and we got an accuracy of 97% for poly SVM and 88% for RBF SVM with (sensitivity and specificity) of (0.98, 0.98) and (0.89, 0.87) respectively. After that we have trained the random forest classifier, we have 1000 estimators in this model and we have achieved and accuracy of 97%, Specificity of 0.98 and 0.96 for specificity. Also we have built CNN models, we have started with 4 convolutional than increased the number of convolutional layers till 6 layers as we were increasing the number of layers the performance increased also from 90% with 4 convolutional layers to 95% for 6 convolutional layers model with sensitivity and specificity of 0.95 and 0.94 respectively.

In the second application, we have used the same models we had from the previous application and we have tried to fit them to the skin cancer dataset. After we did the testing our SVM classifier has achieved 77% accuracy with RBF kernel and 73% accuracy for Random Forest classifier. For the CNN models we have achieved 90% accuracy with 6 convolutional layers and loss of 25%.

From this two application, we can conclude that CNN can be used for both multiclass and binary classification problem if there is enough data for the training, whereas SVM and Random Forest can be used also for both problem but the CNN models would surpass their performance in the presence of enough training data. Also, SVM and Random Forest may have a high performance even if the training data is not large.

Transfer learning can be used in order to improve the performance of the deep learning model. In an other hand, to improve SVM and Random Forest classifiers a proper data preprocessing is needed.

As further work, transfer learning may be used to improve the performance of the classification models, because it can transfer and adapt the pretrained models to the desired application without the need of big amount of data. Also doing image preprocessing may help machine learning models as SVM to get better performance than we what have got earlier.

REFERENCES

[1] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7127800/

[2] https://www.mayoclinic.org/diseases-conditions/actinic-keratosis (Accessed: August 2020)

[3] https://www.mayoclinic.org/diseases-conditions/basal-cell-carcinoma (Accessed: August 2020)

[4] https://www.mayoclinic.org/diseases-conditions/seborrheic-keratosis (Accessed: August 2020)

[5] https://www.medicalnewstoday.com/articles (Accessed: August 2020)

[6] https://en.wikipedia.org/wiki/Melanocytic_nevus (Accessed: August 2020)

[7] https://www.aafp.org/afp/1998/0215/p765.html (Accessed: August 2020)

[8] https://www.mayoclinic.org/diseases-conditions/melanoma/(Accessed: August 2020)

[9] Ali Mahmoud. *"How to Cope with Corona Virus"*. EC Microbiology Vol. 16, N4,pp. 01-02, 2020.

[10] Vinod Kumar Goyal and Chandrika Sharma. *The novel coronavirus 2019: A naturally occurring disaster or a biological weapon against humanity: A critical review of tracing the origin of novel coronavirus 2019.* Journal of Entomology and Zoology Studies; Vol.8, No 2.pp. 01-05, 2020.

[11] Goldsmith CS, Tatti KM, Ksiazek TG, and al. *Ultrastructural characterization of SARS coronavirus*. Emerging infectious diseases. Vol.10, No 2, pp.320-326, 2004. doi:10.3201/eid1002.030913.

[12] https://www.researchgate.net/figure/A-conceptual-cross-sectional-model-of-a-Coronavirus-Adapted-from_fig2_340362730

[13] https://www.sciencedirect.com/science/article/pii/S2090123220300540

[14] Woo, Patrick C. Y.; Huang, Yi; Lau, Susanna K. P.; Yuen, Kwok-Yung. "*Coronavirus Genomics and Bioinformatics Analysis"*. *Viruses.* Vol.*2*, No. 8,pp. 1804-1820, 2010.

[15] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7122471/

[16] https://www.sciencedirect.com/science/article/pii/S2090123220300540

[17] https://www.newscientist.com/term/coronavirus/

[18] https://www.researchgate.net/figure/Potential-animal-origins-of-human-coronaviruses_fig1_338934614

[19] https://www.biophysics.org/blog/coronavirus-structure-vaccine-and-therapy-development

[20] Fehr A.R., Perlman S. *Coronaviruses: An Overview of Their Replication and Pathogenesis*. Methods in Molecular Biology, Vol. 1282. Humana Press, New York, NY, 2015. https://doi.org/10.1007/978-1-4939-2438-7_1

[21] https://www.edureka.co/blog/machine-learning-algorithms/(Accessed: August 2020).

[22] T.Dhiliphan Rajkumar , L. Manish Kumar , N. Akhila , P. SaiKeerthana. *Performance Analysis Of Machine Learning Techniques To Predict Diabetes Mellitus*. Vol. 29, No. 9, pp. 6366-6373, 2020

[23] https://www.ques10.com/p/41200/support-vector-machine-1/ (Accessed: August 2020).

[24] https://www.kaggle.com/khoongweihao/covid19-xray-dataset-train-test-sets (Accessed: August 2020).

[25] https://astrobites.org/2020/01/30/using-a-random-forest-to-classify-asas-sn-variable-stars/ (Accessed: August 2020).

[26] https://wiki.atlan.com/random-forests/ (Accessed: August 2020).

[27] https://www.techemanate.com/random-forest-a-powerful-ensemble-learning-algorithm/(Accessed: August 2020).

[28] https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/ (Accessed: August 2020).

[29] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 (Accessed: August 2020).

[30] Ian Goodfellow, Yoshua Bengio, Aaron Courville : Book, *Deep Learning*, 2016

[31] http://neuralnetworksanddeeplearning.com/chap3.html (Accessed: August 2020).

[32] A link between Cross-Entropy loss and Policy-Gradient expression," https://medium.com/intro-to-artificial-intelligence/a-link-between-cross-entropy-and-policy-gradient-expression-b2b308511867 (Accessed: August 2020).

[33] https://www.python.org/downloads/release/python-370/ (Accessed: August 2020).

[34] https://www.tensorflow.org/install (Accessed: August 2020).

[35] https://keras.io/getting_started/ (Accessed: August 2020).

[36] https://scikit-learn.org/stable/install.html (Accessed: August 2020).

[37] https://www.kaggle.com/docs (Accessed: August 2020).

[38] https://www.kaggle.com/khoongweihao/covid19-xray-dataset-train-test-sets (Accessed: August 2020).

[39] https://www.kaggle.com/tawsifurrahman/covid19-radiography-database (Accessed: August 2020).

[40] https://www.kaggle.com/nabeelsajid917/covid-19-x-ray-10000-images (Accessed: August 2020).

[41] https://www.kaggle.com/bachrr/covid-chest-xray (Accessed: August 2020).

[42] https://www.kaggle.com/nodoubttome/skin-cancer9-classesisic (Accessed: August 2020).