# DEEP-LEARNING CNN-BASED CAD RISK STRATIFICATION VIA RETINAL FUNDUS IMAGES—EARLY HEART ATTACK DETECTION IN A BLINK OF AN EYE

# CHAPTER 1:
# INTRODUCTION:

Coronary Artery Disease (CAD) is a non-communicable disease without a cure and the most prominent heart disease with the highest morbidity and mortality rate. In 2019, CAD contributed to 85% of cardiovascular diseases resulting from heart attack and stroke concerns and led to 32% of global deaths—three-quarters from low- and middle-income countries [1].

Despite being the leading cause of death globally and estimated to increase the rate to 23 million people in 2030, risk stratification for CAD has remained a challenging endeavor [2]. The highly computed Cardiac imaging methods are expensive, risky, and limited to deeper extensive image quantification. The risk calculators don't comply with all risks, and the comprehensive prognosis via machine learning algorithms is also limited to accuracy and training complexity whenever complex features and numerous data are available for training.

This research study aims to achieve a versatile, cost-effective, and intelligent risk stratification tool that accurately prognoses the development of CAD—within five (5) years—by diagnosing the major CAD risk factors—in a quick means, at high speed and safely (non-invasively and non-radioactive): to lower morbidity and mortality rate from heart attack and stroke cardiovascular diseases (CVDs).

This chapter provides an introduction to the study through the background and context, the motivation and study justification, the project significance, scope and limitation, and the structural outline of the thesis chapters.

## 1.1. Research Background

CAD is a cardiovascular disease that emerges from atherosclerosis—the thickening or hardening of heart blood vessels: when fatty material (atheroma) or plaque buildup inside the wall of heart arteries (coronary arteries) and blocks oxygen-rich blood to the heart muscles [3]. The disease may also develop due to malfunction of the coronary arteries—the problems regarding how the blood vessels because of chronic inflammation or excessive strain from high blood pressure—other factors remained unclear and unconfirmed.

CAD is a disease with high global mortality and morbidity rate due to sudden death associated with heart attack and stroke [4]. It may also cause Angina, Heart rhythm problems, Heart failure, cardiogenic shock, and sudden Cardiac arrest. CAD is a complicated disease since the symptoms develop slowly over time (may take a decade), and they may sometimes differ for everyone—more influenced by sex. Although other people may never experience the symptoms, the common ones are chest pain, shortness of breath, body aches, fainting, nausea, and vomiting.

The disease affiliates diverse risk factors categorized as modifiable and non-modified factors. Modifiable risk factors develop depending on the environment (air pollution and unhealthy working environment), unhealthy lifestyle (poor eating patterns, physical inactivity, smoking, and drug abuse), and diseases or medical conditions (such as high blood pressure, high blood triglycerides, diabetes, cholesterol, obesity, and congenital coronary artery defects)**.** The non-modifiable factors such as age, sex, genetics, and family history are permanent and show cumulative effects over time [5].

CAD has existed since the ancient Egyptian era, though the pathogenesis knowledge was retarded and limited until the beginning of the 20th century [6]. Despite having no cure, CAD management is now viable through modern medication or surgery. Thanks to computing and sensory technologies, the diagnostic approaches have evolved from arteriogram to modern imaging (Computed tomography angiogram, Electrocardiogram (ECG), Echocardiogram, Nuclear imaging, Cardiac MRI, and Catheterization) [7] and risk calculators (such as the Pooled Cohort equations, Framingham, and SCORE) [8].

The emergence of Machine Learning (ML)—a branch of Artificial Intelligence (AI) —with a high capacity to influence effective decision-making without being explicitly programmed has also enhanced CAD prognostic techniques to more accurate and efficient outcomes [9], [10].

### 1.2. Research Problem

Risk stratification for CAD is essential to assess and evaluate the risk of developing a disease by identifying and managing groups of people at risk. With the computed and sensor technologies, Cardiac imaging and risk calculators have played a pivotal role in stratifying CAD risks, where the emergence of artificial intelligence through Machine Learning has further enhanced the stratification process.

However, Cardiac imaging is ineffective in analyzing deeper features of image phenotypes as the methods are limited to cardiac structure and functions [7]—as they are expensive, time-consuming, risky (invasive or radioactive), and susceptible to human errors [11]. Risk calculators are not accurate estimators for situations like pregnancy or early menopause, cholesterol levels management, metabolic syndrome, and autoimmune disease [12]. Machine learning through powerful imaging classification algorithms (such as Naïve Bayes, SVM, Neural Networks, and Decision trees) is also limited to accuracy and training complexity: whenever numerous data or complex features are available for training [13].

With these challenges and limitations, CAD remains a prominent cardiovascular disease (CVD)—a global burden that seriously endangers human health and influences the higher global mortality rate—estimated to increase the rate from 17.9 million in 2019 to 23 million in 2030.

### 1.3. Research Goal, Objectives, and Questions

- **Research Aim**

Given the constraints and challenges regarding stratifying risks of CAD, this study aims to achieve a versatile, cost-effective, and intelligent risk stratification tool: that can prognoses the risk of developing CAD and prevent heart attack and stroke risks—in a quick means—with accuracy, high speed, and safety (non-invasively and non-radioactive).

- **Research Objectives**

The practical orientations to achieve the aim of the study are as follows:

1. Analyze and identify a reliable, non-invasive, and non-radioactive means for achieving the CAD biomarker for highlighting risk factors without extensive time consumption.
2. Develop an intelligence, accurate, and high-speed algorithm to handle numerous unstructured and complex data computations: for quality visual assessments and deeper quantification of cardiac structure and functions—without the human hand.
3. Evaluate the precision, sensitivity, and effectiveness of the model, also the strength and drawbacks of the suggested approaches.

- **Research Questions**

From the aim and objectives, these are the questions that this study is seeking to answer:

1. What is an easy, quick, and safe means of achieving CAD risk factors?

2. What versatile and efficient algorithm can achieve multimodal, sensitive, and accurate readings from convoluted features, patterns, and values of cardiac structure and functions?
3. How precise, sensitive, and effective is the model, and what are the strength and drawbacks of the suggested approach?

### 1.4. Research Approach

CAD risk factors can be non-invasive and quickly manifested in an eye through the retina fundus due to the richness of the retinal convoluted texture features, values, and patterns: that change whenever systemic or neurological diseases present in the body. With fundus imaging, the CAD risk factors can easily and fast be evaluated: by examining the alteration of retinal characteristics and variability of retinal blood vessel diameters [14]–[16].

The Deep-Learning advancement in medical imaging has heightened the CAD risk stratification through Convolution Neural Network (CNN) algorithm. The CNN's capability to process the input data through multiple layers in a two-dimensional image—without focusing on feature extraction has made the prognosis viable with high speed, accuracy, and cost-effectiveness [17]–[19].

In this study, retinal fundus images are proposed as a flexible CAD biomarker to incorporate a supervised deep-learning algorithm—Convoluted Neural Network (CNN): to achieve a versatile, cost-effective, and intelligent tool to stratify CAD risks in a quick means with accuracy, at high speed, and safety (non-invasively and non-radioactive).

### 1.5. Research Significance

The following are the implications and contributions of this study—as it will:

1. Lower the global morbidity and mortality rate influenced by the heart attack and stroke risks emerging from the developed CAD.
2. Establish a quick, safe, cost-effective, and intelligent method for easy stratifying CAD and other heart diseases (CVDs)—at high speed without compromising accuracy.
3. Promote the systemic and neurological diseases studies and eye disease examination from retinal fundus imaging complemented by CNN deep learning.
4. Advance the application of CNN deep learning in the medical field.

## 1.6. Research Scope and Limitation

This research only focuses on preventing heart attack-type of heart diseases: by detecting CAD-type of cardiovascular diseases. Since Deep learning requires a vast amount of data, it was impossible to acquire enough data from a single or same distribution—hence data collection was from various databases: in which attaining quality retinal fundus images were expensive. Concerning ethical AI, the system was not trained and evaluated for all social groups and races since the collected data didn't consist of enough datasets from all geographical areas.

## 1.7. Structural Outline

This section outlines the thesis coverage through the logical placement of the chapters as organized as follows:

Chapter one: has introduced the background and context of the study. It has described the approach, significance, and limitations of the study. Finally, the chapter outlines the structure of the thesis in chapters as follows:

Chapter two: explores the theoretical background and empirical literature on history, risk factors, and prognosis challenges regarding CAD. It justifies the research question and the application of the CNN deep-learning algorithm and retinal photographs in enhancing CAD risk stratification.

Chapter three: details the systematic research design and justifies the proposed methodology and methods for the research study. It vindicates the application of deep-learning CNN algorithms incorporating retina fundus images in model designing for CAD risk stratification.

Chapter four: validates the designed proposed methodology and methods through experimentation. It explores environment setups, analyses the model designing processes, and evaluates the performance through obtained results.

Chapter five: presents and evaluates the analysis findings and discusses the significance—contribution and implication—of the achieved results. It answers the research questions and supports the hypothesis made.

Chapter six: concludes the research study by synthesizing the key findings with the research aims and questions. It discusses the value and contribution of the research and reviews the study limitations as it proposes opportunities for future research.

# CHAPTER 2:

# LITERATURE REVIEW:

## 2.1. Introduction

This chapter explores the theoretical and empirical literature on Coronary Artery Disease (CAD) and its risk stratification. It identifies the gaps among the research communities and justifies the research hypotheses, methodology, and proposed methods—in a critical and broader context.

The chapter details the history of CAD, its risk factors, and the prognosis challenges. It also justifies the application of the CNN deep-learning algorithm and fundus photographs in enhancing CAD risk stratification.

## 2.2. Heart and Coronary Artery Disease (CAD)

### 2.3.1. Anatomy and Physiology of Human Heart

As a hollow muscular organ, fit(s) sized, located between the lungs in the front and middle of the chest, behind and slightly to the left of the breastbone—a heart pumps blood to the body's organs, tissues, and cells through the cardiovascular system (network of arteries and veins).

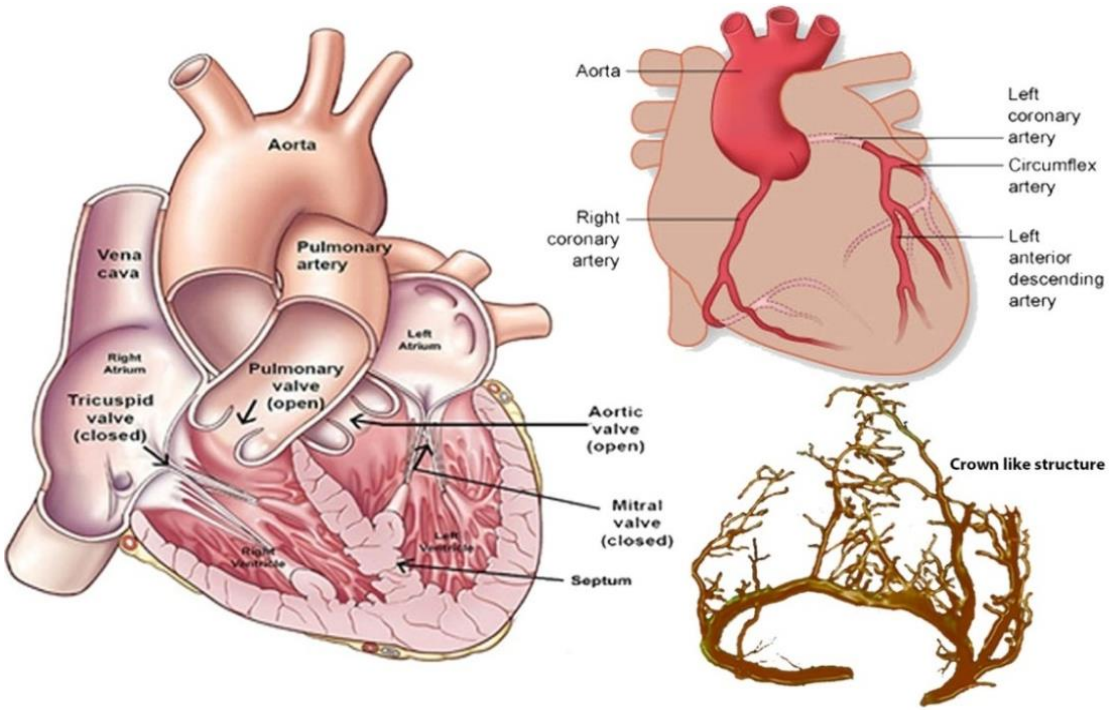


Figure 2-1: Anatomy of Human Heart

Figure 2-1 above shows the heart-inside view—as divided into four chambers: two upper chambers—the upper receiving chamber (right and left atria) and lower pumping chambers (right and left ventricle). The right atrium receives de-oxygenated blood from the various body parts through the veins—superior vena cava and inferior vena cava—and pumps to the right ventricle through the tricuspid valve. The right ventricle pumps the blood through the pulmonary valve to the lungs—to absorb oxygen. Then, the left atrium receives oxygenated blood from the lungs and pumps it to the left ventricle through the mitral valve. Finally, the left ventricle pumps oxygen-rich blood through the aortic valve to the aorta and the rest of the body [20].

### 2.2.1. Coronary Artery Disease (CAD)

The oxygen-rich blood supplied to the heart muscle is through blood vessels called coronary arteries (left and right coronary arteries). The name *coronary* means they encircle the heart in the manner of a crown—as the word *coronary* comes from the Latin word *corona* and the Greek word *koron*—meaning a crown.

When these arteries get damaged or diseased after the fatty material called atheroma or plaque buildup inside the wall of the larger heart arteries—narrows the vessels over time—and blocks the flow of oxygen-rich blood to the heart muscles (myocardium)—CAD emerges. This process is called atherosclerosis [21].

Thus, when the plaque or atheroma breaks off, it forms a blood clot: that can cause a heart attack (myocardial infarction) and stroke.

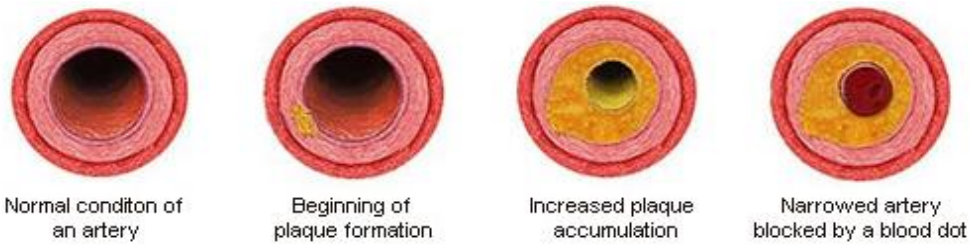


Figure 2-2: Atherosclerosis process. (Adapted from Montgomery country, PA)

CAD, also known as Coronary Heart Disease (CHD), (or) Ischemic Heart Disease (IHD), (or) Myocardial Ischemia—is a non-communicable disease (NCD) without a cure and the most prominent cardiovascular disease (heart disease): with the highest morbidity and mortality rate—since a heart attack and stroke becoming the most issue [3], [4].

In 2019, CAD contributed to approximately 85% of cardiovascular diseases (CVDs) that caused 32% of all global death—through a heart attack (myocardial infarction) and stroke—where the low and middle-income countries (about three-quarters) were the victims. It is also estimated to catalyze the CVDs mortality rate (per year) from 17.9 million in 2019 to 23 million in 2030—thus becoming the concern of United Nations Sustainable Development Goal 3.4 [1].

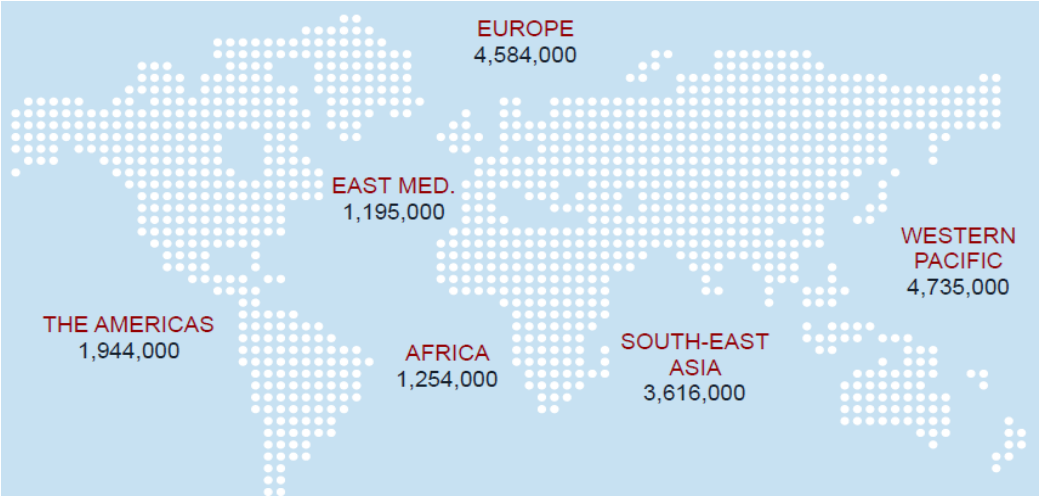


Figure 2-3: 2019 CVDs Global death estimations. (Adapted from World Heart Foundation)

## 2.3. History of Coronary Artery Disease (CAD)

Despite CAD being a modern human disease—described as a 21st-century disease attributed to modern lifestyles, history manifests that concern has existed since the ancient era. At the beginning of the civilization age, Egypt had highly advanced skills in medicine that later influenced the Greek medical tradition. Before the Greek physicians (Herophilus and Erasistratus) performed the first human dissections in Alexandria (Egypt), the ancient Egyptians had managed and treated atherosclerosis among the pharaohs. The conducted investigation through Computed Tomography (CT) scans in 2013: substantiated that 137 mummies from different geographical regions—who lived over 4000 years ago had 34% of atherosclerosis [6].

The pathogenesis knowledge of CAD was retarded, limited, and not elucidated until the beginning of the 20th century. The early discoveries suggested that Leonardo da Vinci (1452–1519) was the earliest to narrate Coronary Artery Disease (CAD). The circulation and heart function discovery of William Harvey (1578–1657) furthered the study.

The Italian anatomist Giovanni Morgagni (1761) described the issue of CAD as atherosclerosis: which means “hardening of the arteries.” However, it was William Heberden (in 1768) who brought angina pectoris to the attention of the medical profession: he coined the term “angina pectoris” from Greek *ankhonē,* which means “strangling” and Latin *pectoris*, meaning “chest”—and many of his aspects remained valid to this era.

After extensive work on angina, William Osler (1849–1919) described angina as more of a syndrome rather than a disease. The 1856 thrombosis study of Rudolf Virchow, the father of pathology, elevated the scientists' serious consideration of CAD implications—and these concepts stayed relevant in the diagnosis and treatment of CAD till the modern medical practices: where the pathogenesis knowledge of coronary heart disease evolved from arteriogram to CT scans and Electrocardiographs (ECG)—the radiology technologies that enhance the treatment and management through modern medication (drug therapy) or surgery.

## 2.4. The Pathology of Coronary Artery Disease (CAD)

Although the Pathogenesis study of CAD explores numerous arteriosclerosis factors, it is more associated with atherosclerosis—which is due to the plaque or atheroma builds up inside the coronary arteries: that narrows and blocks the vessels and leads to an inadequate supply of oxygen-rich blood to the heart muscle (myocardium) [21];

As the blood vessels are responsible for the signals sent by the heart when it demands oxygen-rich blood, the problems with how these vessels work may also emerge the coronary heart disease. Despite these challenges being vague, chronic inflammation and excessive strain from high blood pressure are among the factors that may cause the damage [22], [23].

The problems with how blood vessels work or the formation of a blood clot (when atheroma breaks off) reduce the flow of oxygen-rich blood to the heart (ischemia), which leads to the death of the heart’s muscles. These complications can provoke the emergence of Heart attack (myocardial infarction) and Stroke—which induce sudden human death.

### 2.4.1. The Symptoms and risk factors

While sometimes other people may not experience the symptoms, CAD is associated with numerous symptoms.

These symptoms develop slowly over time (take years to a decade). Although they may differ for everyone (more influenced by sex), most are common and are more associated with angina symptoms such as chest pain, shortness of breath, body ache, fainting, nausea, and vomiting [24].

Varieties of CAD risk factors categorized as modifiable and non-modifiable—are the utmost concern of these symptoms. The non-modifiable factors are permanent and show cumulative effects over time: such as Age, Sex, genetics, and Family history. The modifiable factors develop depending on the environment, lifestyle, and health diseases [5].

- **Age and Sex**

CAD affects people during middle and older ages. Aging causes changes in the heart and blood vessels. The risks of damaging or narrowing the heart arteries increase when people get older. Scientists believe because of estrogen's heart-protective effects, younger women under the age of 45 (perimenopause) are more prevalent in CAD than men. However, at the age greater than 50 (post-menopause), the risks become fare-worse than for men.

- **Family history and Genetics**

Inherited conditions can fault or mutate the genes and cause them to pass on high-risk hereditary diseases from parents to a child. If the diseases like hypertension, hyperlipidemia, diabetes, and obesity can pass to offspring, then the chances of developing CAD are higher.

The family history: the habits shared within the family; or the record of heart or circulatory diseases affecting someone's family members—that if someone's father or brother is diagnosed with CAD before the age of 55; or if someone's mother or sister is diagnosed with the disease before the age of 65, then that individual is also at the risks of developing the disease.

- **Environment and Lifestyle**

Unhealthy lifestyle habits such as poor eating patterns: that consume high amounts of sodium, saturated fats, and refined carbohydrates (sugar) and physical inactivity that increases the risks of developing high blood cholesterol, high blood pressure, diabetes, overweight and obesity; alcohol, drug abuse, and Smoking or long-term exposure to smoke—these are the factors that can raise the conditions of developing Coronary Artery Disease (CAD)**.**

Air pollution as an environmental risk factor can also increase the chance of developing atherosclerosis and high blood pressure—the risk may highly affect older adults, women, and people with diabetes or obesity. The working environment may also be a factor for CAD if someone experiences radiation, toxins, or other hazards; has a lot of stress at work (causes the arteries to tighten), sit for long periods, or works night shifts that affect the sleep.

- **Health diseases and Medical conditions**

The most common medical conditions or health diseases that directly affect the heart—and raise the risk of developing coronary heart disease are: high blood pressure, high blood triglycerides, high blood LDL cholesterol (bad cholesterol), and Congenital coronary artery defects.

The most indirect conditions or diseases consist of autoimmune and inflammatory diseases, chronic kidney disease, diabetes, HIV/AIDS, mental health conditions (including anxiety, depression, and post-traumatic stress disorder-PTSD), metabolic syndrome, overweight**,** obesity**,** and Sleep disorders (sleep apnea or sleep deprivation and deficiency).

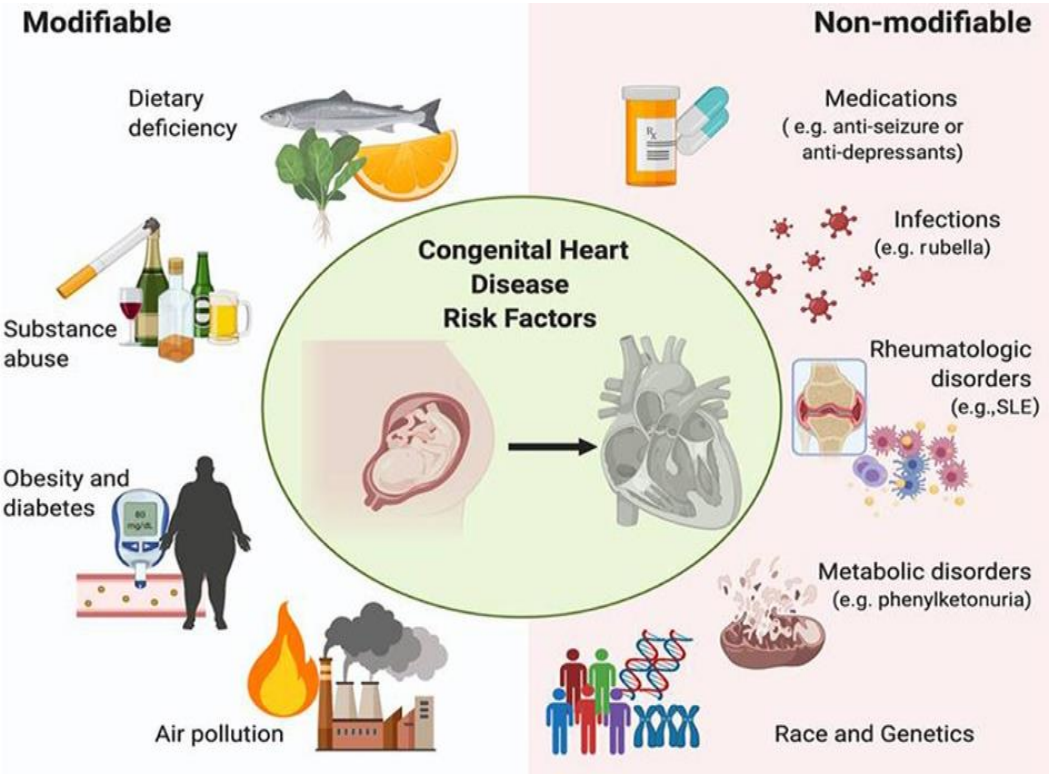


Figure 2-4: CAD Risk factors

### 2.4.2. The Diagnosis and Tests

Traditionally, before conducting any cardiac test, the diagnosis of a heart condition (including coronary artery disease) is based on the view of a cardiologist (heart doctor): who clerks the patient by apprehending the symptoms, observing the signs, examining medical history, reviewing risk factors, and performing a physical examination.

Despite Cardiac Imaging [7] being the advanced method, here are other common CAD diagnostic tests:

- Electrocardiogram (ECG): discovers anomalies in the heart's rhythm and structure through recorded electrical signals or a Holter monitor (a small wearable device that detects heart rhythm abnormalities through capturing ECG in 24 to 72 hours);
- Blood tests (examine the presence of triglycerides, cholesterol, lipoprotein, HbA1ca: a measure of diabetes control, C - reactive protein, and glucose in the blood);
- Echocardiogram (checks the heart rhythm or echoes the blood movement through electrodes and ultrasound technology);
- Computed tomography angiogram (observes 3D pictures of the moving heart and detected coronary arteries blockages from CT and contrast dye);
- Cardiac catheterization (evaluates heart function by inserting a catheter into the coronary arteries)
- Cardiac MRI (detects a damaged tissue or problems with blood flow in coronary arteries from Magnetic Resonance Imaging);
- Nuclear imaging (observes heart images after administering a radioactive tracer);
- A stress test/ Treadmill test /Exercise test (engages a patient in physical activities to determine the heart endurance to the workloads); and
- Chest X-ray.

The diagnosis of CAD may also use the Risk calculators (such as the Pooled Cohort equations, Framingham, and SCORE): which can estimate the risk of having a heart attack in the next ten years by considering age, sex, race, cholesterol levels, blood pressure, and smoking factors [8].


Figure 2-5: Coronary Artery Disease (CAD) Diagnosis

## 2.5. Risk stratification of Coronary Artery Disease (CAD)

### 2.5.1. CAD Risk stratification and its Challenges

CAD Risk stratification is an assessment used to evaluate the risk of developing CAD by identifying and managing groups of people at risk [25]. It is more associated with CAD management—the early diagnosis to reduce the rate of sudden human death: due to myocardial infarction (heart attack)—developed from CAD.

The most challenging endeavors in CAD risk-stratification are the risk-free and inexpensive methods: that can achieve sensitive and accurate readings from convoluted features (like colors and shapes), patterns, and values of cardiac structure and functions [26], [27].

Despite Cardiac imaging playing a pivotal role in CAD diagnostic decision-making through computing and sensor technologies, qualitative visual assessment and crude quantitative measures of cardiac structure and function are their limitations [7]. Their most accurate method, the golden procedure (angiography), has inherent defects—as it is expensive and risky (with a clinical mortality rate of 2% to 3% due to invasiveness) [11]. Moreover, the mapping and interpretation of images are subjective to human expertise (experienced cardiology experts)—hence susceptible to potential errors.

Although the risk calculators incorporate risk factors to predict CAD possibilities, the PINNACLE electronic health record-based cardiovascular registry study indicates that most of the data were available for less than 30% of patients. Moreover, these calculators might not be accurate estimators in pregnancy situations or early menopause, cholesterol levels management using statins, metabolic syndrome, or an inflammatory or autoimmune condition [12], [28], [29].

### 2.5.2. The Emergence of Machine learning

The limitations of the CAD diagnostic tests are the highlight for artificial intelligence, mostly machine learning: wherewith its capacity to influence effective decision-making and predict accurate outcomes—without being explicitly programmed, Machine learning optimizes the tests with affordable, efficient, reliable, and accurate predicted results—through advanced image analysis techniques: that allow deeper quantification of imaging phenotypes and offload cardiologists' extensive iterative works [9], [10].

Machine learning has enhanced CAD diagnosis through powerful supervised classification algorithms (such as Naive Bayes, SVM, Neural Network, and Decision tree): which provide the accurate predictions from trained image models learned from previous experience or adjusted data. These algorithms are applied depending on the number of data and risk factors features [13].

Most of these algorithms are limited to numerous image attributes available for training. As the image attributes increase, the speed and accuracy decrease, also training complexity increases: due to training processing time and hyper-parameter tuning. With the help of deep learning, the trained Artificial Neural Network (ANN) has enhanced the limitations and improved the results.

### 2.5.3. Deep Learning in CAD Risk stratification

Among the machine learning algorithms, Deep learning has quickly become transformative for CAD diagnosis due to its capacity to handle numerous unstructured data and complex image computations: by performing self-learning and analyzing complex algorithms at exceptional speeds without compromising the accuracy [18].

With the ability to mimic human brain behavior by enabling the system to cluster data through multiple computation layers, the algorithm becomes more accurate as it processes more data— as it learns from previous results to refine its ability to make correlations and connections.

Although the initial studies of Deep Learning (DL) were in the 1950s, the overfitting problems (only focused on the specific datasets and unable to process new datasets) made it limited to medicine until the 2000s: when the medical field emerged with the vast training data. The self-advancement of the Deep Learning field has also elevated the application of the algorithms—thanks to computation infrastructures (Powerful computers), enhanced activation functions (like ReLu), and advanced algorithms (like CNN) [30], [31].

The 2012 advancement of Deep Learning in imaging from AlexNet architecture has heightened the CAD risk stratification through advanced Cardiac imaging. With the Convolution Neural Network (CNN) as a supervised deep-learning algorithm, CAD diagnosis and treatment is now viable with high speed, accuracy, and affordable cost. This possibility is due to CNN's capability to process the input data through multiple layers in a two-dimensional image without focusing on feature extraction [17], [32].

CNN is recently considered the more powerful deep-learning algorithm with versatility, effectiveness, and efficiency in detecting CAD and other cardiovascular diseases. However, a simple way of achieving multimodal and high-quality data in a quick, simple, non-invasively, and non-radioactive remained a challenge [26], [27].

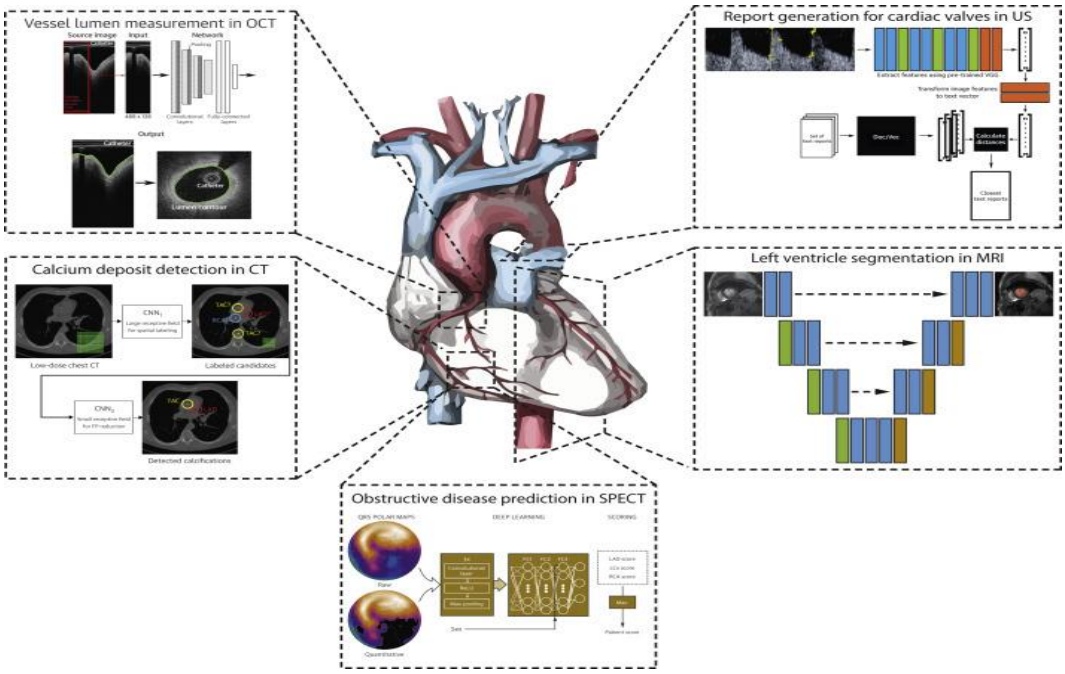


Figure 2-6: Deep Learning algorithms in CAD diagnosis

## 2.6. Retina Fundus in CAD Diagnosis

Since the eye is a distinctive organ that can non-invasive examine the tissues, nerves, and blood vessels, biomedical research has realized not only eye diseases but also the presence of cardiovascular diseases (CVDs) through fundus images: which means CAD diagnosis is also viable through an eye examination since the fundus can manifest risk factors (such as age, sex, smoking, hypertension, cholesterol, diabetes, and obesity) through fundus images [14], [15].

In medicine, the word fundus refers to the part of a hollow organ across from or farthest away from the organ's opening—simply the bottom or base of an organ—as in Latin, the word fundus means *the bottom.* In eye anatomy, the fundus is located inside, at the back surface of the eyeball, and the opposite of a pupil: made up of the retina (the light-sensitive screen), optic disc (the head of the nerve to the eye), macula with fovea (the small spot in the retina where vision is keenest), posterior pole (the layer between the optic disc and the macula), and the blood vessels [33].

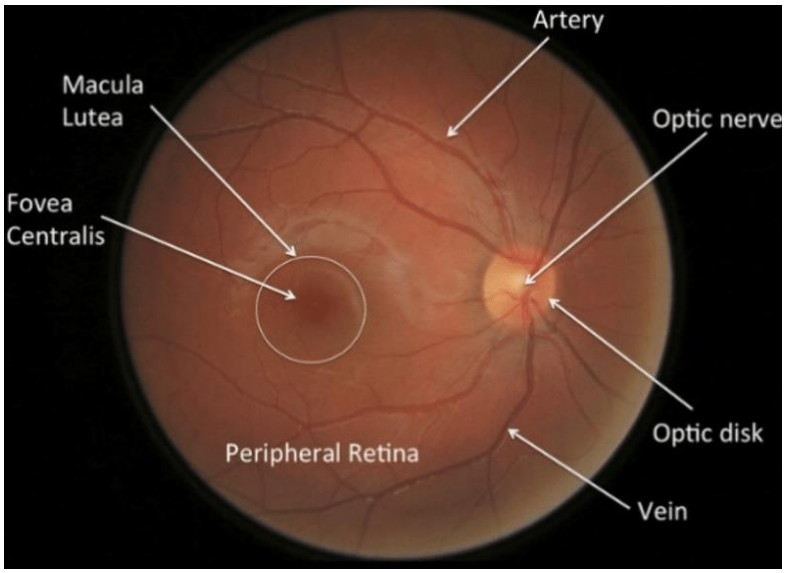


Figure 2-7: A cross-section of the Retina Fundus anatomy

The French physician (Jean Mery) was the first to come up with the idea to image the retina: by demonstrating the visibility of a cat's retinal vessels when submerged in water. In 1891, a German ophthalmologist (called Gerl) obtained the first retinal picture that showed retinal vessels and gross anatomy resulting in the discovery of the first fundus camera in the nineteenth century by Gullstrand—where the modern cameras advanced from the model [34]. Since then, retinal imaging has become a salient approach in medicine. Hence, to realize the approach mechanism, it is crucial to understand the role of the retina fundus as a part of an eye.

### 2.6.1. Human Eye Anatomy and Mechanism

Since vision is the most dominant sense of human beings, the eye is a vital sensory organ for all people with a sight—as it gathers information from the surroundings and translates it into a form of light before transforming it into an image—just like a camera—to aid human beings to see, differentiate colors (approximately 10 –12 million colors), and maintain the biological clock of the human body [35], [36];

An eye is the most complex part of the body, with an unperfected (rough) spherical shape, made up of muscles, tissues, nerves, and blood vessels—classified into two fused separated segments (external and internal structure) [37]. The eye anatomy is explained as follows [38]:

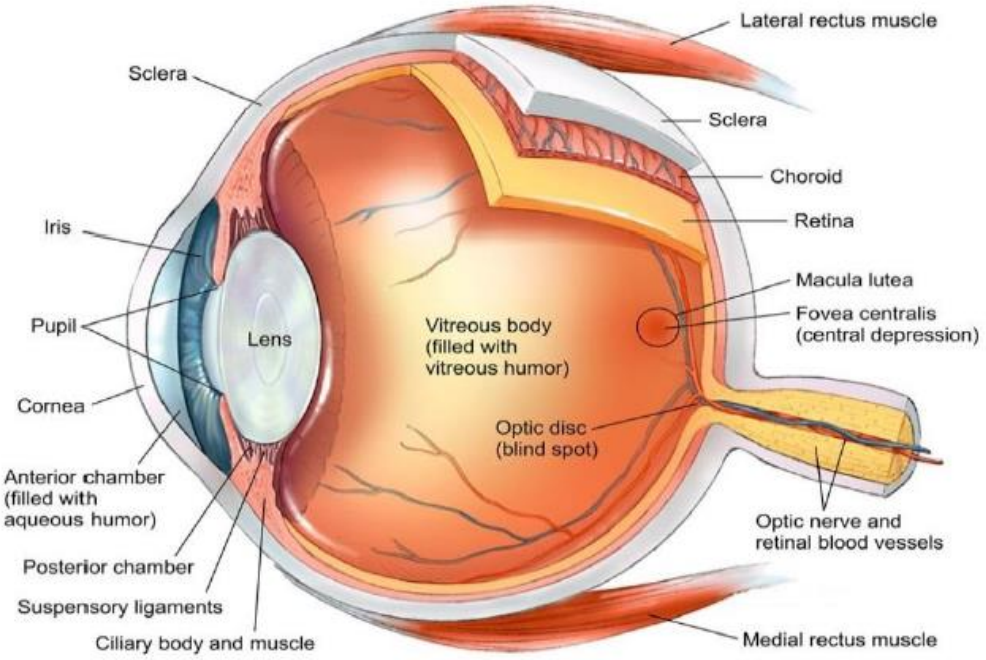


Figure 2-8: A cross-section of the human eye anatomy

Figure 2-3 above shows a cross-section of the human eye, where the light is filtered and refracted by the cornea (an outermost-domed shaped- eye layer), passed through the iris (an eye-colored-light regulator part) to the pupil (an iris-dark center opening) before being bent onto the lens (a clear inner part of the eye) and converged as an image (electrochemical impulses)—on the retina (a light-sensitive-eye-layer);

The iris regulates the amount of light from the cornea and adjusts the pupil's size for the light to pass. The auxiliary muscles help the lens to change its shape and refract the light rays (for the second time) to bring objects into focus or further improve and project the already refined image onto the retina to form an inverted (upside down) image.

This reversed image is passed through optic nerves (as neural signals) to the brain: which re-inverted the image (right-side-up)—this reversal of the images is much like a mirror in a camera. That means the eyes look, but the brain sees—as the brain adapts to new surroundings and translates the most important information captured through the eyes.

- **Similarities between Human Eye and Camera**

The anatomy of the camera resembles the human eyeball structure and mechanism. Both capture and similarly converge images—as they have lenses and light-sensitive surfaces. With these similarities, a camera appears as a robotic eye [39], [40]—however, they are by no means identical.

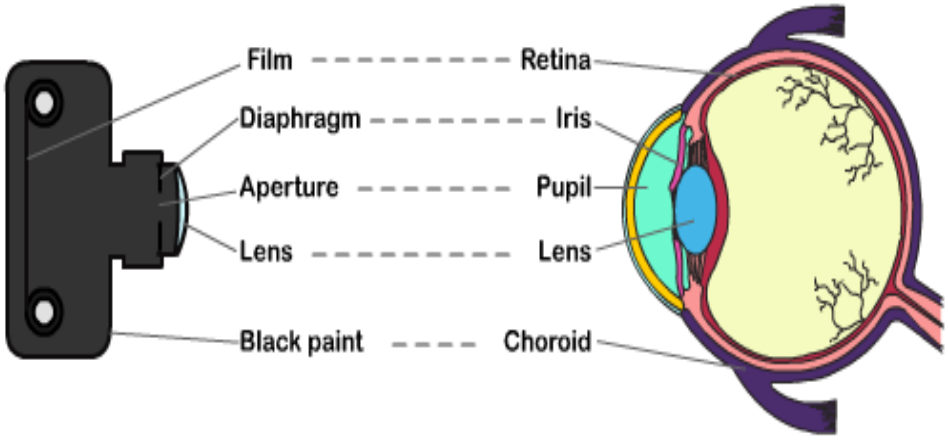


Figure 2-9: Human eye and camera

Like the cornea (transparent spherical membrane) in the human eye, the lens of a camera also maintains a spherical curvature to allow the camera to view through refracted light passed inside. As the iris and pupil are to the eye, the diaphragm (iris) in a camera regulates light by controlling the size of the aperture (pupil). The film or sensors in the camera, like the eye retina, receive the refracted light from the lens to form an inverted (upside-down) version of the image—as the lens is convex or curved outwards. Then the sensor chip (similar to the human brain) re-inverted the image (right-side-up).

The eye and camera also have similarities in the ability to focus, as their limitation is the scope of view: whether in the foreground or off at a distance, they only focus on one single object and blur the rest—likewise, the eye can focus on a larger image, just as a camera can focus and capture a large scape.

### 2.6.2. The Retina

The total surface area of the human retina is approximately 1,100square mm. Its average thickness is 200 micrometers—as it is slightly thicker near the head of the optic nerve and the macula and gradually thins out at the ora Serrata and fovea [41].

The retina constitutes six different cell lines: Rods, Cones, Retinal Ganglion cells, bipolar cells, horizontal cells, and Amacrine cells. These cells are classified into three (3) types (photoreceptors, neuronal cells, and glial cells) and divided into ten distinct layers of neurons interconnected by synapses [42].

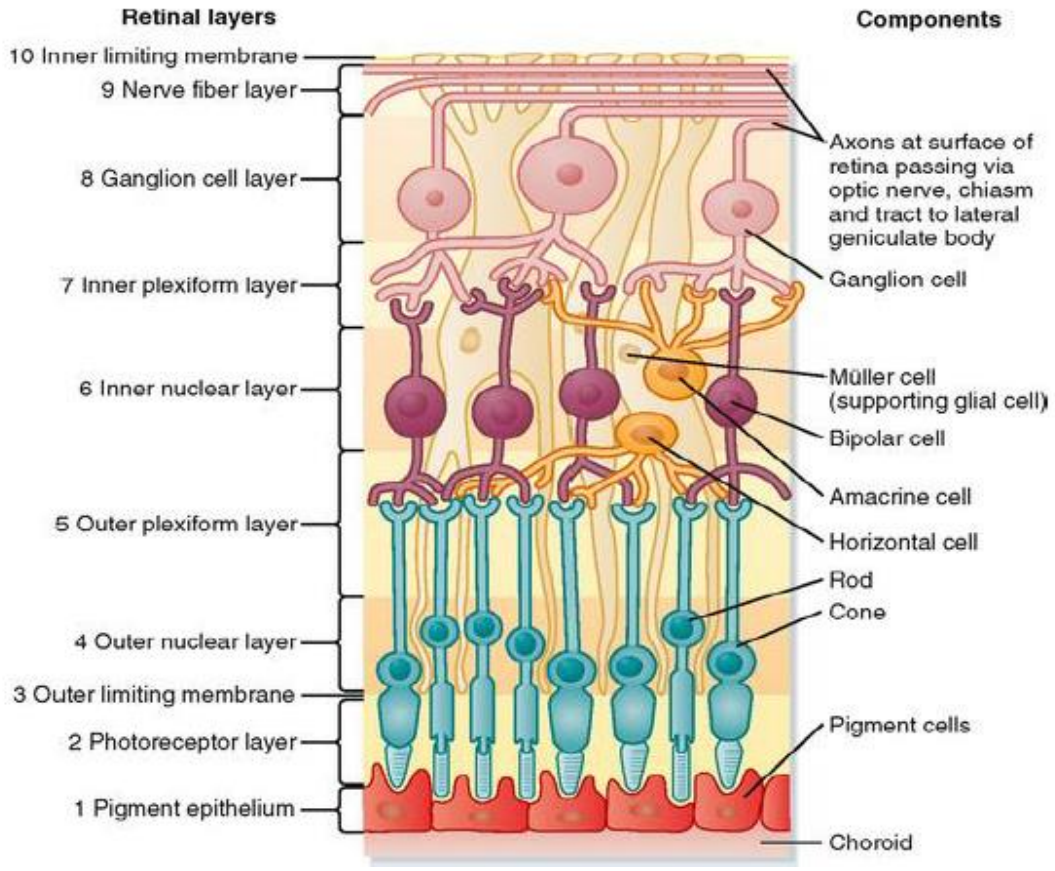


Figure 2-10: Organization of Retina layers

The figure above shows the layers from the innermost layer (closest to the vitreous) as follows:

- 1. The Inner Limiting Membrane (ILM)**:** the innermost layer of the retina that separates it from the vitreous body;
- 2. Nerve Fibre Layer (NFL)**:** the layer formed by the ganglion cell axons, which are the expansion of the fibers of the optic disc;
- 3. Ganglion Cell Layer (GCL)**:** the layer encompasses the ganglion cells and projects axons toward the optic nerve;
- 4. Inner Plexiform Layer (IPL)**:** the layer contains the synaptic connections between the axons of bipolar cells and dendrites of ganglion cells;
- 5. Inner Nuclear Layer (INL)**:** the layer consists of the bipolar cell, horizontal cells, and amacrine cells;
- 6. Outer Plexiform Layer (OPL)**:** the layer of neuronal synapses that consists of bipolar cells, horizontal and synapses of photoreceptors;
- 7. Outer Nuclear Layer (ONL)**:** the layer detecting the light portion of the eye consists of the nuclei of photoreceptors;
- 8. Outer Limiting Membrane (OLM)**:** the layer situated at the bases of the photoreceptors, separates the nuclei from the inner and outer segments;
- 9. Photoreceptor Layer**:** the layer contains the rods and cones;
- 10. The Pigment Epithelium Layer**:** the outermost layer contains pigmented cells called Melanin and is attached to the choroid that nourishes the retinal cells.

Photoreceptors are the special cells that process the light into an electrical signal to the brain. Two types of these cells (rods and cones) are responsible for night vision (white and black images) and color vision. Being sensitive to light levels (contrast and brightness), Rods enhance a good vision in dim light. Cones are responsible for vision in bright light and process colors with a high and spatial resolution—both work together to produce a clear and accurate two-dimension image of the visual world.

As shown in figures 2-3, the choroid is the pigmented layer that isolates the retina and sclera, located in retinal pigment epithelium (RPE): to supply nutrients to posterior layers of the retina, prevent it from overheating, and provide a network of blood vessels for supplying oxygen.

- **The Macula and Fovea**

The macula or macula lutea—as in Latin macula means a *spot* and lutea means *yellow*—is an oval-shaped yellow-pigmented area: to protect the retina by absorbing excess blue and ultraviolet light. The macula is at the center of the retina, made up of 200 million neurons with a diameter of 5.5 mm (0.22 in) responsible for central, high-resolution, and color vision in bright light.

Macula has six subdivisions: umbo, foveola, foveal avascular zone, fovea, parafovea, and perifovea areas. The fovea is at the center of the macula—the best visual acuity area—as it contains numerous cones for the sharp vision of primary human activities.

- **The Optical disk and posterior pole**

The optic disk or optic nerve head is the circular area in the retina fundus which connects the information from photoreceptors to an optic nerve. This area is also known as a blind spot —as the nerve fibers leave the eye and transfer the visual information to the brain.
The posterior pole is the area between the optic disc and the macula.

- **The Blood Vessels**

The retina consists of two blood vessels—the arteries and veins. The arteries carry oxygen and nutrients from the optic nerve to the retina through the central retinal artery (main artery)—and form a network of tiny thin blood vessels called capillaries. The wastes and carbon dioxide leave the retina through the joined capillaries to form the central retinal veins connected to the optic nerve before being transferred back to the heart. These vessels are essential that if they get damaged, the visual area of the retina may get affected.

### 2.6.3. Manifestation of CAD risk factors in retina fundus

Several studies have shown a positive affiliation between retinal microvascular abnormalities and CAD. Both modifiable and non-modifiable CAD risk factors such as Age, Sex, Diabetes, Hypertension, Cholesterol, and Obesity can manifest in the retina fundus by analyzing and evaluating retinal microvascular abnormalities via fundus images—through examining the alteration of retinal characteristics (convoluted texture features, patterns, and values) and observing the variability of blood vessels diameter [15], [16].


Figure 2-11: Normal and Abnormal Retinal Microvasculature Features

## 2.7. Chapter Summary and Conclusion

The literature survey has provided the theoretical background regarding the history, risk factors, and effects of CAD—outlined the evolution of risk stratification methods, and justified the breakthrough of machine learning in the prognosis and diagnosis of CAD. The survey has also highlighted the CNN deep learning algorithm to enhance the limitation of machine learning and underlined retinal fundus imaging as an easy, quick, and safe CAD biomarker for analyzing and evaluating the risk factors. Nevertheless, the literature overlooked the studies and practices in integrating the CNN algorithm and retinal fundus imaging to achieve an intelligent tool for stratifying the risks of CAD.

# CHAPTER 3:
# METHODOLOGY AND METHODS:

## 3.1. Introduction

This chapter details the practical how of the research study—the systematic research design. It explores the proposed methodology—CAD risk stratification using deep learning via retina fundus photography—and justifies the methods to achieve a reliable output that addresses the research goal and objectives.

In a brief introduction, the chapter justifies why deep learning is a suitable machine learning technique. It vindicates the application of MobileNet CNN architecture incorporating retina fundus images in model designing for CAD prognosis.

## 3.2. Overview: Deep Learning and Artificial Neural Network (ANN)

### 3.2.1. Why Deep Learning?

In this study, Deep Learning is a proposed approach to enhancing CAD prognosis. Compared to traditional Machine Learning (ML) algorithms, the technique performs better with the large volume (high-dimensional spaces) of unstructured data (like images) and classification problems of multiple domains. With automatic features engineering, DL saves time by automating complex tasks and provides desirable and adaptable solutions—thanks to its architecture [43], [44].

DL is a specialized subset of ML that synchronizes the connection between AI and data science. The algorithm mimics human brain behavior—inspired by a biological neuron network that processes information within the brain [45]. Although it has existed since the 1940s, DL has gained momentum recently and become a vast field (the mainstream) with a fast-evolving pace—regarded as the cornerstone of the next revolution in Information Technology. From known as cybernetics in the 1940s–1960s to connectionism in the 1980s–1990s before its resurgence in 2006, deep learning emerged in medicine in 2012—thanks to the availability of numerous data, the emergence of highly performed architecture (AlexNet), and the advancement of computational infrastructures capable of training large artificial neural networks (ANN) [43], [44], [46];

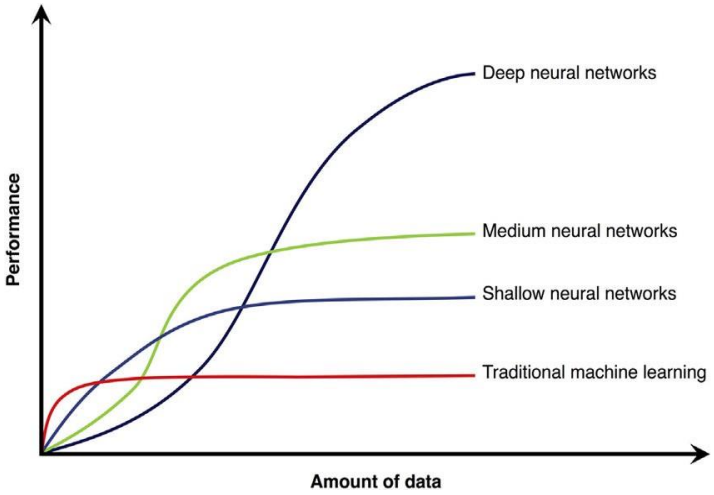


Figure 3-1: Performance of Deep Learning with Large Data

### 3.2.2. Biological Neural Network (BNN)

An artificial Neural Network (ANN) resembles a biological neuron network (BNN) as it simulates or mimics the human brain functions and behavior based on learning, relearning, and unlearning. To understand ANN, let's first explore the biological neural network [47], [48]. A Neural network consists of a series of densely connected neurons. A neuron is an elementary computation unit in the network. In the human body, a neuron is the nerve cell and a fundamental unit or primary function of the nervous systems and the brain. A neuron receives sensory input from the external world, sends motor commands to the body muscles, and transforms and relays the electrical signals from one to another. Neurons vary in size, shape, and structure depending on role and location—where the human brain contains approximately 85 billion neurons—the core reason why the brain is the most powerful organ [48].

A neuron—a tree-like structure—consists of three main parts: the branches and leaves (Dendrites and Spines), the roots (Axon and synapses), and the trunk (Soma). The Dendrites are an area that receives information from other neurons through receptors. The received signals are in the form of chemicals (called neural transmitters) and cause the electrical change translated into an area called the cell body or soma. A Soma consists of a nucleus enclosed by a membrane to maintain the neuron's structure and provide energy to drive activities. A Soma transmits the signal to the axon through the axon hillock. The axon, insulated by fatty materials (Myelin), conducts the electric impulses (known as the action potential) to other cells through the neuron synapses.

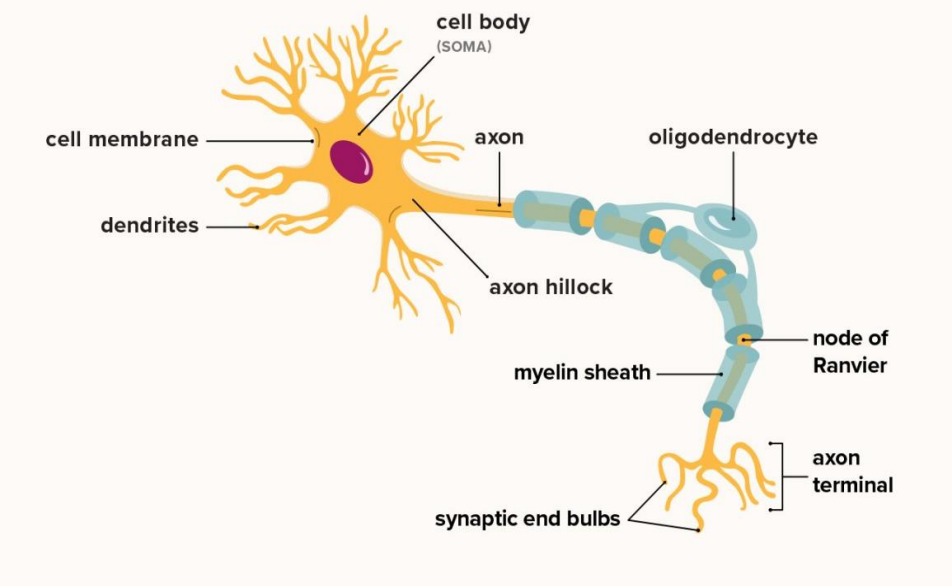


Figure 3-2: Anatomy of Biological Neuron

### 3.2.3. Artificial Neural Network (ANN)

As Biological Neural Network, the ANN (Multilayer perceptron-MLP) also consists of interconnected artificial neurons. Similar to a biological neuron, the artificial neuron (perceptron) also consists of three parts: the input (as dendrites), the body (as soma), and the output (as axon): where the neuron receives inputs (electric spikes) from other neurons to perform computations in the body network and transmits the information to other neurons in the output layer [49].

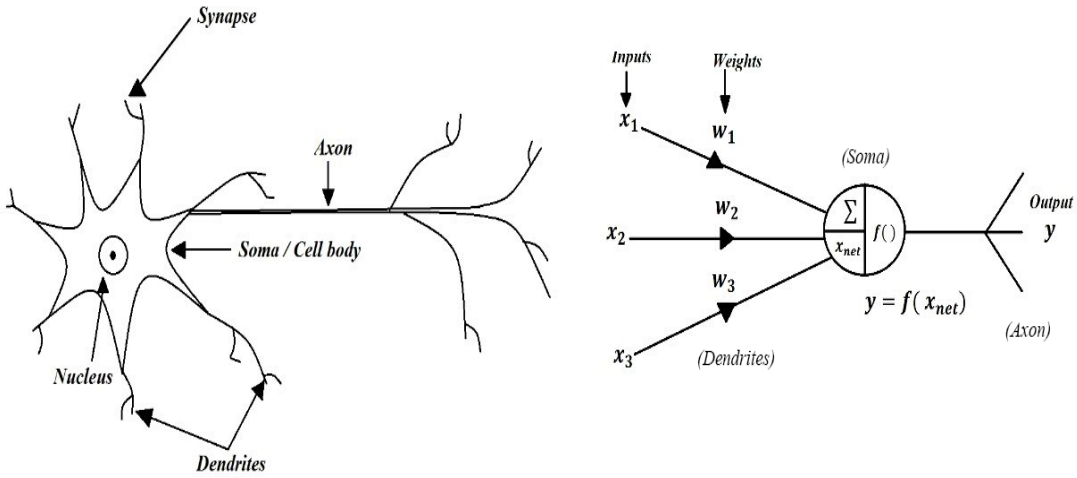


Figure 3-3: Relationship between ANN and BNN

Each neuron carries the weighted sum of the input parameters (*w* and *x*) from other neurons added along with the bias (*b*)—and optimized throughout the training: to enable the network to establish a strong connection. The magnitudes of the input parameters are strategically analyzed—since they influence the amplitude of the output.

The bias as a constant value or common vector makes the adjustments within the neurons and offsets the result. Whenever bias is not applied, the network will only perform matrix multiplication causes high variance (over-fitting model). It is crucial for flexibility and generalization as it can shift the activation function towards the positive or negative side to offset the result for the best model fitting.

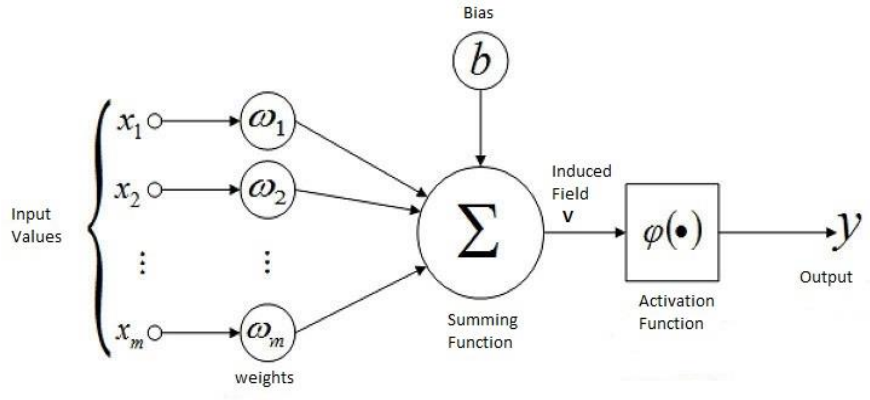


Figure 3-4: The Structure of Artificial Neuron (Perceptron)

The Activation (Transfer) Function ($\varphi$) suggests the degree to activate the neuron from the input of another neuron. The function decides the activation of the neuron—whether the received information is relevant to pass or not. The activation function non-linearly transforms the neuron to enable the network to learn and perform more complex tasks (through hidden layers). Without the transfer function, the neural network is essentially a linear regression model.

From figure (3-4) above, the summing function performs weighted sum of input $x_i$ and $w_i$ ($x_1w_1+x_2w_2+\ldots+x_mw_m$) added with the bias *'b'*, where '*i*' represents the number of corresponding input and *m* the total number of inputs in the neuron. In mathematical expression, the summation function is represented as $((\sum_i^m x_i w_i) + b)$. The output *y* is the result of activation function of the summation function represented as $y= \varphi((\sum_i^m x_i w_i) + b)$.

### 3.2.4. The ANN Topology—Multilayer Perceptron (MLP)

The ANN consists of three organized layers: the input layer, hidden layer(s), and output layer—as presented in figure 3.5. The input layer is responsible for receiving initial data (electrical signals) from other neurons. The hidden layers are intermediate layers consisting of densely connected neurons stacked vertically between the input and output layers; they are known as hidden layers because the parameters are unknown, and the process is not directly observable in the training set. The output layer consists of single or multiple neurons for the final output [43], [45], [50].

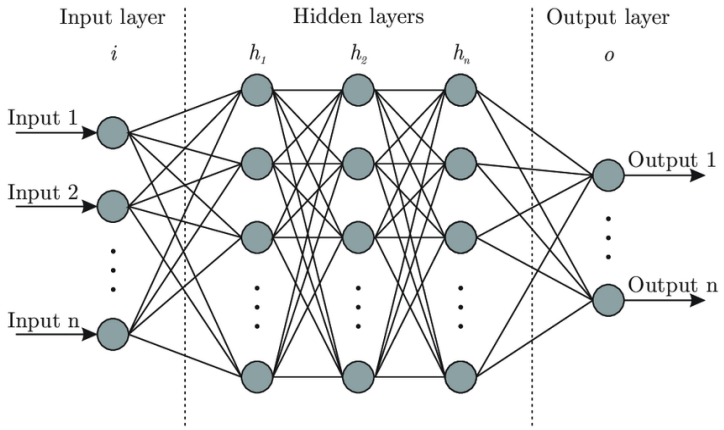


Figure 3-5: The Multilayer Perceptron or ANN Architecture

Since there is no activation in the input layer, the layer is inconsiderable. Hence, for the network with two layers, there is one input, one hidden, and one output layer. The number of nodes and hidden layers always determines the type of ANN—as classified into deep and shallow neural networks. A Shallow neural network—as shown in figure 3.6 (b), consists of a single hidden layer, and a deep neural network consists of multiple hidden layers—as in figure 3.6 (a) [50], [51].

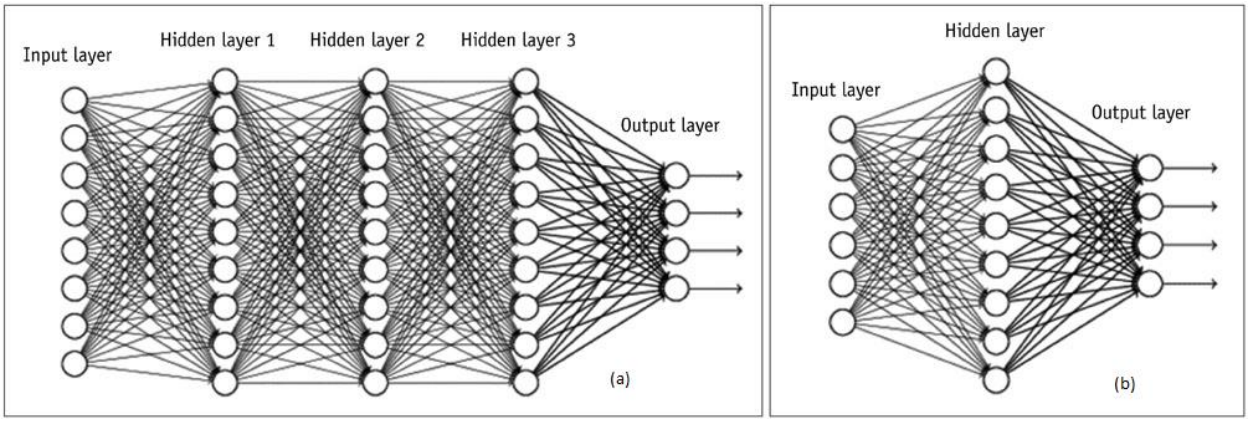


Figure 3-6: Deep and Shallow Neural Network

### 3.2.5. The ANN Training and Optimization

#### 3.2.5.1. Forward and Back propagation

The training or supervision of ANN is an iterative process approached into two stages: forward propagation and backward propagation. Learning methods depend on changing parameters for each neuron to achieve minimal error for accurate output. The training consists of the selected parameter coefficients (weight and bias) for every layered neuron in the network [43], [44], [50].

During the forward propagation process, the neurons receive data, process the information, and generate output in forward direction by random initializing the weights, biases and (or) filters (in the case of CNN algorithm). The stage involves the calculation and storage of intermediate values in input, hidden and output layer. It deals with already optimized parameters in pre-activation (linear transformation of input weighted sum) and activation (calculated input weighted sum passed to activation function). Figure 3-7 below shows forward propagation process where the activation output *'a'* of each neuron *'i'* in the corresponding layer *'l'* is given as $a_i^l = g(w_i^l\, a_i^{l-1} + b_i^l)$ where '*l-1*' indicates the input from the output of previous layer, '$g$' for activation function, and '$f_{wb}(x)$' for the final estimated output calculated at the last network layer.

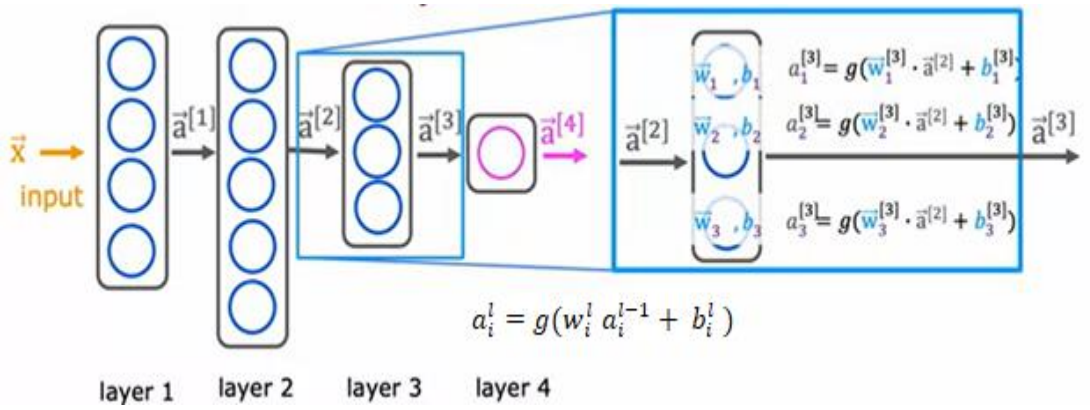


Figure 3-7: Forward Propagation in ANN

Backward Propagation optimizes the network by reversing the forward process. It **c**alculates the error and propagates it back to update the parameters of the network for accurate results. The supervision is through the training dataset that contains examples with the 'true' values: tags, classes, or indicators. The process calculates any possible errors or deviations of the estimated output from ground truth (Cost function) and updates the weight and bias parameters through gradient descent.

- **Cost function**

In Artificial Intelligence, the cost function estimates how successfully the model performs for a given dataset: by calculating the error resulting from the difference between the predicted (estimated) value and the actual (grounded truth) value—as a single real number. The difference is squared to avoid negative values and halved to give the small value possible. During training, the smaller the cost function, the closer the model is accurate. In linear regression, squaring the sum errors from each training example and half the mean gives the cost function. However, the technique doesn't apply to neural networks since resulting in optimization problems-the non-convex (without a global minimal) with many local optimum points. For neural networks, the cost function *'J'* is the average of the sum of loss (error) of each neuron in the layer of the entire training dataset given as $J(w,b) = \frac{1}{m}\sum_1^m L(f_{wb}(x), y)$ where the loss function is given as:

$$L(f_{wb}(x), y) = -ylog\left(f_{wb}(x)\right) - (1-y)\log(1 - log\left(f_{wb}(x)\right).$$

- **Gradient Descent**

The feedback from the cost function requires the adjustment of weight and bias parameters to minimize the errors and optimize the model. These updates are through an iterative optimization algorithm called Gradient descent. Gradient descent (Steepest descent) implements the first-order derivative to follow a negative of a gradient downhill: for finding a local or global minimum of a differentiable function by repeating steps in the direction of the corresponding point until the cost function is close to or equal to zero. The gradient of a given function is through differentiating the cost function concerning weights and biases, where the learning rate ($\alpha$) determines the iterative steps until the error is minimal. The gradient descent is as follows:

Repeat until converging

{

$$w_i = w_i - \alpha \frac{\partial}{\partial w_i}\ (J(w,b), \text{ where } \frac{\partial}{\partial w}\ (J(w,b) = \frac{1}{m}\sum_1^m (f_{wb}(x) - y)\, x$$

$$b_i = b_i - \alpha \frac{\partial}{\partial b_i}\ (J(w,b), \text{ where } \frac{\partial}{\partial b}\ (J(w,b) = \frac{1}{m}\sum_1^m (f_{wb}(x) - y)$$

} Simultaneous Update

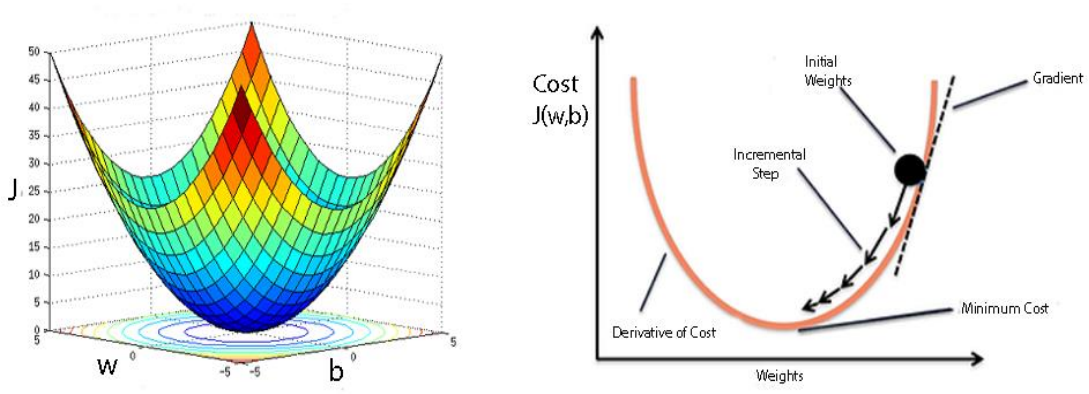


Figure 3-8: Gradient Descent Algorithm

The gradient descent algorithm consists of three main types: batch, stochastic, and mini-batch. Batch gradient descent (BGD) calculates and evaluates the loss (error) for each example within an entire training dataset in a single iteration but only updates the model after assessing all the training examples. Although it provides computation efficiency that produces gradient error stability and convergence, BDG is slow due to the long processing time and usually costs memory in training. Stochastic gradient descent (SGD) processes each example within a training dataset—in each iteration— and updates the parameters of each training example one at a time. SDG is fast, saves memory, and optimizes the local minimum problems, but is limited to less computationally efficient and noises from frequency updates that cause the error rate to fluctuate rather than gradually go down—continue oscillating around and never reach the global minimum. Mini-batch gradient descent combines the concepts of both BGG and SGD to provide a balance between computational efficiency, training speed, and memory. It splits the training dataset into small batch sizes, processes it into small subsets, and updates the parameters on each batch —in each iteration.

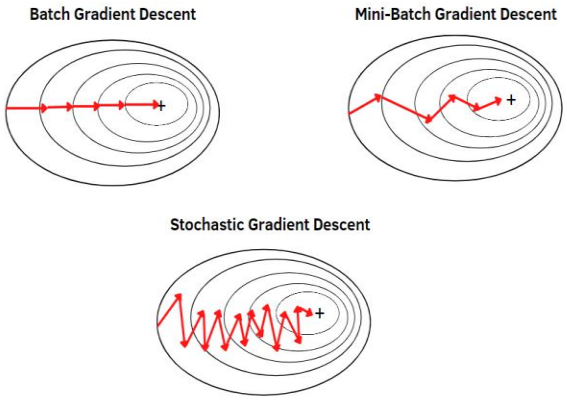


Figure 3-9: Gradient Descent Algorithm

- **Challenges with Gradient Descent**

Despite gradient descent being the most common approach for optimization problems, it has its own set of challenges involving Local minimum and saddle points and vanishing and exploding gradients:

### (i) Local Minima and Saddle Points

The gradient descent can find a global minimum in convex but not for non-convex problems. In non-convex—which is highly in neural networks, gradient descent results in local minima and saddle point challenges. Local minima mimic the global minima shape, but it has minimum values at different examples. However, the saddle point (the negative gradient) has one side that reaches a local maximum and local minimum on one side—which gives the shape inspired by the horse saddle. The local minima problem happens when failed to random initializing the small weight parameters to prevent neurons from adopting similar features (breaking symmetry) or using a few iterative steps during the training. The first-order derivate $(\frac{d}{dx}(f_{wb}(x) = 0)$ determines the minima or maxima points of the function then the second-order derivative identifies which point is the local minima by testing if they yield positive results $(\frac{d}{dx}(F_{wb}(x) > 0)$.

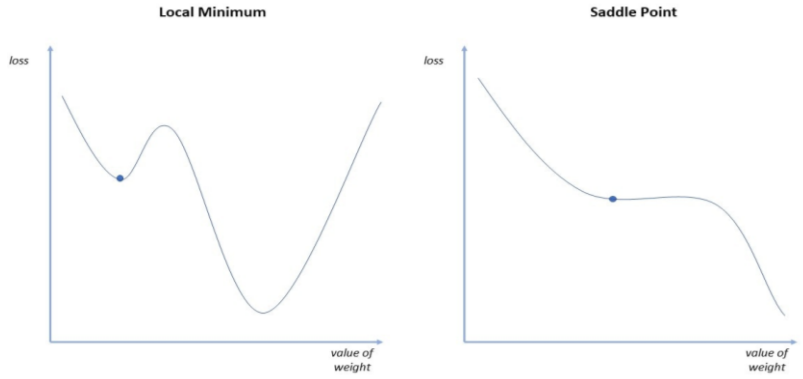


Figure 3-10: Local Minimum and Saddle Point

### (ii) Vanishing and Exploding Gradients

Failing to initialize or control the weight parameters may result in vanishing or exploding gradient problems. A vanishing gradient occurs during backward propagation when the derivatives (Gradient) are too small to cause the network's earlier layers to learn more slowly than later layers. The situation can cause the weight updates to approach zero—resulting in an algorithm that is no longer learning—never converging to the global minimum. The model will also diverge when the gradient is large, causing instability with large weights—a problem called Exploding gradient.

### 3.2.5.2. Improving Deep Neural Networks in Training

To achieve an optimized model for accurate results: it is necessary to consider the methods that improve the model during training. The techniques include hyperparameter tuning, regularization, and advanced optimization algorithms [43], [50] [52].

#### 1. Hyper-parameter Tuning

Unlike the parameters which learn during training, the hyperparameters determine how to train the model and control the learning process. They aren't directly trained (from the data) but are set before the training process or during model optimization. The tuning of hyperparameters defines the model architecture and improves the deep-neural networks for accurate results. This process of tuning hyperparameters in deep learning is called Orthogonalization. The following are crucial hyperparameters to consider for achieving a robust, accurate, and reliable model:

- **The number of neurons and layers in hidden layer**

The selection of neurons in the input and output layers is not a concern as it is for determining the number of layers and neurons in the hidden layer. In the input layer, the number of neurons equals the number of features of the given dataset, whereas in rare cases, there will be one input layer for the bias parameters. For the output layer, the number of neurons depends on the used model type: whether it is a regressor or a classifier. If it is a regressor, then the output layer will have a single neuron unlikely for the classifier with a single neuron or multiple neurons (softmax multi-class)—depending on the model class label.

There is no fixed approach for selecting the number of layers and neurons for the hidden layer. A few neurons may cause an underfitting (high bias) model, and many neurons lead to overfitting (high variance) problems. There are some rule-of-thumb methods for determining the number of layers and neurons in ANN that the number of neurons should be between the sizes of the input and output layers. They should be two-thirds (2/3) the size of the input layer plus the size of the output layer. The number of neurons should also be less than twice the size of the input layer.

- **The Batches sizes and Iterations**

To ensure computational efficiency, high training speed, and saving training memory—for large datasets (>2000), it is significant to split the training dataset into small sets called mini-batch. A single mini-batch contains training samples known as Batch sizes.

The training samples in a single batch can determine the total number of mini-batches required for training by taking the entire training dataset and dividing it by the batch sizes. The number of mini-batches is the one that determines the number of iterations.

In ANN training, iteration refers to a single update of the parameters to adjust a gradient descent. A single mini-batch iterate once, and the total number of iterations in an entire training dataset is called epoch—the completion of forward and backward propagation training at once.

The proper selection of mini-batch sizes is crucial for determining the number of mini-batches, iterations, and epochs. The approach for selecting batch sizes depends on the computer memory, CPU, and GPU. Batch size always has to be a power of '2' ($2^n$) since the architecture and memory capacity of CPU and GPU is of the power of '2'.

- **The Activation function in hidden and output layer**

The choice of activation function in hidden and output layers impacts the network's performance in learning complex (no-linear) features. Although there are many options for activation functions (including binary step, linear, and leaky ReLU), the Logistic (sigmoid), hyperbolic tangent (tanH),

ReLU, and the leaky ReLU function are the most popular.

The Sigmoid activation function —as in logistic regression—takes any input values and outputs in the range from 0 (for larger negative values) to 1 (for larger positive values) with a threshold of 0.5. The hyperbolic tangent activation function is similar to the sigmoid function (also with the same S shape), except the range is from -1 (for larger negative values) to 1 (for larger positive values) with a threshold of zero. The Softmax function outputs the class probabilities with the vector values summed to 1—the higher the class probability, the higher the accuracy of the classified label. The ReLU (Rectified Linear Unit) only deactivates the activation of neurons when the output is negative (<0), otherwise gives maximum positive values (z) for every positive input value (z) —mathematically as $g(z) = max(0, z)$.

For the output layer, Sigmoid is reliable for binary or multi-labels classification, Softmax for classifying multi-class, and ReLU for positive values prediction. In the hidden layers, ReLU activation is the better and more prevalent activation function since it can quickly converge for complex features without saturation.

The sigmoid and hyperbolic tangent functions are inappropriate activations in hidden layers as they perform poorly and lead to the problem of vanishing gradients for larger positive and negative function values. The softmax is also unsuitable in hidden layers since it makes the neurons linearly dependent and fails to generalize complex features.

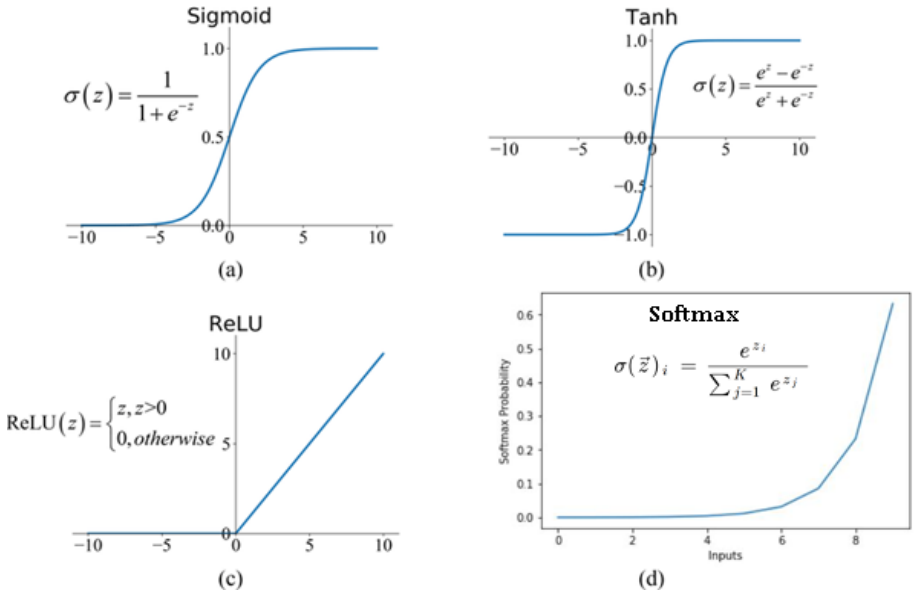


Figure 3-11: Activation Functions in the Neural Network

- **The Learning rate**

The learning rate ($\alpha$) is a vital configurable hyperparameter that determines the size of iterative steps and the speed or rate at which the model learns or updates the network's parameters. It should neither be large nor small —as a large-learning rate makes the model overshoot (diverge) towards the global minima while the small learning rate slowly makes the model converge—as it requires many steps of iterations. The desirable learning rate is through trials and errors but often has a small positive value ranging between 0 and 1—(<1.0 and >10^-6)—where traditional default values are 0.1, 0.01, and 0.001.

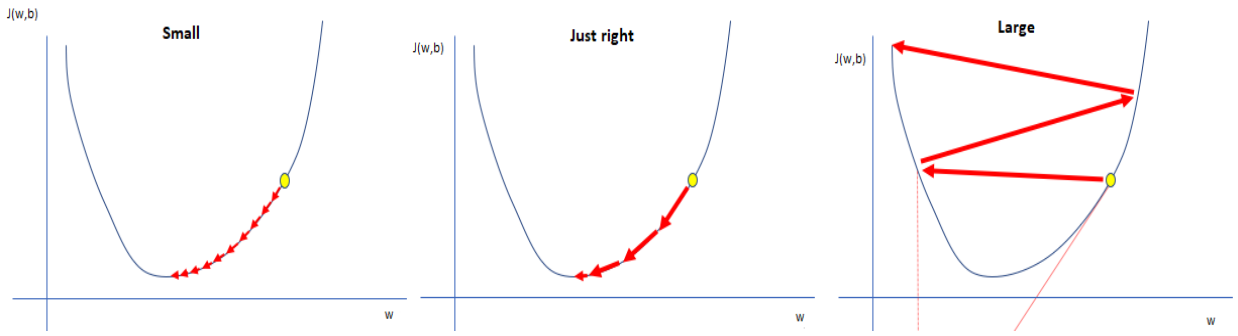


Figure 3-12: The Learning rate Hyperparameters

## 2. Regularization

Among the shortcoming of the neural network is the overfitting (high variance) problem—the model generalizes well in training but fails in new examples. The problem is when the model has complex features (many neurons or layers) or with fewer data available for training. Regularization is the technique to optimize the model from overfitting problems. The regularization rate determines how penalizing the weight parameters (making features smaller) can optimize the model for a simpler network. Regularization involves the techniques such as L1 and L2 regularization, Dropout, Early stopping, and Data augmentation [53].

- **L1 and L2 Regularization**

L1 (Lasso Regression) and L2 (Ridge Regression) are regularization techniques for preventing overfitting problems. They modify the network by normalizing the parameters in the cost function. L1 regularization penalizes the unessential feature's coefficient to zero by adding absolute an matrix value ($|| w^{[l]}||_1$). This technique becomes relevant whenever a huge network requires feature selection. There is no feature selection in L2 regularization since it adds squared magnitude using the Frobenius norm ($|| w^{[l]}||_2^2$) values to shrink the parameters—approximately but not zero. Considering the L2 regularization, the cost function will be;

$$J(w,b) = \frac{1}{m}\sum_1^m L(f_{wb}(x), y) + \frac{\gamma}{2m}\sum_1^{[l]} || w^{[l]}||_2^2,$$

$$\text{Where } || w^{[l]}||_2^2 = \sum_{i=1}^{n^{[l]}} \sum_{j=1}^{n^{[l-1]}} (w_{ij}^{[l]})^2$$

$$\frac{\partial}{\partial w}\ (J(w,b) = \frac{1}{m}\sum_1^m (f_{wb}(x) - y)\, x + \frac{\gamma}{2m}\sum_1^{[l]} || w^{[l]}||_2^2$$

$$\frac{\partial}{\partial w}(J(w,b) = \frac{1}{m}\sum_1^m (f_{wb}(x) - y)$$

- **Dropout Regularization**

While L1 and L2 regularization reduce overfitting by modifying the cost function, the Dropout technique modifies the network itself by temporarily eliminating randomly selected network features during forward pass—in each iteration—based on the probability. It involves going through each of the layers and setting the chances of dropping out some neurons—which means with a probability (p), a neuron gets turned off for achieving a diminished network—a simpler network. Figure 3-13 (b) below shows the dropout network with a probability of p=0.5, indicating that half of the neurons in the layers are not active.

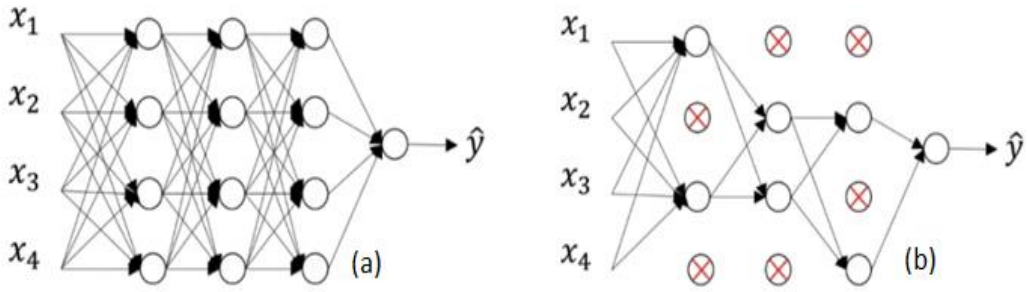


Figure 3-13: Feed Forward (a) with Dropout Regularization (b)

Dropout is applicable during training—in both forward and backpropagation (without updates for dropped neurons). However, it is not appropriate during the testing where all the neurons are present. Dropout has the same effects as L2 regularization, though its cost function is not well defined and hence not easy to debug the network. Despite being useful in the fully-connected layer in the CNN algorithm, the technique is less relevant in the convolution and pooling layers since the layers have fewer parameters [54].

- **Early Stopping**

During training, sometimes more epochs are used than required—causing the model to overfit. Early stopping is the regularization technique that optimizes the model to overcome overfitting problems during training—without compromising the model's accuracy. The main idea behind this technique is to halt training the model when the training error stops decreasing; and the validation error starts increasing—indicating no improvement in the validation set. That means the process also improves the model performance on the validation dataset as it stores the trainable parameters periodically while tracking the validation loss until it starts rising. For this reason, early stopping is the effective and efficient hyperparameter selection algorithm since it determines the number of epochs to run before overtraining the model [55], [56].

The technique also has the same effect as L2 regularization—since the process stops when the parameters get small. Despite the similarity, the early stopping technique doesn't simultaneously minimize the loss and overfitting problems—rather than concentrating more on reducing the cost function— contradicting the orthogonalization concept. However, this approach is relevant as it doesn't concern other hyperparameters like lambda in L2 regularization. The early stopping analysis is through the graph that plots the costs of the training and development sets for each iteration—as shown in figure 3-14 below.

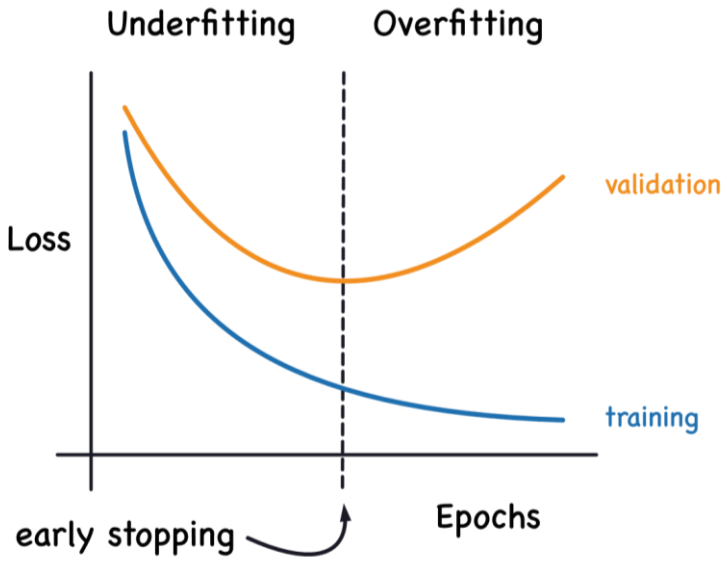


Figure 3-14: Early Stopping Regularization Technique

- **Data Augmentation**

Since the DL algorithm requires a vast amount of data to reduce the overfitting problem, it is not always possible to attain enough data for training—as the process may be slow and expensive. Hence, improving or expanding the available data to increase the training set becomes a relevant approach. The technique of generating new training examples from unstructured data: by slightly modifying the existing dataset is called Data augmentation. Although these new data aren't as good as the real independent ones, they are much more worth optimizing the model performance from high variance—lowering generalization error. This regularization technique modifies data in several ways: data manipulation using geometrical transformation like rotating, reflection (mirroring), translation, or random cropping. Other approaches include Feature Space Augmentation, GAN-based Augmentation, and Meta-learning.

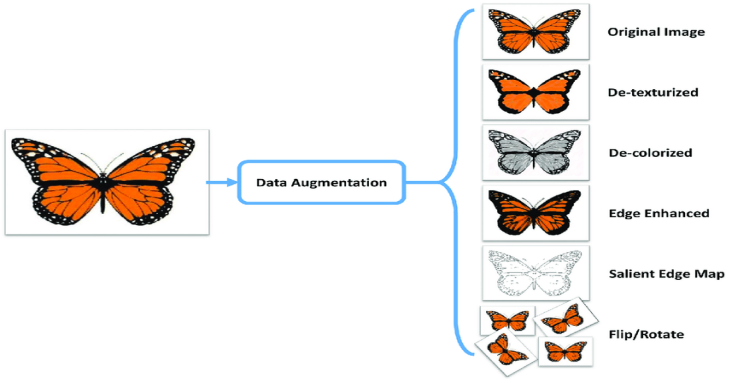


Figure 3-15: Data Augmentation Regularization

Apart from data augmentation regularization, Data synthesis is another technique that provides robust and versatile sufficient data. In most cases, this approach is dominant in computer vision tasks. Unlikely data augmentation, data synthesis makes new data examples from the ground without manual labeling. The technique generates and customizes data from artificial inputs (computer) to mimic real-world observations whenever real examples are not supportive. It also minimizes the constraints of regulated or sensitive data.

## 3. Advanced Optimization Algorithms

Traditional gradient descent has some limitations for achieving an optimized model. The algorithm can be noisy with many oscillations that require high computation time due to many epochs for large and complex datasets. The method only works on convex and not on non-convex functions—which is not ideal in the real world. Gradient descent can also lead to local minima, vanishing, or exploding gradient problems [57]. Due to these drawbacks, traditional gradient descent doesn't always work and needs to be advanced. Although stochastic and mini-batch techniques have enhanced the method, these are efficient algorithms designed and proven for advanced optimization:

- **Momentum**

Although SGD is faster than traditional gradient descent, it takes many iterations to converge, and the weights updates are noisy and result in non-convex behavior. Momentum is an extended gradient descent algorithm advanced to optimize non-convex functions in deep neural networks. It outperforms traditional SGD as it improves gradient descent by reducing oscillations (noises) through leveraging exponential weighting average for a fast converging. Momentum accelerates the convergence towards the relevant direction and minimizes the fluctuations of the irrelevant that the gradient won't stay in the local minima. Generally, this technique can optimize all the problems that traditional SGD can solve [58].

Momentum reduces the oscillations using an exponentially weighted average that controls the weight and bias parameters by smoothening out the skewed data points averages and giving new values that update the gradients based on the value $\beta$. The update of gradient descent with momentum is as follows:

$$w_i = w_i - \alpha Vdw, \text{ where } Vdw = \beta Vdw + (1-\beta)dw$$

$$b_i = b_i - \alpha Vdb \text{ , where } Vdb = \beta Vdb + (1-\beta)db$$

The default best average values of $\beta$ are between 0.9 and 1— where 0.98 is preferable. The '$dw$'and '$db$' indicate the derivatives of new sample value. Sometimes, the momentum may miss the local minimum and continue rising when it is too large. Hence, to resolve the issue, Nesterov Accelerated Gradient (NAG) algorithm comes into play [59].

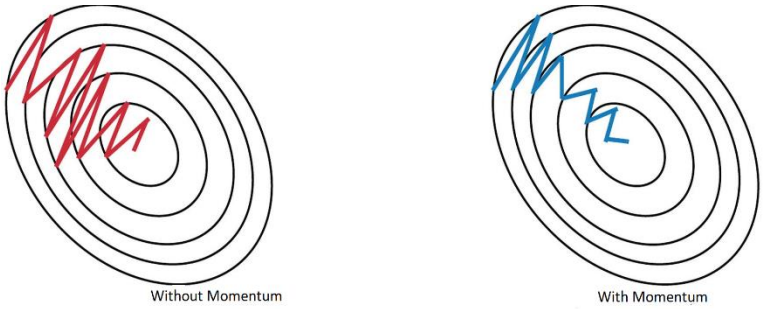


Figure 3-16: SGD with and without Momentum

- **RMS Prop**

RMSprop (Root Mean Square Propagation)—proposed by Geoff Hinton—is an advanced algorithm extended by optimizing the Adaptive Gradient (AdaGrad) [60] and Adaptive Delta (AdaDelta) [61]. These algorithms suggest the learning rate is never constant or considered as a hyperparameter rather than adapting to each weight of the function. RMSprop resembles the first-order updates of the AdaDelta algorithm. It uses an exponentially decaying average of squared gradients: to automatically adapt to the step size of each input parameter—as it discards the history of early derivatives and concentrates on the most recent 'gradients' to converge rapidly at a global minimum. The algorithm slower the cost function in a vertical direction but speeds up the horizontal direction. RMSprop updates the parameters by adapting the step size of input variables by dividing the learning rate with the square root of exponentially average parameters of squared gradients. The updates are as follows:

$$w_i = w_i - dw\frac{\alpha}{\sqrt{({S_{dw}}^2 + \varepsilon)}}, \text{ where } S_{dw} = \gamma S_{dw} + (1-\gamma)\, dw^2$$

$$b_i = b_i - db\frac{\alpha}{\sqrt{({S_{db}}^2 + \varepsilon)}}, \text{ where } S_{db} = \gamma S_{db} + (1-\gamma)\, db^2$$

Here $\gamma$ represents the RMSprop hyperparameter. The addition of epsilon '$\varepsilon$' which by default may be $10^{-8}$, is for numerical stability to ensure the denominator is not zero.

- **Adam Optimization**

Adam (Adaptive Moment Estimation) is an advanced algorithm that achieves optimal results by combining RMSprop and SGD with momentum. It is a computation-efficient algorithm that requires less memory for the 'large' data and non-stationary problems. The algorithm automatically adjusts the learning rates (reduce the larger and increase the smaller)—as it adapts from the estimates of the average of the first moment (as in RMS prop) and second moments of the gradients (the uncentered variance): to optimize high oscillations (very noisy) and sparse gradients problems. Adam is the recommended method for ANN as it has proven to be accurate and faster than other algorithms since it implements bias correction and its hyperparameters have an intuitive interpretation that require less tuning. By default, the hyperparameters are as, $\beta = 0.9$, $\gamma$ =0.999, and $\varepsilon$ =$10^{-8}$ where $\alpha$ it has to be tuned for every single parameter (maybe 0.1, 0,01, or 0.001) [62].

Parameters updates

$$w_i = w_i - \alpha_i \frac{Vdw^{corr}}{\sqrt{(S_{dw}{}^{corr} + \varepsilon)}}, \text{ where } Vdw^{corr} = \frac{Vdw}{(1-\beta^t)}$$

$$b_i = b_i - \alpha_i \frac{Vdb^{corr}}{\sqrt{(S_{db}{}^{corr} + \varepsilon)}}, \text{ where } Vdb^{corr} = \frac{Vdb}{(1-\beta^t)}$$

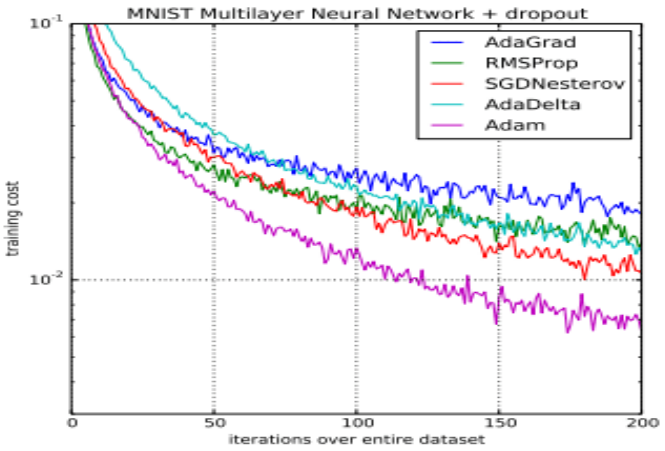


Figure 3-17: Adam Compared to Other Optimization Algorithms

### 4. Learning rate decay and Batch Normalization

Learning rate decay and Batch normalization speeds and stabilizes the model training. Learning rate decay reduces oscillations— decays the huge (α) in every iteration: for a model to converge [63]. Batch Normalization normalizes the input layer's parameters to mitigate the problem of internal covariate shift (the mean and standard deviation changes in batches distribution) [64].

## 3.3. Convolution Neural Network (CNN) Algorithm

### 3.3.1. Overview and History of CNN

Over the decades, numerous studies attempted machines to recognize visual patterns—the computer to mimic human brain vision. In 1959, David Hubel and Torsten Wiesel experimented on a cat's visual cortex that discovered the cells that recognized the edges and bars of a particular part (simple cells) and any part of the image (complex cells) [65]. The 1963 BLOCK WORLD study of Larry Roberts achieved the field of Computer vision where it became possible to extract 3D geometrical information from 2D perspective views [66]. Inspired by the study of David Hubel and Torsten, Dr. Kunihiko Fukushima (1980s) designed the earliest neural network that mimics simple and complex cells. He called it Neocognitron [67]. After Fukushima's discovery, the postdoctoral computer science researcher—Yann LeCun—developed the first CNN architecture (LeNet) for recognizing hand digits [68], [69]. Yet LeNet was limited to less training data and computational resources that performed on low-resolution images. The AlexNet emergence in 2012 became the limelight for computer vision tasks and a breathtaking innovation for visual recognition systems, where later other architectures (like VGG, ResNet, InceptionNet, MobileNet, EfficientNet, and GoogleLeNet) for the purpose [70], [71].

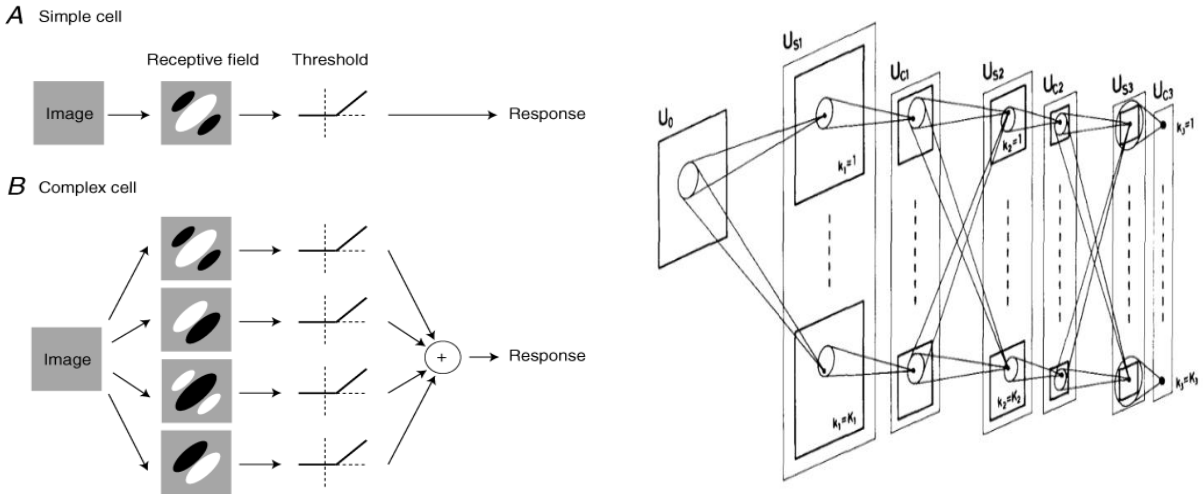


Figure 3-18: Simple and Complex Cells and Interconnected layers in Neocognitron

CNN (ConvNet) is the multi-layer and feed-forward type of neural network (FNN) designed for visual systems recognition [71]. Inspired by the visual cortex of the animal's brain, the CNN deep learning algorithm has enhanced the computer vision field—aids a computer in classifying images and video, detecting objects (localization and segmentation), and generating art [72], [73].

As animals can recognize visual patterns through interconnected biological neurons in the brain, the Computer is also capable of identifying images or object through interconnected artificial neurons—thanks to the CNN architecture. To recognize the visual features, the neurons in the brain form a network layered into three main layers: the lower, mid, and higher features layers. The low-level layers are responsible for detecting primitive features like lines, curves, and dots. The mid-layer features combine the primitive features to extract more information like corners, basic patterns, textures, or shapes**.** The Higher-level layer integrates complex patterns to recognize the original object or image. This mechanism is the one that inspires the CNN architecture—that it also layers multiple artificial neurons where each layer becomes responsible for recognizing specific features—before being fully connected to extract the original object or image [74], [75].

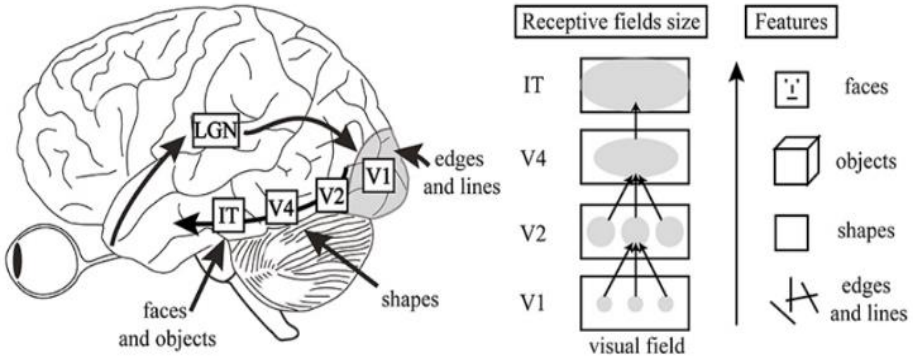


Figure 3-19: Human Visual Cortex resembles the CNN

Not only autonomous (self) driving, hand digits recognitions, face verification and recognition, and neural style transfer, CNN has become integral to medicine. Recently, CNN has outperformed human experts in analyzing and understanding medical images: this includes medical anomalies such as cancer, glaucoma, diabetic retinopathy, tumors, interstitial lung diseases, and tuberculosis [76], [77]. Due to its high image performance, CNN has also become a breakthrough in detecting heart diseases through cardiac imaging [17], [78].

As for other neural networks, CNN also consists of an input layer, hidden layers, and an output layer where each neuron is initialized with weights and updated to optimize the error [44]. Compared to a Regular Neural Network, the CNN algorithm scales huge three dimensions (3D) images in high-dimensional (in a deeper network) with fewer parameters: which speeds the training, requires less memory (due to fewer layers), and reduces overfitting risks (with less training data) [43], [44], [79].

It maintains spatial information (positional and neighborhood pixels) and intelligently adapts to the properties of an image like transformation [71], [80]. For example, CNN can handle invariance property—recognizing the picture even when its orientation or sizes or appearance changes [81]–[83]—as shown in Figure 3-20 below.


Figure 3-20: Recognition of Image properties in CNN

The algorithm is known as Convolution because it mathematically performs Convolutional operations to a place of general matrix multiplication of the traditional neural networks—for at least one layer. The Convolution operation is applied to the input data to filter the information and produce a feature map. For example, in the input image of thousands or millions of pixels, meaningful feature detection (such as edges, points, curves, and corners) might be achieved with the filter of only tens or hundreds of pixels [73], [84].

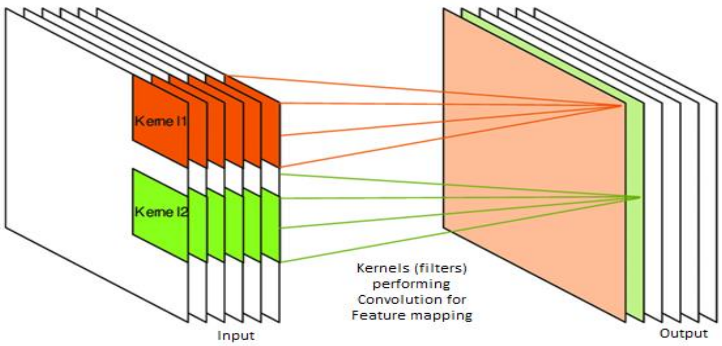


Figure 3-21: Feature Mapping Using Convolutional Operation

- **The Convolution Operation—the motivation behind CNN**

Convolution is a mathematical operation that performs an integral product (the amount of overlap) of one function shifted over another to produce a third function—simply intertwining two sources of information to form a new one [85].

Convolution symbol

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau$$

y(t) is the new function convolved from function x(t) and h(t)

The input functions can have different sizes, but they should be of the same dimensionality that the output will have similar dimensions. One common application of convolution operation is in image processing during the blurring and sharpening of an image and enhancing edges and emboss [86]. In the case of CNN, convolution leverages CNN in three ideas: sparse interactions, parameter sharing, and equivariant representations [43].

The sparse interaction concept (also known as spatial connectivity) refers to— instead of using all image pixels, only a few with relevant information are for feature extraction. The process increases the computation speed and saves the storage requirements. The accomplishment is through the filtering technique by making the filter (kernel) small than the original input [87]. Figure 3.22 below shows the sparse connectivity between the node x3 and output nodes (s2, s3, and s4) — indicating that only these three nodes are affected by the input node x3.

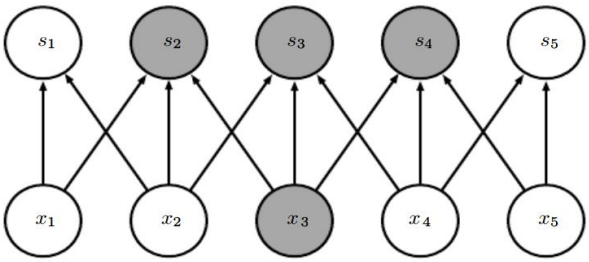


Figure 3-22: Sparse Connectivity in CNN

Distinguishing from regular neural networks that each weight matrix is computed once in the output layer, the CNN performs parameter sharing (tied weights) with the same set of parameters for more than one function. Each output neuron in a layer shares the same spatial features (as indicated by a black arrow in figure 3-23 (a)). The tie of weighted parameters reduces the computational complexity and storage requirements [88].

The concept of parameter sharing leads to equivariant representation—the symmetric presentation and data preservation—concerning image properties like translation, rotation, or mirroring—means when the function (g) acts on the input and output matrices, the results of the two networks will be equal (as shown in figure 3.23 (b). For example, translating the input image to a convolutional layer will also shift the output image. The equivariant concept is pivotal in CNN as — it reduces the number of trainable parameters— only cares about the features' presence and not their position. However, the property is the major demerit of CNN—the inability to detect the image's position [89].

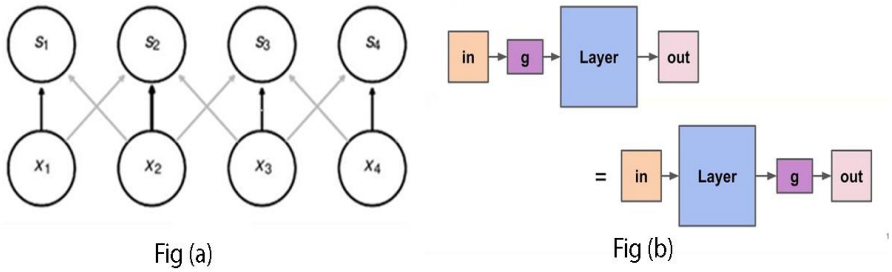


Figure 3-23: Parameters Sharing and Equivalent Representation in a Network

### 3.3.2. The ConvNet (CNN) Layers

Figure 3-24 below shows the CNN architecture with two main parts (feature extraction and down-sampling layer) staked into three main layers: Convolution layer, Pooling layer, and Full connected layer. Each layer passes its results into another layer to reduce dimensionality, save storage requirements, and achieve results at high computation speed [43], [68], [71], [90].

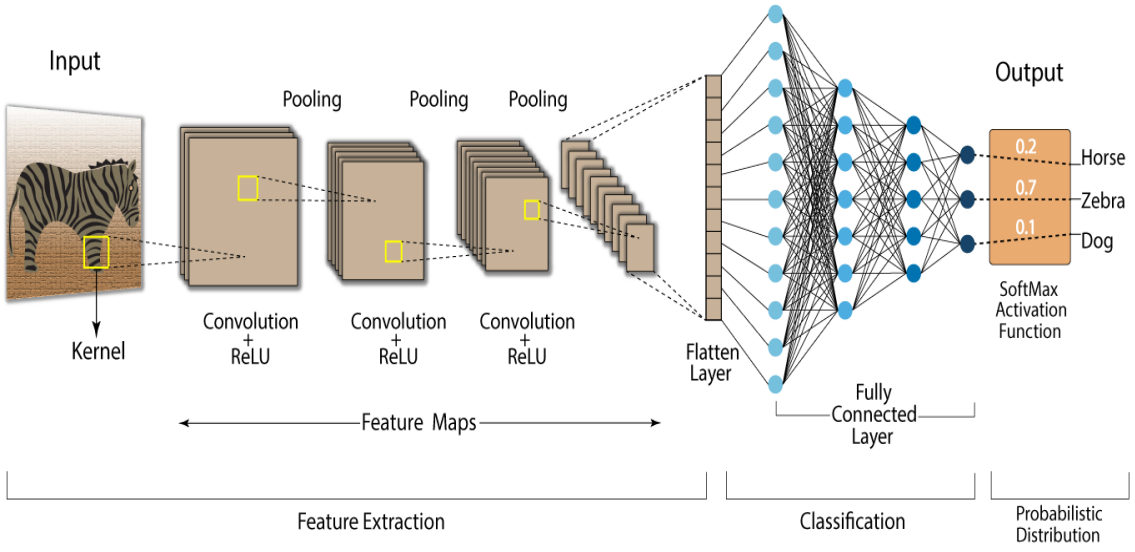


Figure 3-24: The CNN Architecture

#### 3.3.2.1 The Convolutional Layer

Apart from dense, the Convolutional is another ANN layer where the earlier and majority computation of CNN occurs. The layer is the core building block of the network and is responsible for feature extraction. The convolution layer ensures spatial relationship. It allows the extraction of relevant patterns of the image through matrix multiplication that converts pixels into numerical values. The Convolution layer architecture consists of input data, a filter (Kernel), and a feature map (the output): which are achieved through the techniques such as filtering, striding, and padding.

- **The Kernel (Filter) or Feature Detector**

The kernel (or) filter (or) feature detector is a weight matrix implemented to extract spatial features of an image. It performs a convolution operation by sliding over the input image from the top left corner and bringing information into a single pixel. Kernel reduces the size of an input image by performing matrix multiplication of each element in a receptive field (an area where convolution takes place) and sums them all to vector out a feature map.

The kernel has the same dimension as the input image, but its size is always smaller than the original input. If the input image is 2D (grey-scale) or 3D (RGB), the kernel will also be 2D or 3D. However, the output dimensionality depends on the number of filters (kernel) used. The filter reduces the output image size and saves computational storage.

For the input image with the '*n'* dimension and the kernel *'k'* dimension, the output (feature map) will be the dimension difference between the input image and the kernel plus the bias (which is one by default). Hence, mathematically the output will be '*(n-k) +1'*.

Figure 3-25 below shows how the filter reduces the pixel size through convolution operation in a receptive field. It shows that the 2D (5x5) image is convolved with the one 2D (3x3) filter to give a 2D (3x3) output image. The kernel convolves the input image pixels by perfuming the dot product in a field and the feature map dimension is given as:

$$\textit{Feature map (Output) dimension}$$
$$=[(9\text{x}0)+(4\text{x}2)+(1\text{x}4)+(1\text{x}1)+(1\text{x}0)+(1\text{x}1)+(2\text{x}0)+(1\text{x}1)]=16.$$

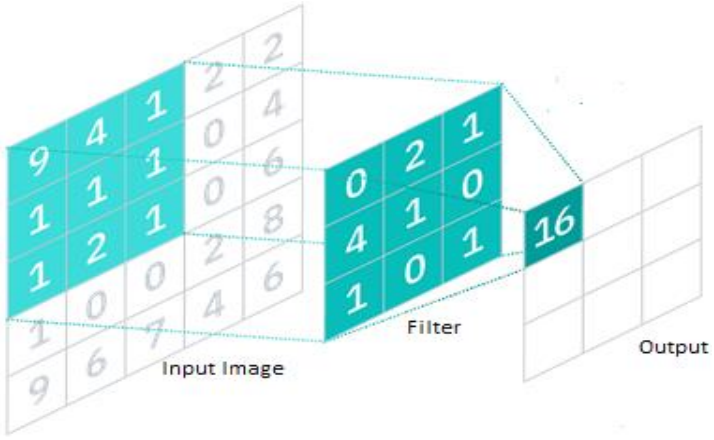


Figure 3-25: Filtering in Convolution Layer

- **Padding and Striding**

Striding and Padding are the techniques for performing convolution in CNNs. They are hyperparameters that process images more accurately and at high computation.

Striding is the technique for the convolution layer to achieve output with a smaller dimension. Stride refers to the distance (the number of pixels) that specifies how much the kernel slides over the input matrix—and by default, the value is one (1). Striding controls how the filter convolves the input matrix by providing less overlap within receptive fields as it skips over some areas to deal with potential locations. The technique reduces spatial resolution (down-sampling) and makes the network computationally efficient.

For the input image with the '*n'* dimension and the kernel *'k'* dimension slides over stride *'s'* distance, the output (feature map) will be the dimension difference between the input image and the kernel divides the stride, plus the bias (which is one by default). Hence, mathematically the output will be $\left(\frac{(n-k)}{s}+1\right)$.

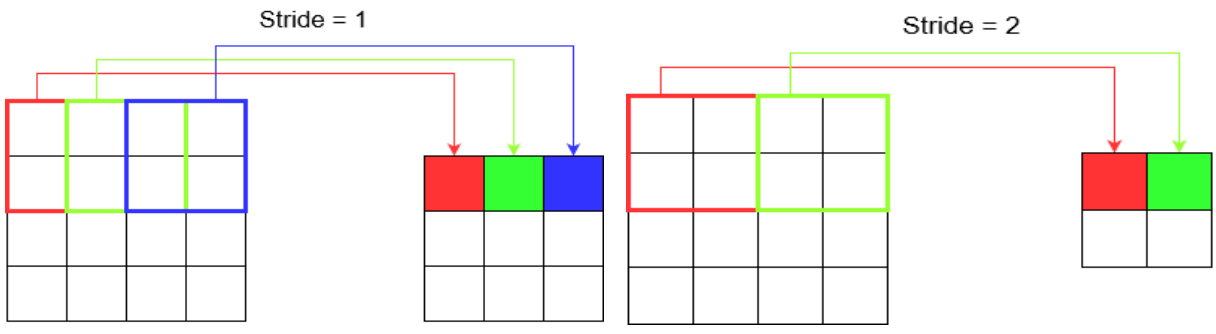


Figure 3-26: Striding in the Convolution Layer

Padding is the technique that solves the convolutional problem of shrinking the output and throwing away more relevant information, especially in the perimeter or edges of the image. Padding refers to adding some rows and columns to the image border by using fake pixels that are systematically added as zeros to ensure the output (feature map) matches the input dimension for maintaining dimensionality and preserving the original information. For this reason, it is also called zero-padding—expressed into three main types:

a. **Valid padding:**

Any convolution without padding is considered a valid convolution. The Valid padding (also known as no padding) assumes all the dimensions are 'valid'—that kernel slides inside over the input image and drops the convolution if the image dimensions do not align.

b. **Same padding:**

It appends zero values in the input border and ensures that the kernel slides over even outside the input image. It is known as the same padding because it guarantees the feature map (the output) will have the same dimension as the input. Mathematically it is given as:

$p=(\frac{(k-1)}{2})$ where k is the kernel's dimension.

**c. Full padding:**

It increases the size of the output by adding pixels (zeros) to the row and column of the image's border—and allows the kernel to slide over without dropping any convolution.

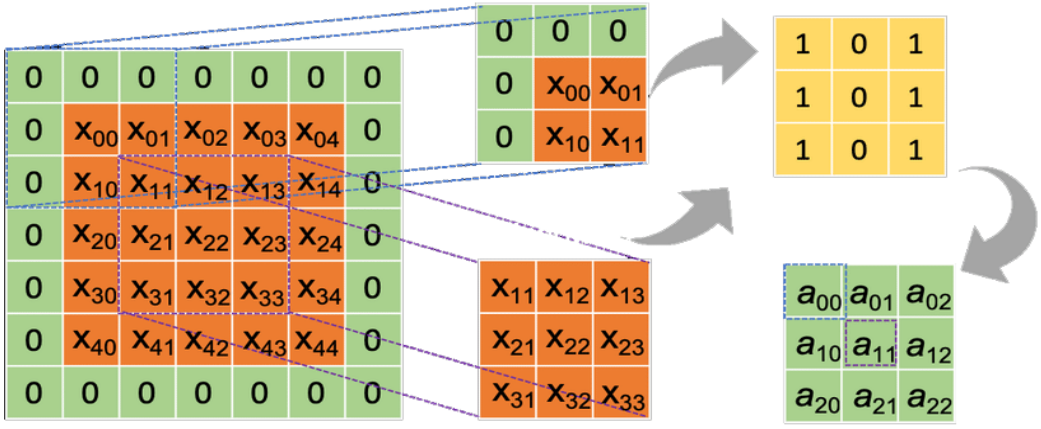


Figure 3-27: Padding in the Convolution Layer

Given the kernel 'k' slides over the input matrix of 'n' dimension with the padding 'p', the output matrix (feature map) will mathematically presented as: $[(\frac{n-k+2p}{s})+1]$.

To summarize, the kernel, stride, padding, and the number of kernels ($n_k$) are the hyperparameters in the convolution layer that determine the nature, dimension, and depth of the feature map. For example, a 3D matrix of ($n_h$ x $n_w$ x $c$) dimension with zero-padding *'p'* in which $n_k$ kernels of ($k_h$ x $k_w$ *x* $c$) dimension slides with the stride *'s'*. The output matrix will be given as:

$[(\frac{(n_h-k_h+2p)}{s}+1)$ x $(\frac{(n_w-k_w+2p)}{s}+1)$ x $n_k]$ where '$h$'=height, '$w$' =width and '$c$' =depth.

- **Non-Linearity (The activation Layer)**

The convolution layer only performs a linear operation. The activation layer has become crucial in achieving non-linearity operations that fire mechanism to decide which part of the input signal should pass to the next layer. Although the activation layer is a layer after the convolution operations, it is not among the main CNN layer because there are no learnable weight parameters. The activation layer consists of several types like Sigmoid, Tanh, Softmax, and Dropout. However, ReLu is common and is responsible for replacing all negative pixels with zero values.

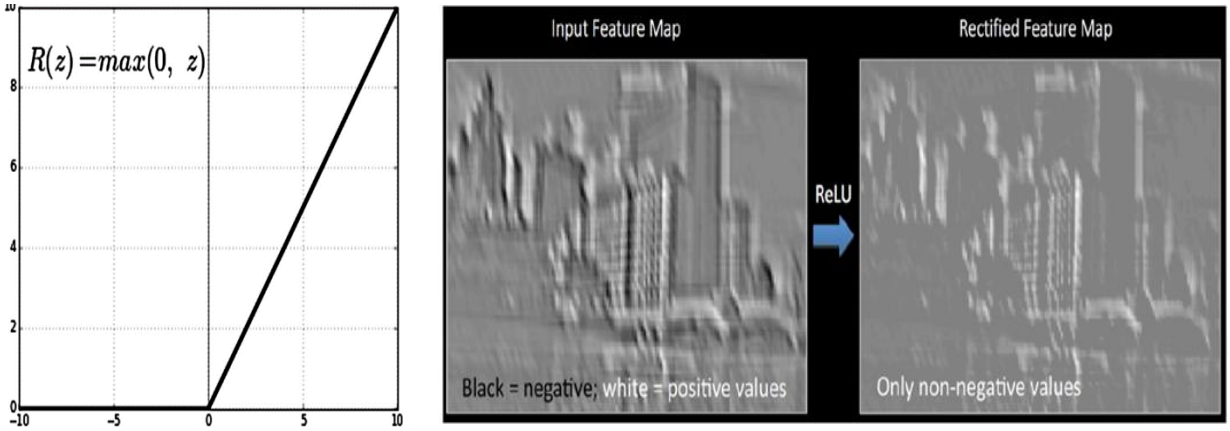


Figure 3-28: Feature map of CNN before and after passing through ReLU

### 3.3.2.2 The Pooling layer

The next layer after convolution is the pooling layer. A pooling layer receives the output from a convolutional layer and compresses it. The pooling layer filter is always smaller than a feature map. Apart from striding, pooling is another technique for reducing and managing the dimension of a feature map. It always gives a good and robust structure because it performs down-sampling to minimize the dimensional size of the feature map without throwing away relevant features—which means it maintains channel dimensions to retain crucial information.

The layer summarizes the statistics of nearby locations—that it introduces strong image properties like translation invariance. With high performance, a pooling layer also provides high-speed computations, saves memory, and reduces the overfitting problem.

Pooling has different types (like max, average, and sum)—however, max and average pooling are the prominent ones. These two types have the same goal, though they operate differently. The max pooling calculates the maximum values by identifying a spatial neighborhood and picking the highest dimensional element in the rectified feature map. Average pooling calculates the average of feature values in the field of the feature map. In practice, Max pooling is preferable because it outputs high accuracy. Compared to average pooling, which smooths out image pixels and loses the sharper features, Max pooling selects the brighter pixels of the image—even with darker backgrounds. Both operations—Max and Average pooling—depend on the two hyperparameters—the size of the window and the stride.

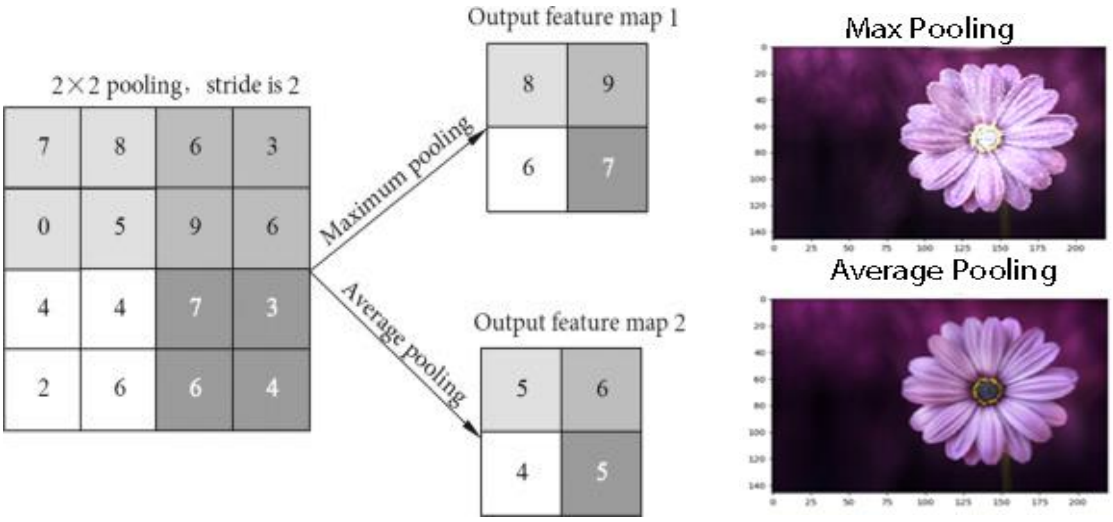


Figure 3-29: Pooling in the CNN

### 3.3.2.3 The Fully-connected layer

The Fully Connected (FC) layer is the final layer of CNN architecture. It resembles the regular Multi-Layer Perceptron (MLP). This layer directly connects the pixel values of the input image to the output layer —which means connecting each neuron in the previous layer to all neurons of the next layer—and the approach is the so-called full connection. The Full Connected layer consists of three interconnected layers: flatten (input), first, and output layers.

The flattened layer takes the output of the pooling layers and flattens them into a single vector ready for the next layer. The first full-connected layer inputs the output from the flattened layer and applies weights to predict the correct labels. The output layer is responsible for the classification or recognition. Unlike convolutional and pooling layers that use the 'ReLu' functions, the FC layer leverages a Softmax activation function for classification tasks.

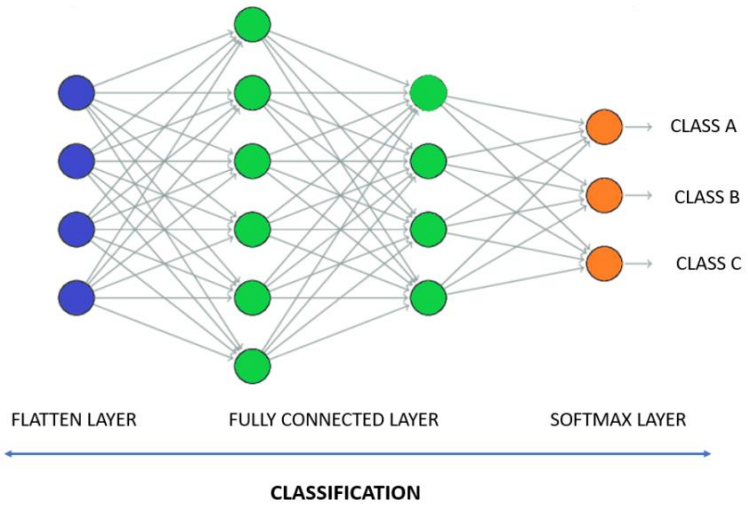


Figure 3-30: The full-connected layer in the CNN

### 3.3.3. The Types of CNN Architectures (The CNN's Case Studies)

Over the decades, since the 1980s, researchers are worked on connecting the CNN layers to achieve robust and well-performed CNN architectures. Until today, several studies are on research for modification. This section reviews the most common and popular CNN architectures where the MobileNet architecture is the proposed network in this research:

**LeNet-5:** LeNet was the earliest CNN architecture to promote the development of deep learning. The architecture's name was after a French Computer scientist Yann LeCun who combined the trained back-propagated CNNs to identify handwritten zip code numbers from US Postal Service. LeCun started this research in 1988 before achieving the LeNet-5 architecture with his colleagues: Leon Bottou, Yoshua Bengio, and Patrick Haffner in 1998. The goal of LeNet-5 architecture was to identify handwritten digits in a 32x32x1 gray image. It used five (5) layers with 60,000 trainable parameters in total. LeNet-5 had three (3) convolutional layers with no (valid) padding, two average pooling layers, and two fully-connected layers. It used sigmoid and Tanh activation functions and a softmax classifier at the output [69].

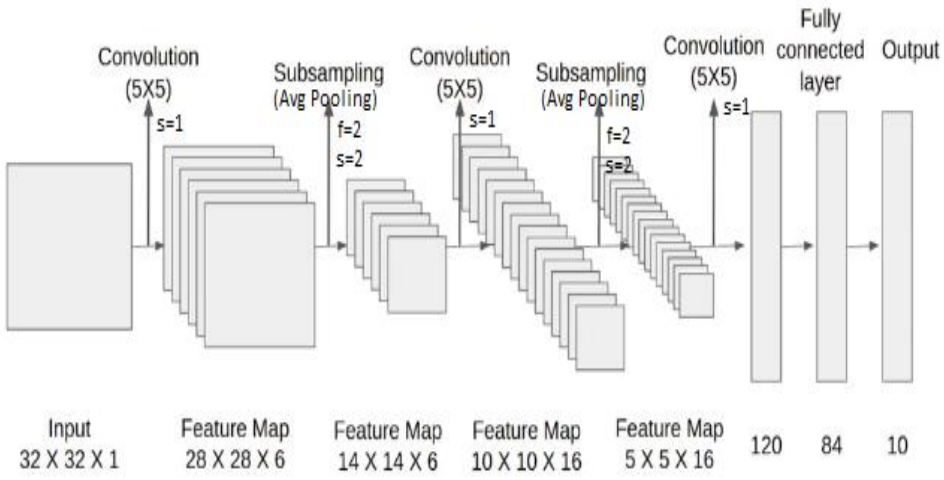


Figure 3-31: The LeNet-5 Architecture

**AlexNet:** This was a CNN architecture similar to LeNet—but bigger and deeper. Its name was after computer scientist Alex Krizhevsky who cooperated with Ilya Sutskever and Geoff Hinton in design. The goal of AlexNet was to classify images into 1000 classes which made it win the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) in 2012—with outstanding performance—outsmarted the second winner by far. This result made the CNN deep learning algorithm a breakthrough in Computer vision applications (including medical imaging classifications). It also justified the emergence of Artificial Intelligence due to computations via GPUs and Local Response normalization (RN).

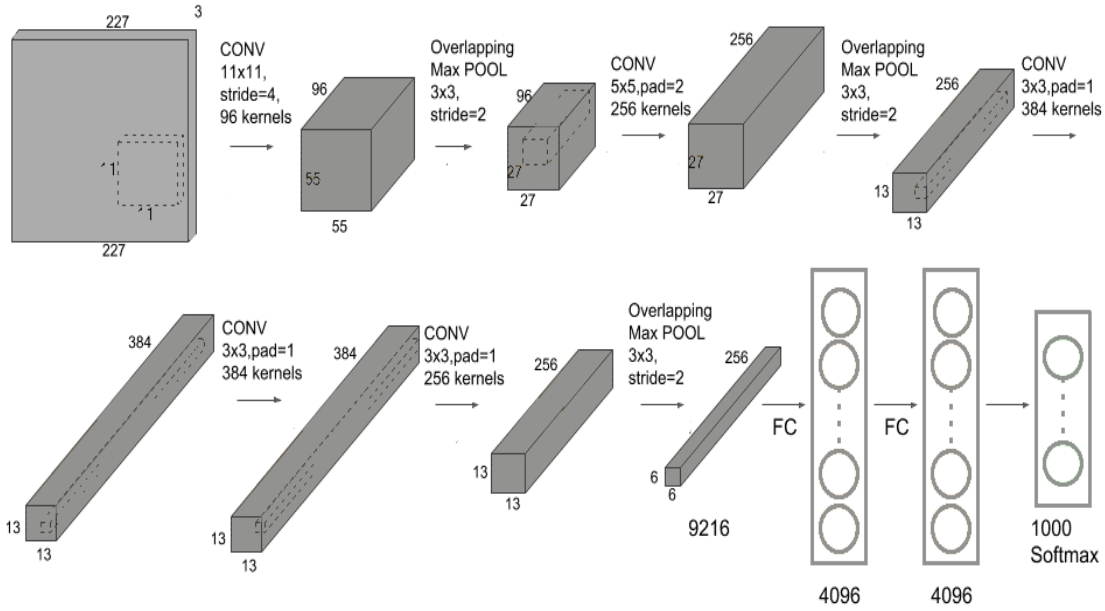


Figure 3-32: The AlexNet Architecture

The AlexNet architecture (as shown in figure 3-30 above) used eight layers with a total of 60 Million trainable parameters. It had five (5) convolutional layers with the same padding, three (3) Max-pooling layers, and three (3) fully-connected layers. AlexNet used the ReLu activation function with a softmax classifier at the output. In computations, the architecture used multiple GPUs to train 1.2 million images because a single GPU handled around 3GB of memory [70].

**VGGNet:** refers to Visual Geometry Group Network. It was the modification of the AlexNet architecture designed by Karen Simonyan and Andrew Zisserman. The network became popular during ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014—became a runner-up. The most used architecture is VGG-16 since instead of having a lot of hyperparameters, it uses a simpler network containing 16 convolution and fully-connected layers resulting from (3x3) convolution filters and (2x2) max-pooling with the same padding. The VGG-16 has around 138 million parameters—most in the fully connected layers—and requires a total memory of 96MB per image for only forward propagation. The main contribution of VGGNet was the notion that the depth of the network achieves the best performance, though it is limited to expenses and high-memory requirements [91].

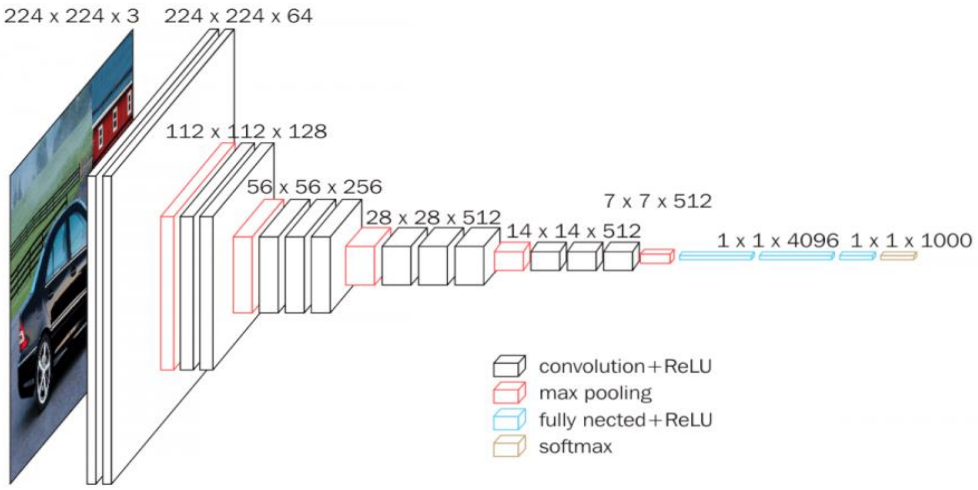


Figure 3-33The VGGNet Architecture

**GoogLeNet (Inception V1)**: was a 22-layer deep CNN designed through research conducted at Google (under a scholar Christian Szegedy with the collaboration of various universities). The architecture inspiration was from LeNet but implemented through the inception module—use multiple kernel filter sizes (instead of the single kernel) in the layer and concatenate the results to the next layer.

This network won the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) in 2014 — achieve very close to human-level performance—minimum errors compared to AlexNet, ZF-Net, and VGGNet. The introduction of the Inception Module dramatically reduced the network's parameters and computational expenses from 60 million (AlexNet) to 4 million due to using 1x1 convolution and global average pooling techniques to create a deeper network that reduces the problem of overfitting. The network also used batch normalization, image distortions, and RMSprop [92].

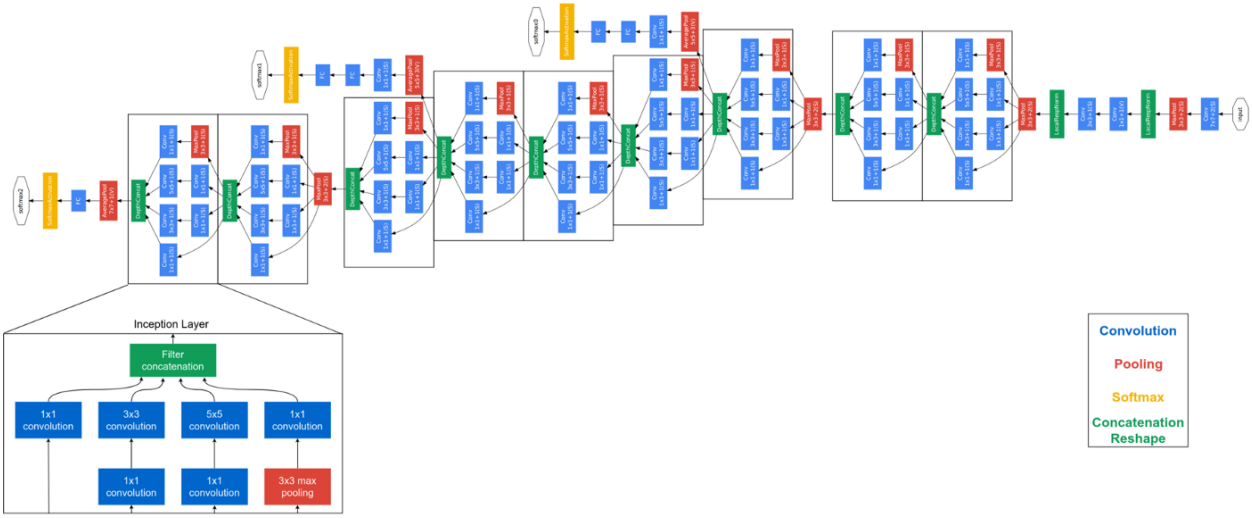


Figure 3-34: The GoogLeNet Architecture

**ResNet**: Residual Network is the CNN architecture designed by Shaoqing Ren, Kaiming He, Jian Sun, and Xiangyu Zhang. It is a network with more than 100 layers, using the same padding convolution without fully-connected layers. It won the Image uses the Net Large Scale Visual Recognition Challenge (ILSVRC) competition in 2015. The network introduced a so-called "identity shortcut connection"—implemented with one or more skipped layers that contain nonlinearities—the extensive uses of the ReLU and batch normalization in between layers. This approach makes it possible to train a thousand-layered network with compelled performance.

As deep networks are hard to train because of the vanishing and exploding gradient problems—the concerns of the previous architectures like AlexNet, VGGNet, and GoogLeNet, the ResNet blocks have provided the solution by skipping the layers that hurt the performance—by using regularization and same padding to maintain the dimensions. Due to its powerful representational performance, the architecture has become salient—not only to image classifications but also to other computer-vision fields like object detection and facial recognition [93].

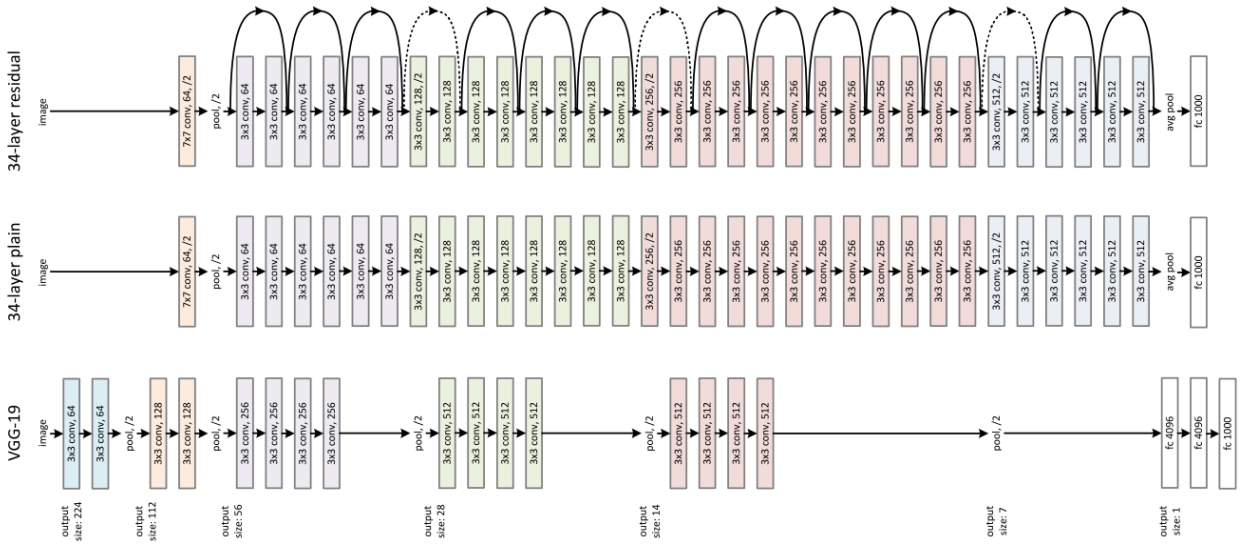


Figure 3-35: The ResNet Architecture

**MobileNet** is a type of CNN architecture and TensorFlow's first mobile computer-vision model designed for mobile and embedded vision applications. It is a Google open-source designed in 2017 by google engineers led by Andrew G. Howard. The MobileNet uses depthwise separable convolutions to significantly reduce the number of parameters to give an excellent starting point for training the classifiers that are smaller and faster than the network with the same depth performing regular convolutions. The Depthwise separable approach gives the idea of separating the filter's depth and spatial dimension. Thus, the network builds lightweight deep neural networks with low latency and lower power consumption [94].

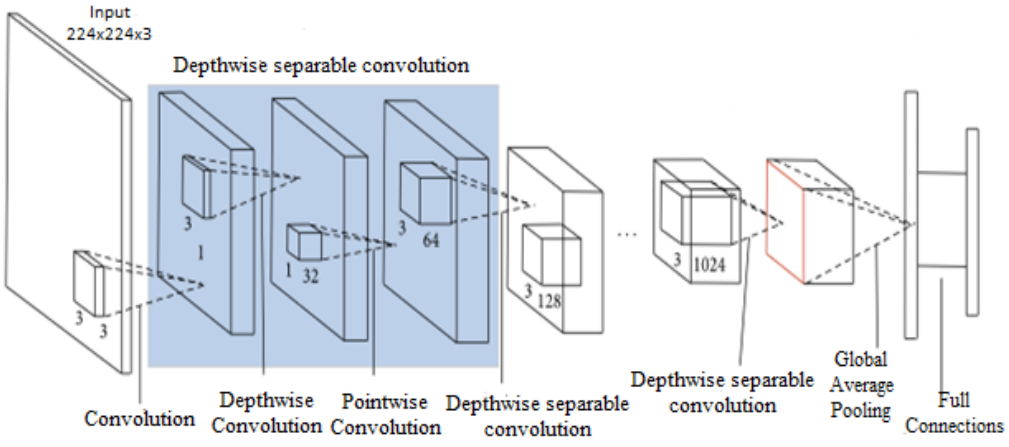


Figure 3-36: The MobileNet Architecture

Instead of performing the channel-wise and spatial-wise computation in a single step and transforming an image multiple times—like in traditional CNN, MobileNet architecture splits the operations into Depthwise and Stepwise convolutions. These operations reduce computational costs with high accuracy. Depthwise convolutions separate the input channels and perform a single convolution to each of them with the respective filter before stacking together their convolved outputs. Pointwise Convolutions use a 1x1 kernel (with the same depth as the input image) to iterate through every single point of the stacked Depthwise results. Whereas the output image dimensions in the MobileNet depend on the number of kernels in stepwise convolution [95].

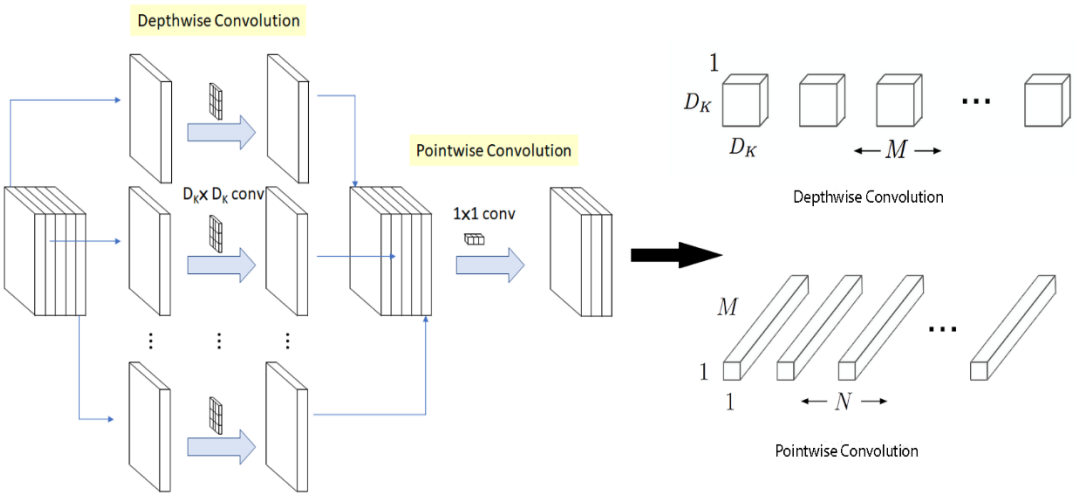


Figure 3-37: The Depth-wise Separable Convolution

Figure 3-37 above shows the depthwise separable convolution filters extract the spatial features from the input image without affecting the image depth. Each filter ($D_K$x$D_K$) iterates on only one channel of the input image and is stacked together to produce an intermediate output image ($D_F$x$D_F$xM). The N pointwise convolution filter (with a shape of 1x1xM) operates after depthwise convolution to increase the image depth—into a feature map ($D_F$x$D_F$xN)—the output image. The total computational costs in these operations will be the sum of individual computations from the Depthwise [($D_F$x$D_F$xM) x ($D_K$x$D_K$)] and the stepwise [($D_F$x$D_F$x N) x (1x1xM)], where $D_k$ is kernel/filter's dimensions (for height and width), $D_F$ is feature map dimension (height and width of intermediate output), N is the number of filters, and M is the number of input channels. The ratio between Depthwise separable convolution to Normal convolution is given as ($\frac{1}{N}+\frac{1}{M}$). This concept applies to MobileNet V1, which uses 3x3 regular convolutions to perform this operation—13 times.

### 3.3.5. MobileNet V2 Architecture

MobileNetV2 is a significant refinement of MobileNetV1. It also uses Depthwise separable convolutions—with 53 layers structured in 17 Residual-bottleneck connections—consisting of an inverted residual block (between the thin bottlenecks) and a linear bottleneck block (at the output without activation). This connection involves 3- layers: Expansion, Depthwise, and Projection. The Expansion layer uses a 1×1 convolution and expansion factor hyperparameter (6 by default) to expand input channels for more output data. The Depthwise convolution layer extracts spatial information with 32 lightweight filters. Unlikely V1, the stepwise convolution in V2 doesn't maintain or double the number of Depthwise channels—instead, it makes them smaller—projecting the high number of dimensional data to a low dimension tensor—hence called the Projection layer. The Expansion and Depthwise layers use the activation function (ReLU6) and Batch normalization. However, no activation in the projection layer: to preserve relevant information from low-dimensional data. With these bottleneck connections, V2 becomes faster, more efficient, and more powerful —with (2 times) less computational costs than V1 [96].

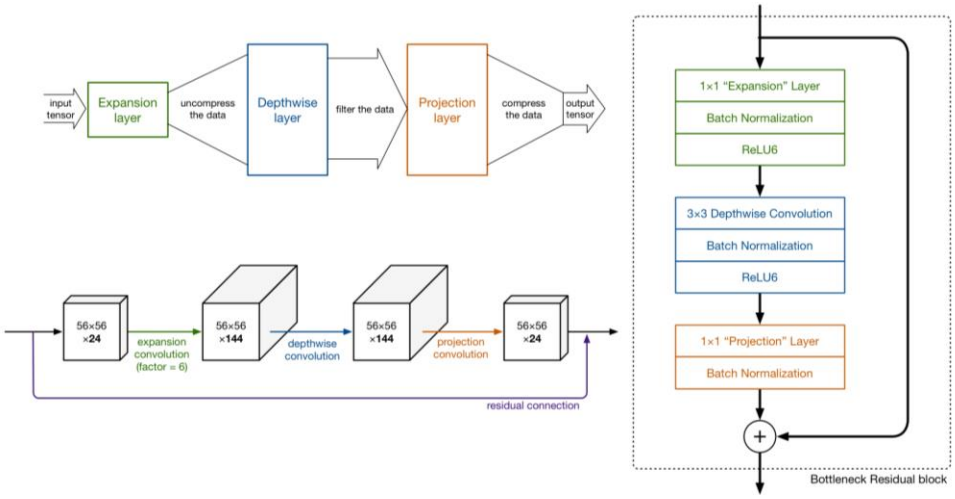


Figure 3-38: The MobileNetV2 Architecture

### 3.3.6. The limits of CNN

Despite achieving state-of-the-art performance in visual pattern recognition, CNNs do not have coordinate frames to keep tracking different features and orientations of an object. The CNNs don't recognize images as components rather than clusters arranged in distinct patterns. They are not effective in classifying objects at different positions. However, in this research, the classification doesn't depend on positioning. CNNs demands huge data structured in multiple layers –requires a long training and expensive GPUs. With operations like max pooling, CNNs are slower [71], [97].

## 3.4. Retinal Fundus Imaging in CAD

Since diagnosing cardiovascular risk factors is quickly, cheaply, and obtained safely (noninvasively) in an eye retina, the fundus imaging is a proposed approach to incorporate the CNN-deep learning algorithm in CAD prognosis [98]. The fundus image capturing is through a Fundus ophthalmoscopy or Smartphone camera—as indirect ophthalmoscopy.

### 3.4.1. Diagnosing CAD Risk Factors in Retina Fundus

- **Age and Sex**

Eye Retina features change with aging. Through the light-sensitive cell layers or nerve tissues, the accessible retina fundus can determine a person's biological age—as it provides evaluation possibilities of pathological processes of systemic vascular—through small vessels (microvasculature)—and neurological diseases. With the risks of CAD, the retina age—as proportional to a person's biological age become high—resulting in a big difference from a person's chronological age that increases death risks [99].

As male and female eyes are unalike, the retina also possesses different biological features influenced by sex: due to sexual hormone profiles (like estrogen). These features differ in layers' structure and cellular mechanisms—as identified with the help of fundus images [100].

- **Diabetes and Hypertension**

The fundus manifests diabetes through the microvasculature damages emerge as visual problems from abnormalities or fluid leakage of the blood vessels of light-sensitive tissue. That causes them to narrow, close, or enlarge to form balloon-like sacs—resulting in diabetic retinopathy. The fundus image can determine diabetic retinopathy based on the changes in retinal microvasculature features: retinal blood vessel area, exudes (swelling and formation of deposits), hemorrhages, microaneurysms, color, and texture [101], [102].

High Blood Pressure (Hypertension) also damages the blood vessels and optic nerve in the retinal fundus: which leads to vision loss from multiple adverse causes like hypertensive retinopathy (blood vessels damage), choroidopathy (fluid buildup), and optic neuropathy (nerve damage), or glaucoma (complete-damage of the optic nerve). Through Fundus images, it is easier and quicker to manifest these effects in an eye [103], [104].

- **Cholesterol and Obesity**

The cornea, iris, and retina detect cholesterol levels in the eye. In the retina, cholesterol appears in retinal blood vessels as fat deposition—as shown through fundus images. The high cholesterol levels in the eye can cause long-term vision loss when cholesterol breaks off a blood vessel wall and deposit a retina with a clot (Retinal vein occlusion). Cholesterol also causes diabetic retinopathy and age-related macular degeneration (AMD). The fundus images manifest these effects based on the retinal microvasculature features [105].

Obesity is excessive weight due to accumulated body fat caused by excessive food intake, lack of physical activity, genetic susceptibility, endocrine disorders, or medications—as measured by human Body Mass Index (BMI). In the retina, obesity manifests with thickness alterations of the retinal-choroidal microvasculature. It affects the retina and causes various ocular diseases: glaucoma, diabetic retinopathy, cataract, and age-related macular degeneration (AMD)—leading to vision loss. The fundus access these effects through fundus images: by realizing the diameter of retinal arterioles and venules [106], [107].

- **Smoking**

Due to the many oxidants in cigarette smoke, smoking has an anatomical and physiological effect on retinal circulation. It contributes deposit formation in the macula that results in age-related macular degeneration (AMD). It can also cause cataracts, glaucoma, diabetic retinopathy, retinal vein occlusion, and Dry Eye Syndrome—which causes vision loss. Smoking as a CAD risk factor can also manifest in the retina through fundus images by observing the changes in choroidal and retinal thickness, and macular functionality [108]–[110].

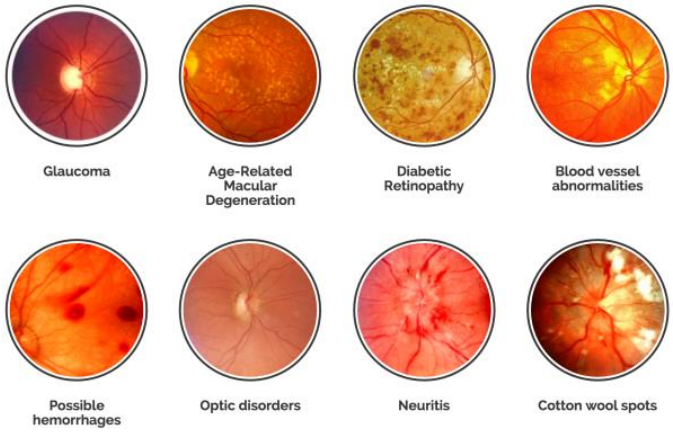


Figure 3-39: Unhealthy Retina shown by Retinal Fundus Images

### 3.4.2. Smartphones Fundus Photography

The advent of smartphone-based fundus imaging (SBFI) in 2010 has become a breakthrough in detecting retinal and systemic diseases: since the approach is portable, cost-effective, and applicable for both with and without dilation—compared to digital fundus photography. With this approach, documentation becomes easier with many settings that were impossible for fundus cameras. The process is viable since smartphones come with built-in connectivity and post-processing capabilities. The device's mobility and flexibility aid the examination of immobilized patients and are applicable in emergency medicine, pediatrics, also telemedicine [34].

- **Principles, Techniques, Devices, and Difficulties for SBFI**

The Smartphone fundus photography quality is comparable to standard fundus imaging. It also works with the same principle as indirect ophthalmoscopy as it uses a condensing lens where the phone display serves as a viewer's eye and its flashlight as a source of light (illuminating system), while the camera adjustment films the distance for the phone to focus on the retina images.

The process is through a smartphone camera application with a flashlight on with the camera changed into a video mode to record while filming the distance: by adjusting the lens forward or backward until the fundus fills the condensing lens area—focused for capturing. It is also possible to magnify the view of a retinal lesion by moving both the phone and lens toward the patient. Smartphones with a flashlight at backside of the phone near the camera and camera setting adjustment including exposure, light intensity, focusing, and zooming- can perform this process. These include android OS, iOS, Windows Mobile, Symbian, and Java operating systems.

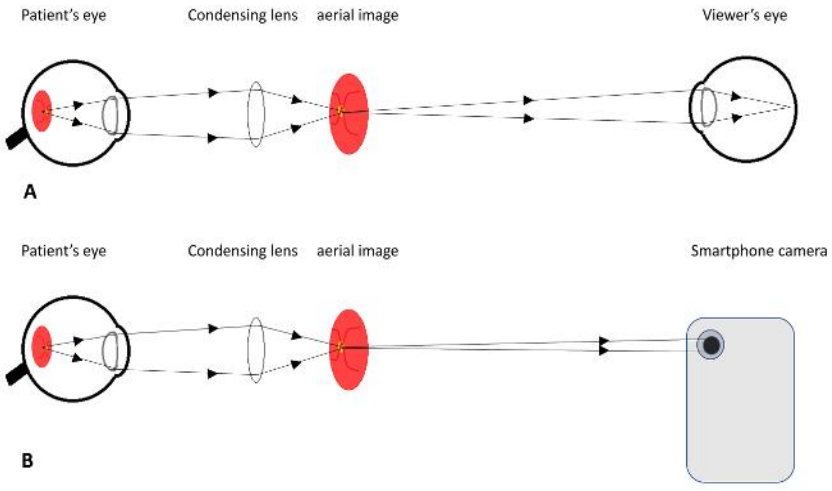


Figure 3-40: (A) Indirect Ophthalmoscopy (B) Smartphone Fundus Photography

In this process, beginners can face some difficulties while learning to adjust the filming distance—the first time. Also, it often becomes hard to perform this process using the picture camera mode—as the flashlight will not be continuous throughout the process. The glare may also occur in the pictures. And, since the image is the screenshot captured in the video, it will sometimes have low resolution: whenever using the oldest version smartphone model.

- **Safety Profile of Smartphone Flashlight for Retina**

Smartphones have different types of flashlights with various radiant power. Although these powers are safe for photography and videography, they may be a risk factor for damaging the retinal photoreceptors during fundus imaging. Despite many satisfying the safety limits profile for ophthalmic instruments set by ISO 15004-2.2, the latest models need more assessments.

## 3.5. Conclusion

This chapter has detailed the practical how of the whole research study by justifying deep learning as the appropriate technique over traditional machine learning—where the CNN is a proposed algorithm for CAD prognosis through medical imaging recognition—as it has outperformed human experts. The chapter has explored several CNN structures and highlighted MobileNet as the suitable architecture for designing an Artificial Intelligent prognosis model. It has also vindicated why retinal fundus images are a quick, easy and cheap approach to incorporating the CNN algorithm in observing CAD risk factors through an eye. The chapter has also declared that, despite CNN's limitations, an algorithm is a suitable approach for this research.

# CHAPTER 4:
# EXPERIMENTAL VALIDATION:

## 4.1. Introduction

The chapter validates the proposed MobileNet V1 CNN architecture and experimental environment setups—the Databases, Frameworks/Libraries/APIs, Computational architecture (GPU), IDEs, and Project Repositories. It explores the model designing and the diagnostics techniques: to analyze errors, evaluate the model performance, and suggest the best methods for model optimization. The chapter also presents the model designing system, experimental procedures, and the obtained optimal results.

## 4.2. Experimental Environment

In this project, TensorFlow Framework '2.11.0' and MobileNetV2 architecture—with a 16GB NVIDIA® GeForce RTX 3070 GPU—are set for designing and model training. The distribution of train/dev/test sets is from several databases containing retinal fundus imaging. The Jupiter notebook IDE on Anaconda '22.9.0' platform—with various imported libraries— is also set for code files that are stored and tracked by distributed version control system—GitHub.

### 4.2.1. TensorFlow Framework

Compared to PyTorch, Chainer, Apache MXNet, Caffe, and Microsoft CNTK, Tensorflow has recently become the popular library for machine learning applications—from researchers at universities to developers in big tech companies or startups. Being an open-source framework for constructing neural networks, TensorFlow has contributed to the democratization of deep learning and evolved the AI industry since it uses symbolic mathematics to perform operations like automatic differentiation on a computational graph instead of purely numerical computations. Google Brain team developed this framework from an earlier proprietary framework called DistBelief and released the first version as TensorFlow 1.0.0 in 2017. The latest release of version 2 (2.11.0 for this project), which uses Keras API as the default high-level abstraction, has aided the construction and customization of neural networks in a simplified numerical implementation—with a few lines of codes: for achieving optimal results in a faster and more intuitive algorithm implementation [111].

Since TensorFlow can perform computations on GPU hardware, it potentially speedups up the operations. TensorFlow has the fundamental feature of developing and deploying models in multiple platforms and environments: web browsers, mobile apps, embedded devices, and large-scale production environments. The TensorFlow ecosystem supports development in Python, JavaScript, or Swift—with data preprocessing, model building, and training pipelines. That means it aims to ensure the portability and scalability for paving the way toward a future where any device could apply neural networks through IoT sensors or cloud servers. With an active community of contributors, forums and user groups, blogs, and YouTube channel—with tutorials, presentations, and interviews, TensorFlow gives the platform for a group of developers to share codes and tackle problems together in application areas as varied as medicine, NLP, finance or computer vision.

### 4.2.2. Databases

In this research study, retinal fundus images (as train/dev/test sets) are the dataset collected from various publicly distributional databases. That is because: it is not always possible to achieve the required amount of data from a single distribution. The famous databases are as follows:

**DRIVE Database:** The Digital Retinal Images for Vessel Extraction (DRIVE) database consists of 8 bits, 40 JPEG compressed retinal images with (768 x 584) pixels captured using a Canon (CR5 non-mydriatic 3CCD) camera at a 45-degree field of view (FOV). It provides the platform for comparative studies on the segmentation of retinal blood vessels (such as length, width, tortuosity, branching patterns, and angles). DRIVE contains (33) healthy images and (7) affected images (showing signs of mild early diabetic retinopathy). These images are from the assessed 400 diabetic subjects aged between (25-90) years old [112].
(https://drive.grand-challenge.org/)

**STARE Database:** Structured Analysis of the Retina (STARE) was a project introduced in 1975 by Michael Goldbaum at the University of California, San Diego. The database contains 400 raw TIF format images created by scanning and digitizing the retinal image photographs captured by a narrow field of view of 35 degrees camera—hence having low quality with (700 x 605 pixels) resolution. STARE includes images of healthy and abnormal retinas affected by various diseases such as Hypertensive Retinopathy, Diabetic Retinopathy, Retinal Vein Occlusion, and Retinal Artery Occlusion [113].
(http://cecas.clemson.edu/~ahoover/stare/index.html)

**IDRiD Database**: Indian Diabetic Retinopathy Image Dataset (IDRiD) is a retinal image database which recently published to evaluate the performance of developed algorithmic systems for automatic detection and grading of Diabetic Retinopathy (DR) and Diabetic Macular Edema (DME). The database consists of JPEG (4288 X 2848) high-resolution pixels images. The 516 data (413 trainsets and 103 test sets) contain images with manually marked OD center and fovea locations, and the 81 data (54 trainsets and 27 test sets) with manually segmented optic-disc boundary ground truth versions. These Images are centered near the macula and captured using a Kowa VX-10 alpha digital fundus camera with 50 degrees of field of view (FOV) [114], [115].
https://idrid.grand-challenge.org/

**Smartphone-based Retinal Image Montaging Data** is a database with 16 retinal fundus image sets—captured by indirect ophthalmoscope (oDocs nun 0phthalmoscope and nun IR fundus camera). The 'reconstructions' folder with these images is montaged using the oDocs state-of-the-art montaging algorithm.
https://drive.google.com/drive/folders/16Keaq6bK-5Hb_cPwmXQtxrBtUwoaxdue

**CHASE Database:** Child Heart and Health Study in England (CHASE) is a retinal image database with reference datasets acquired from the cardiovascular health survey in 200 primary schools of multi-ethnic children—in London, Birmingham, and Leicester. The database contains 28 images (with 999x 960 pixels) taken using a hand-held NM-200-D fundus camera at 30 degrees of a field of view [116].
https://www.medicmind.tech/store

**EyePACS Database** is a non-proprietary digital image system—a web-based application—for storing and managing retinal images. It provides a large public dataset of high-resolution retinal images from diabetic screening programs. EyePACS contains a total of 35127 images—labeled by clinicians according to the severity of Diabetic Retinopathy (DR)—graded as 0 (No DR), 1 (Mild), 2 (Moderate), 3 (Severe), and 4 (Proliferative DR). The database consists of 25810 healthy images, 2443 images with mild Non-Proliferative Diabetic Retinopathy (NPDR), 5292 with moderate NPDR, 873 with Severe NPDR, and 708 with Proliferative Diabetic Retinopathy (PDR). It is also available on the Kaggle database since the competition for the automatic detection of Diabetic Retinopathy in 2015—sponsored by the California Healthcare Foundation [117].
(https://www.kaggle.com/c/diabetic-retinopathy-detection/data)

**MESSIDOR Database: The** Methods to Evaluate Segmentation and Indexing techniques in the field of Retinal Ophthalmology (MESSIDOR) was a research program funded by the French Ministry of Research and Defence under the 2004 TECHNO-VISION program to facilitate the studies on computer-assisted diagnoses of diabetic retinopathy for comparing and evaluating various segmentation algorithms developed for the detection of lesions in color retinal images, and Tools for indexing and managing image databases. The database consists of 8 bits, TIF 1200 colored retinal fundus images of the posterior pole captured using a 3CCD camera (with a Topcon TRC NW6 non-mydriatic retinography) having a 45-degree field of view (FOV). The image sizes are in three different resolutions: (1440×960), (2240×1488), and (2304×1536) pixels—where each image diagnoses two types of diseases: Retinopathy grade and Risk of macular edema [118]. http://www.adcis.net/en/Download-Third-Party/Messidor.html.

### 4.2.3. Computational Architectures, IDEs and Project Repository

All the codes and project files in this project are written on the Jupyter Notebook IDE—imported with various python libraries—and locally accessed on Anaconda '22.9.0'. Through CUDA, the Jupyter Notebook runs on NVIDIA® GeForce RTX 3070 with 16 GB memory: to speed up the computational processes—especially for high-resolution images and visual data. GitHub provides the repository for the codes and files to store and track the changes.

The Jupyter Notebook is a proposed IDE since it is a web-based application providing an interactive computational and integrated development environment: for creating, editing, and sharing computational documents. The IDE yields standard data analysis and evaluation—as it can combine code executions, rich texts, mathematics, visualizations, and rich media. Moreover, the notebook is versatile and flexible to be executed on a local machine or installed on a remote server while accessed through an internet connection. As GPU uses Single Instruction, Multiple Data (SIMD) architectures, it has become salient in this research for performing multiple and simultaneous computations that require the same processes for numerous data—which means accumulating many cores with fewer resources and low power. GitHub is also crucial to provide a web-based distributed version control system—the Git repository hosting services—that not only stores and tracks the changes but also enables private and public sharing on integrated platforms like Amazon and Google Clouds.

## 4.3. Model Diagnostic in DL

Achieving an optimal ANN model is an iterative (empirical) process that requires several steps of training, experimentation, analysis, and evaluation. It involves implementing various ideas: collecting data and selecting the architecture or hyperparameters to train (code) the model before experimenting or diagnosing to evaluate and analyze the model performance for deployment.

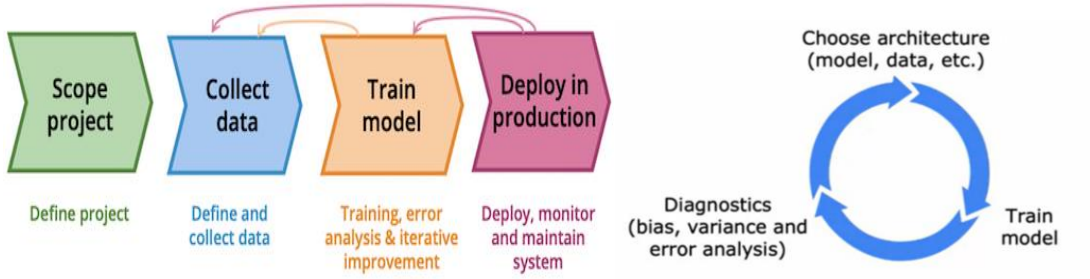


Figure 4-2: Life Cycle and Iterative Loop for ML Development

In Deep Learning, Model diagnostic refers to a test performed to analyze and evaluate the model performance to gain insight that guides how to improve its performance. It includes splitting data into train/dev/test sets to diagnose the bias and variance or performing error analysis before evaluating the model accuracy/recall /precision.

### 4.3.1. Training, Development, and Test Sets (Train/Dev/Test Sets)

A training dataset refers to a set of data examples or samples used during the learning process for a model to fit them. The Development (Cross-validation) and Testing datasets are independent of the training dataset and used to provide an unbiased evaluation of the fitted model on given training datasets. A Cross validation dataset evaluates the model on the training set while tuning that model's hyperparameters. It is also crucial in model selection—the less dev error, the better the model. A testing dataset accesses the final model performance before deploying it for real-world applications. Although the development and test datasets must come from the same distribution, it may not be necessary for the training dataset. The training dataset may come from different distributors since it may not be possible to collect enough data from the same distribution—that means the train set and the dev/test sets may sometimes differ. However, the data mismatch technique may mitigate this difference. The sizes and strategies for splitting datasets into train/dev/test sets depend on a given problem or the amount of data available. Thus, the (70% training, 30% test) or (60% training, 20% dev, 20% test) dividing approach was valid for training dataset (<100000), while in Deep Learning with a million or more examples, a reasonable split

would be (98% training, 1% dev, 1% test). These train/dev/test dataset phases always influence the model selection—since the chain assumption of the created DL model is to perfectly generalize (fit well) on the train/dev/test datasets and then in real-world examples [119].

### 4.3.2. Bias, Variance, and Error analysis

#### 4.3.2.1. Bias and Variance

An inaccurate DL model gives errors during predictions. These errors are known as 'reducible' and 'irreducible' errors. Irreducible errors are present in a model due to unknown variables with unreduced values. Reducible errors have variables with values that are viable to further reduce for improving a model. These errors are known as Bias and Variance [120].

Bias is the systematic error resulting from the difference between actual and predicted values. It happens when the model is too simple or undertrained to learn complex patterns—resulting in underfitting problems—the model is unable to capture the patterns in the given datasets—leading to higher training and validation errors compared to the overall cost function**.** Variance is a random error that specifies the degree of variation when using a different dataset. It happens when the model is too complex (many polynomial features) or overtrained—generalizes on training datasets but fails on the new examples. This error results in overfitting problems—the model captures noises in the given datasets and fails to generalize on both the validation and test sets (high errors on the dev/test set compared to the overall cost function).

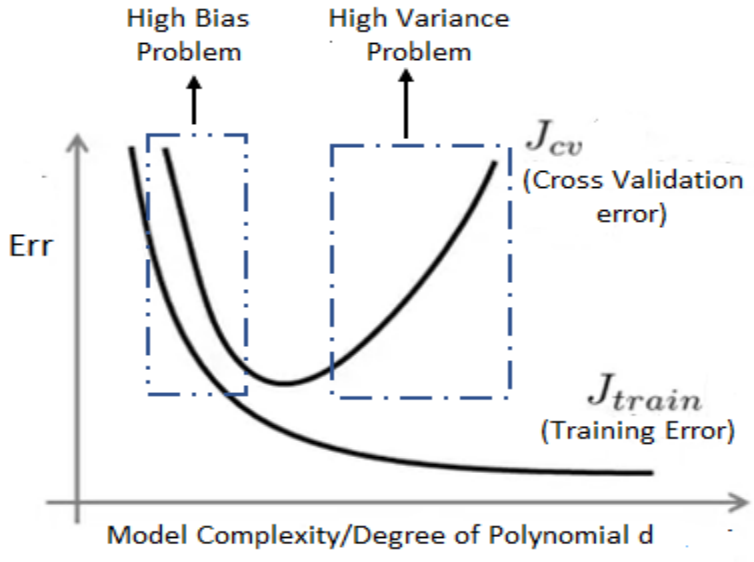


Figure 4-3: The Bias and Variance

The modal can have low bias and high variance resulting in overfitting problems, and high bias and low variance resulting in underfitting problems. It can also have high bias and high variance when Predictions are inconsistent and inaccurate targets on average and have low bias and low variance for an optimal model—which determines consistent predictions and accurate targeted output.

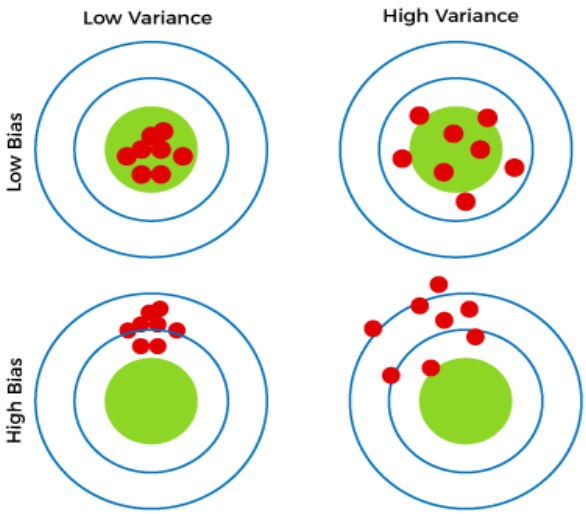


Figure 4-4: Different Combination of Bias and Variance

- **Learning Curves for Bias and Variance**

A learning curve is a tool for measuring the model performance: diagnosing if it suffers from either bias or variance. It plots training and validation curves to show the relationship between the training and the cross-validation errors against the given training dataset: to determine how much the model benefits from more data and how sensitive concerning bias or variance [121].

A learning curve determines high bias or high variance by examining the gap between training and cross-validation errors. This gap will be small if both errors are high—indicating the model has high bias and hence underfitting. The learning-curve gap will be huge when a cross-validation error *(Jcv)* is high compared to a training error *(Jtrain)*—meaning the model has high variance and hence overfitting. Figure 4-4 below shows the cross-validation error decreases when training data increases. It also indicates that adding more training datasets won't improve the model's performance since the training error will become a plateau—flattening the curve—suggesting the model has failed to learn anything —as it is a simple (linear function) model.

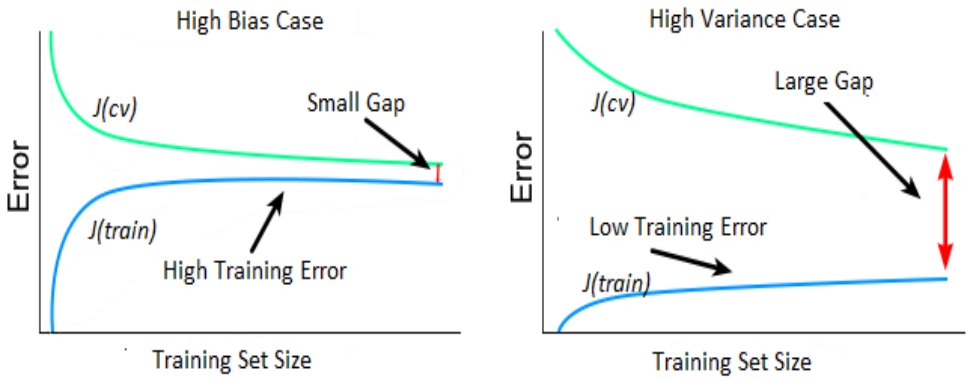


Figure 4-5: The DL Learning Curves

- **Bias-Variance Trade-Off**

For a model to be optimal with accurate prediction, it requires low bias and low variance attained simultaneously. But this is not always possible since bias and variance are interrelated—decreasing the 'bias' increases the 'variance' and vice versa. The balance between the two is only possible through the Bias-Variance trade-off that finds the sweet spot to ensure that the model captures the essential patterns while ignoring the noises simultaneously [122], [123]. Figure 4-6 below indicates that achieving bias-variance trade-off is through balancing the bias and variance in a way that the total error becomes minimum—as mathematically expressed as:

$$Total\ Error = Bias^2 + Variance + (Irreducible\ Errors/Noises)^2$$

$$\mathbf{E}\left[\left(y - \hat{f}\,(x)\right)^2\right] = \mathbf{Bias}\left[\hat{f}\,(x)\right]^2 + \mathbf{Var}\left[\hat{f}\,(x)\right] + \sigma^2$$

Where:

$$\mathbf{Bias}\left[\hat{f}\,(x)\right] = \mathbf{E}\left[\hat{f}\,(x) - f(x)\right]$$

and

$$\mathbf{Var}\left[\hat{f}\,(x)\right] = \mathbf{E}[\hat{f}\,(x)^2] - \mathbf{E}[\hat{f}\,(x)]^2$$

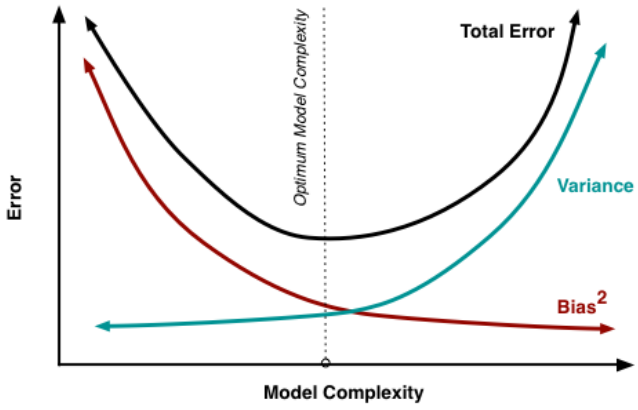


Figure 4-6: The Bias-Variance Trade-Off (Optimal Model Complexity)

- **Regularization and Bias/Variance**

Regularization ($\gamma$) can influence the variance and bias of the model. The very large lambda will make weights smaller, approximated to zero—that the model becomes too simple to learn, resulting in high bias—hence underfitting. The very small lambda will make the model complex with many features—resulting in high variance—hence overfitting. With a good lambda selection, the regularization can act as a bias-variance trade-off—simultaneously achieving a low bias and low variance model. Plotting train/dev errors against the lambda values—as shown in figure 4-7—will give insight into the proper value of lambda that can achieve the optimal model [124].

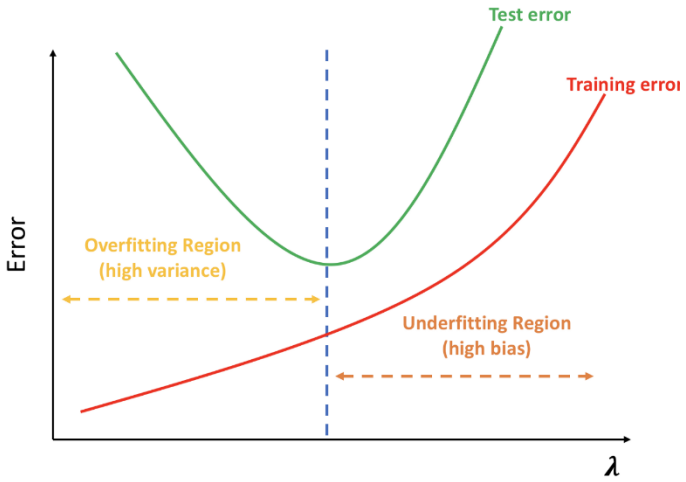


Figure 4-7: Bias and Variance with Regularization

- **Establishing Baseline Level of Performance**

The baseline level of performance is crucial to measure how well the model performs. It establishes a reference point to determine the level of reasonable errors for the model to be considered optimal: by comparing its bias and variance levels. Among the best and most common ways to establish this baseline is through comparing to human-level performance, competing with already accomplished models, and through prior experience.

**Human-level performance:** Humans are excellent at natural perception tasks associated with unstructured data like images, audio, natural language, and text. Despite the advancement of AI, Machines cannot surpass humans in performing these tasks. It turns out that the workflow of designing and building an AI system is much more efficient when trying to perform tasks that humans can also do. Hence, the human-level performance becomes a perfect baseline for measuring the model's accuracy.

**Bayes Optimal Error:** This is the lowest possible error for any classifier used as baseline accuracy for problems where the AI system has significantly surpassed the human-level performance—mostly in structured tasks like online advertisements, product recommenders, Logistics (estimating transit time), and Loan approval. In medicine, AI has also outclassed human-level performance. However, no model can surpass the Bayes error—unless it is overfitting.

**Avoidable Bias and Variance:** the established baseline can determine the level of bias and variance. Avoidable bias is the difference between the training error and human or Bayes error (Avoidable bias = Training error – Human/Bayes error). Variance is the difference between development and training errors (Variance = Dev error - Training error). The model is said to underfit when the avoidable bias is larger than the variance and overfit when the variance is higher than the avoidable bias.

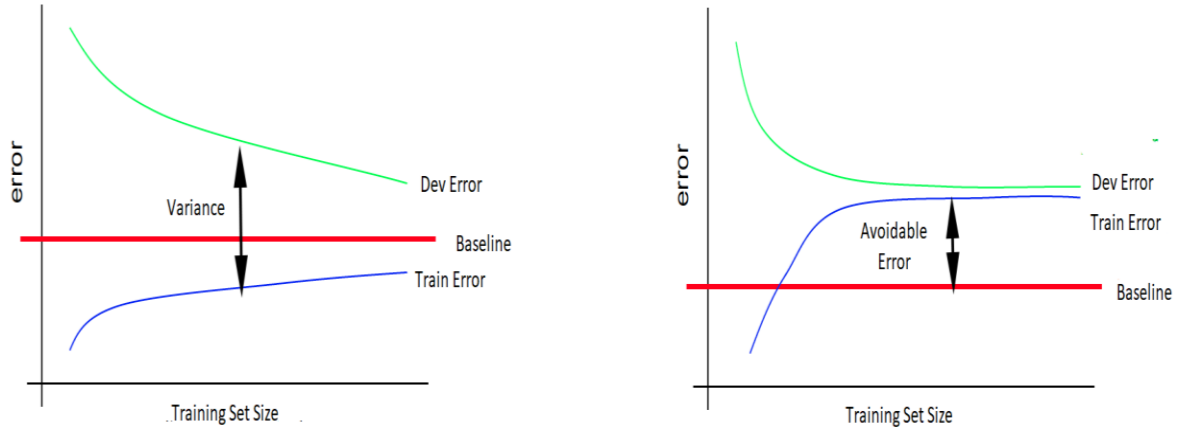


Figure 4-8: Baseline Level Performance

- **Bias and Variance with mismatched data distributions**

For matched data—the data from the same distribution—it is common to split data into three phases: train/dev/test sets. However, this technique doesn't work for mismatched data—data from different distributions—since the training set and dev/test set have distinct characteristics—leading to improper bias and variance analysis. The complication in analyzing bias/variance error may be either one set has become easier or harder to train that error is unrealistic [125].

Since DL algorithms require a vast amount of training datasets, considering mismatched data distributions is inevitable—since obtaining enough data from a single distribution is not always possible. There are some strategies employed for dealing with data from different distributions.

These strategies include combining data to belong to the same distribution and randomly shuffling them by splitting them into the training and dev/test sets. The drawback of this strategy is that the target distribution will have less data hence spending more time optimizing untargeted data. The better approach to perform well on the target distribution is by adding some of the dev/test set examples to the training set —hence the distribution now will have four splits: the train, train-dev, dev, and test sets. Yet, with this strategy, there will be mismatched problems since the training distribution will differ from the dev/test distribution means it will require a longer time and more effort to optimize the model since training untargeted distribution for the most part.

**Data-Mismatch Problem:** By creating a new set called the train/dev set, the variance/bias analysis becomes viable. Through a baseline, it becomes possible to calculate avoidable bias by comparing the baseline error and training error. The difference between the training and train/dev errors gives the variance. And the difference between train-dev error and dev-test error indicates a Data-mismatch value. When the Data mismatch value is much greater, then the Data-mismatch problem emerges: which means the model is not optimal.

**Addressing Data Mismatch Problem:** Unfortunately, there aren't many or completely systematic solutions to the data mismatch problem. Among the strategies for mitigating the problem is to incorporate the characteristics of the dev/test and the train sets by trying some approaches like carrying out manual error analysis to manually examine the difference between training and dev/test data. Another technique is to make training data more similar to dev/test data or to collect more similar data to dev/test sets which can be achievable through artificial data synthesis technique.

- **Minimizing the Bias and Variance**

Since variance and bias are reducible errors, by minimizing these errors, achieving an optimal model becomes viable. The techniques to reduce high variance include: decreasing regularization, making a complex model by adding more features, building proper architecture with well-tuned hyperparameters, or increasing the duration of model training using better optimization algorithms (like Rms prop and Adam). The techniques for reducing high variance include: collecting more training data, increasing the regularization, making a simpler model with fewer or relevant features, and implementing early stopping to avoid overtraining the model [123], [126].

#### 4.3.2.2. Error Analysis

Error analysis refers to manually examining the training examples and getting insight into what or where the algorithm is poor. It is also a diagnostic procedure that aids in early identifying the errors before an action or decision that could unnecessarily cost a lot of time. With this approach, it becomes easy to evaluate multiple ideas by creating a grid that helps to identify and select the best idea to improve the model's performance. Error analysis is a very salient procedure in machine learning. However, it only performs well on the tasks that humans can perform.

Error analysis not only can address the mismatched data problem but is also used to identify incorrectly labeled data from the same distribution. The technique gives insight into improving the issue by analyzing mislabeled data in the training or dev/test sets. Since DL algorithms are robust in random errors, considering the systematic error is the key to cleaning up mislabeled data in the training set. While in dev/test sets, correcting the mislabeled data depends on the error percentage of mislabeled data to overall dev/test error compared to the error from other causes. That means if the mislabeled data error is higher than other errors in created error-analysis grid, correcting the error worth it.

### 4.3.3 Model Evaluation Metrics

Despite data preparation and model training being the key steps in building a DL model, they can't guarantee an optimal result. Hence, it is necessary to measure the performance of the trained model to evaluate how it generalizes to new or unseen data. Depending only on the accuracy won't improve the overall predictive power or determine the success of the model rather than leading to poor performance when deployed to new data. So, there is a need to evaluate the quality of the model from several metrics—known as performance or evaluation metrics. These metrics quantify the model performance to measure how it generalizes to the new data. They also play a role in monitoring the model. The popular and common metrics are Confusion Matrix, Classification Accuracy, Precision, Recall, Logarithmic loss, Specificity, F1 score, and AUC/ROC [127].

#### 4.3.3.1 Evaluation Metrics Terminologies

- **Confusion Matrix**

A confusion matrix, also known as an error matrix in supervised learning and a matching matrix in unsupervised learning, is a tabular layout that visualizes the model performance—as represented in the NxN matrix—where N is the number of predicted classes or categories.

The matrix represents the actual values instances in a row and predicted values in a column or vice versa. It has a name confusion matrix since its theories and terminologies are confusing. These terminologies are as follows:

**True Positives (TP):** the case where the model predicts Yes and it's true in reality.
**True Negatives (TN):** the case where the model predicts No, and the actual output is No.
**False Positives (FP):** the case where the model predicts Yes, but the actual output is No.
**False Negatives (FN):** the case where the model predicts No, but the actual output is Yes.

| | | **Predicted** | |
|---|---|---|---|
| | | Negative **(N)** - | Positive **(P)** + |
| **Actual** | Negative - | True Negative **(TN)** | **False Positive (FP) Type I Error** |
| | Positive + | **False Negative (FN) Type II Error** | True Positive **(TP)** |

Table 4-1: Confusion Matrix

- **Accuracy**

Accuracy is the evaluation metric for classification problems that measure systematic observation error to determine how often the classifier is correct to a 'true' value. It calculates the proportion of the total number of 'correct' predictions—the ratio of the number of 'correct' predictions to the total number of predictions.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ number\ of\ Predictions\ made} = \frac{TP+TN}{TP+TN+FP+FN}$$

The accuracy metric is very intuitive, understandable, and easy to implement. It measures the range from 0 to 100 percent or 0 to 1: which becomes useful for simple modeling cases. The metric is applicable in almost all ML libraries. However**,** it heavily relies on data specifics—that it works well on the balanced data—the classes that are roughly equal in size or data with an even number of samples belonging to each class.

- **Precision (or) Positive Predictive Value (PPV)**

Precision/PPV is the evaluation metric that measures random observation errors to determine how close the measurements are to each other. It explains how many correctly predicted cases turned out to be positive by calculating the ratio of correct 'positive' results to the total number of 'positive' predictions (true positive and false positive).

$$Precision = \frac{Number\ of\ True\ Positive}{Total\ Positive\ Prediction} = \frac{TP}{TP+FP}$$

The precision metric has overcome the limitation of the accuracy metric since it is capable of dealing with imbalanced data. However, the metric is more useful in cases where a False Positive is of more concern than a False Negative.

- **Recall or Sensitivity or Total Positive Rate (TPR)**

Recall or sensitivity is an evaluation metric similar to the Precision metric as it provides a better way of evaluating model performance with imbalanced data. However, the metric aims to calculate the True Positive Rate (TPR), which determines the proportion of actual positive cases out of all positives the model could identify. That means unlikely precision metric Recall can determine all missed 'positive' predictions. Mathematically, recall is the ratio of all True Positives to the sum of all positives (True Positives and False Negatives) in the dataset.

$$Recall = \frac{Number\ of\ True\ Positive}{Total\ number\ of\ (actual)\ all\ Positive} = \frac{TP}{TP+FN}$$

- **F1-Score**

Recall and precision metrics separately don't always determine the best algorithm in some cases where the classifier will have to be precise and sensitive simultaneously. Thus, calculating these metrics into a single metric will better evaluate the model performance. While taking the arithmetic mean of these metrics could achieve the goal, there is a decent chance of getting wrong predictions since the recall and precision values are not on the same scale. Hence, the Harmonic mean is the better approach to provide a tradeoff because it emphasizes small values and punishes extreme values. The intricate metric that can achieve a Harmonic mean is known as F1-Score. This metric simultaneously balances the recall and precision and measures a test's accuracy within a range of 0 and 1 values. Thus, the greater the F1 Score, the better the performance of our model.

Mathematically, the F1-Score is as follows:

$$F1\text{-}Score = \frac{2}{(\frac{1}{precision})+(\frac{1}{Recall})} = \frac{2\ (precision\ x\ Recall)}{Precision+Recall} = \frac{2\ TP}{2TP+FP+FN}$$

Ideally, the F1 Score is an effective metric in classification scenarios: when False Positive (FP) and False Negative (FN) are equally costly — that means they are missing true positives or false positives; the case where adding more data doesn't effectively change the outcome; and the when True Negative (TN) is high and uninterested.

- **Specificity or Selectivity or True Negative Rate (TNR)**

Specificity or Selectivity is the opposite of the recall metric. It is an evaluation metric that aims to calculate the True Negative Rate (TNR), which determines the proportion of actual (true) negative cases out of all negatives the model could identify. Thus, if the results are negative, how often the model predicts negative? Always Specificity can't tolerate any false positives. Mathematically, the metric is the ratio of all True negatives to the sum of all negatives (True Negatives and False positives) in the dataset.

$$Specificity = \frac{Number\ of\ True\ (actual)\ Negative}{Total\ number\ of\ all\ Negative} = \frac{TN}{TN+FP}$$

- **False Positive Rate (FPR) or Miss Rate**

The false Positive Rate (FPR) is the evaluation metric that determines the proportion of negative data values (events) mistakenly categorized as positive (false positives). Mathematically, it is the ratio of the number of false positives to the total number of all negatives (True Negatives and False positives) in the dataset.

$$FPR = \frac{Number\ of\ False\ Positive}{Total\ number\ of\ all\ Negative} = \frac{FP}{TN+FP}$$

- **AUC-ROC**

AUC (Area under the Curve)-ROC (Receiver Operating Characteristics) curve is the visualization metric that evaluates the model performance in classification problems: by determining the classification values computed at various threshold settings. The metric is also known as AUROC (Area under the Receiver Operating Characteristics) curve.

The ROC curve is the probability curve that plots the TPR against FPR metrics—both with the range [0, 1]—to identify the better threshold which can separate the signal from noise. AUC is an area under two dimensional of the ROC curve that measures the degree of separability that distinguishes the performance of different classes or categorical algorithms. The greater the AUC, the better the model performance. Thus, the best model will have AUC near the value of 1, indicating a better classification performance, and the poor model will have a near zero AUC value. When the value is 0.5, the model cannot separate the classes or categorical methods.

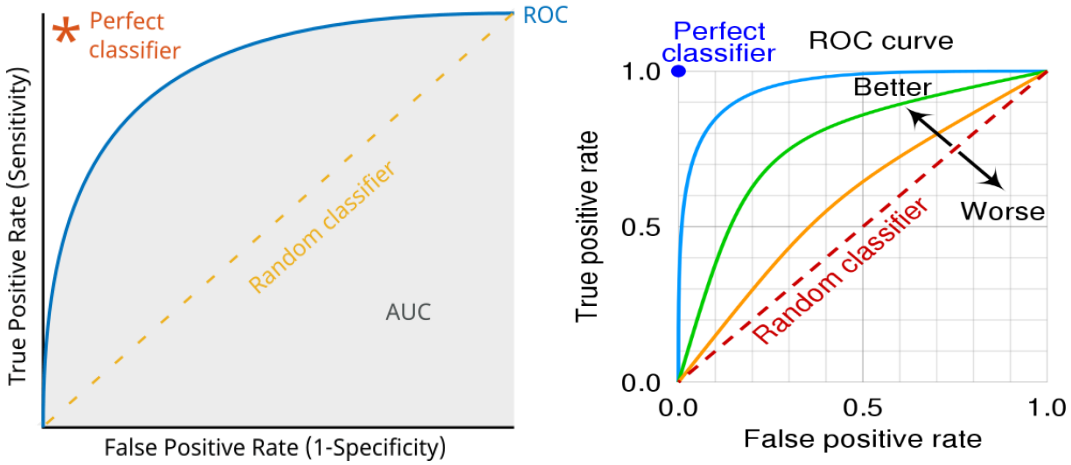


Figure 4-9: AUC-ROC Graphs

#### 4.3.3.2 Satisfying and Optimizing metric

Evaluation metrics are into two classes: optimizing metrics and satisfying metrics. An optimizing metric ensures the metric is as good as possible to maximize the model's accuracy. A satisficing metric ensures that the metric has met the expectation set. Accuracy is an example of optimizing metrics, and memory and running time are an example of satisfying metrics. In choosing evaluation metrics in training/development/test sets, it is crucial to consider which factor (satisfaction or optimization) can achieve the best model. The general rule for selecting the type of metric is:

$$N_{metric}: \begin{cases} 1 & Optimizing\ metric \\ N_{metric} - 1 & Satisfying\ metric \end{cases}$$

## 4.4. Model Designing, Experimentation, and Performance Evaluation

## 4.5. Experimental Results

# BIBLIOGRAPHY


[1] W.H.Organization, "Cardiovascular diseases (CVDs)." https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[2] G. A. Roth *et al.*, "Global, Regional, and National Burden of Cardiovascular Diseases for 10 Causes, 1990 to 2015," *J. Am. Coll. Cardiol.*, vol. 70, no. 1, pp. 1–25, Jul. 2017, doi: 10.1016/j.jacc.2017.04.052.

[3] S. Mendis, P. Puska, B. Norrving, W. H. Organization, W. H. Federation, and W. S. Organization, "Global atlas on cardiovascular disease prevention and control / edited by: Shanthi Mendis ... [et al.]." World Health Organization, p. Published by the World Health Organization in collaboration with the World Heart Federation and the World Stroke Organization, 2011. [Online]. Available: https://apps.who.int/iris/handle/10665/44701

[4] H. Ritchie and M. Roser, "Causes of death," *Our World Data*, Feb. 2018, Accessed: May 15, 2022. [Online]. Available: https://ourworldindata.org/causes-of-death

[5] R. Hajar, "Risk Factors for Coronary Artery Disease: Historical Perspectives," *Heart Views Off. J. Gulf Heart Assoc.*, vol. 18, no. 3, pp. 109–114, 2017, doi: 10.4103/HEARTVIEWS.HEARTVIEWS_106_17.

[6] R. Hajar, "Coronary Heart Disease: From Mummies to 21st Century," *Heart Views Off. J. Gulf Heart Assoc.*, vol. 18, no. 2, pp. 68–74, 2017, doi: 10.4103/HEARTVIEWS.HEARTVIEWS_57_17.

[7] R. Rehman, V. S. Yelamanchili, and A. N. Makaryus, "Cardiac Imaging," in *StatPearls*, Treasure Island (FL): StatPearls Publishing, 2022. Accessed: Sep. 26, 2022. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK448128/

[8] "Cardiovascular disease risk assessment for primary prevention: Risk calculators - UpToDate." https://www.uptodate.com/contents/cardiovascular-disease-risk-assessment-for-primary-prevention-risk-calculators#H2960596787 (accessed Sep. 28, 2022).

[9] A. Akella and S. Akella, "Machine learning algorithms for predicting coronary artery disease: efforts toward an open source solution," *Future Sci. OA*, vol. 7, no. 6, p. FSO698, Jul. 2021, doi: 10.2144/fsoa-2020-0206.

[10] C. B. C. Latha and S. C. Jeeva, "Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques," *Inform. Med. Unlocked*, vol. 16, p. 100203, Jan. 2019, doi: 10.1016/j.imu.2019.100203.

[11] J. W. Kennedy *et al.*, "Mortality related to cardiac catheterization and angiography," *Cathet. Cardiovasc. Diagn.*, vol. 8, no. 4, pp. 323–340, 1982, doi: 10.1002/ccd.1810080402.

[12] F. D. R. Hobbs, J. W. Jukema, P. M. Da Silva, T. McCormack, and A. L. Catapano, "Barriers to cardiovascular disease risk scoring and primary prevention in Europe," *QJM Int. J. Med.*, vol. 103, no. 10, pp. 727–739, Oct. 2010, doi: 10.1093/qjmed/hcq122.

[13] S. Uddin, A. Khan, M. E. Hossain, and M. A. Moni, "Comparing different supervised machine learning algorithms for disease prediction," *BMC Med. Inform. Decis. Mak.*, vol. 19, no. 1, p. 281, Dec. 2019, doi: 10.1186/s12911-019-1004-8.

[14] J. Flammer, K. Konieczka, R. M. Bruno, A. Virdis, A. J. Flammer, and S. Taddei, "The eye and the heart," *Eur. Heart J.*, vol. 34, no. 17, pp. 1270–1278, May 2013, doi: 10.1093/eurheartj/eht023.

[15] K. McGeechan *et al.*, "Retinal Vessel Caliber and Risk for Coronary Heart Disease: A Systematic Review and Meta-Analysis," *Ann. Intern. Med.*, vol. 151, no. 6, pp. 404–413, Sep. 2009.

[16] R. Allon, M. Aronov, M. Belkin, E. Maor, M. Shechter, and I. D. Fabian, "Retinal Microvascular Signs as Screening and Prognostic Factors for Cardiac Disease: A Systematic Review of Current Evidence," *Am. J. Med.*, vol. 134, no. 1, pp. 36-47.e7, Jan. 2021, doi: 10.1016/j.amjmed.2020.07.013.

[17] G. Litjens *et al.*, "State-of-the-Art Deep Learning in Cardiovascular Image Analysis," *JACC Cardiovasc. Imaging*, vol. 12, no. 8, Part 1, pp. 1549–1565, Aug. 2019, doi: 10.1016/j.jcmg.2019.06.009.

[18] H. Lu *et al.*, "Research Progress of Machine Learning and Deep Learning in Intelligent Diagnosis of the Coronary Atherosclerotic Heart Disease," *Comput. Math. Methods Med.*, vol. 2022, p. e3016532, Apr. 2022, doi: 10.1155/2022/3016532.
[19] K. C. Siontis, P. A. Noseworthy, Z. I. Attia, and P. A. Friedman, "Artificial intelligence-enhanced electrocardiography in cardiovascular disease management," *Nat. Rev. Cardiol.*, vol. 18, no. 7, Art. no. 7, Jul. 2021, doi: 10.1038/s41569-020-00503-2.
[20] "Anatomy of a Human Heart." https://healthblog.uofmhealth.org/heart-health/anatomy-of-a-human-heart (accessed Sep. 28, 2022).
[21] Cleveland Clinic, "Atherosclerosis: Causes, Symptoms, Risks & Tests," *Cleveland Clinic*. https://my.clevelandclinic.org/health/diseases/16753-atherosclerosis-arterial-disease (accessed Sep. 28, 2022).
[22] S. DeWeerdt, "Inflammation in heart disease: do researchers know enough?," *Nature*, vol. 594, no. 7862, pp. S8–S9, Jun. 2021, doi: 10.1038/d41586-021-01453-6.
[23] "The connection between hypertension, heart disease, and stroke," Feb. 04, 2022. https://www.medicalnewstoday.com/articles/how-are-hypertension-heart-disease-and-stroke-related (accessed Sep. 28, 2022).
[24] "All About the Symptoms of Coronary Artery Disease (CAD)," *Healthline*, Aug. 09, 2021. https://www.healthline.com/health/coronary-artery-disease/symptoms (accessed Sep. 28, 2022).
[25] S. Singh and R. Zeltser, "Cardiac Risk Stratification," in *StatPearls*, Treasure Island (FL): StatPearls Publishing, 2022. Accessed: Sep. 28, 2022. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK507785/
[26] "ACCF/SCAI/STS/AATS/AHA/ASNC 2009 Appropriateness Criteria for Coronary Revascularization." https://www.jacc.org/doi/epdf/10.1016/j.jacc.2008.10.005 (accessed Sep. 26, 2022).
[27] D. Chin, A. Battistoni, G. Tocci, J. Passerini, G. Parati, and M. Volpe, "Non-Invasive Diagnostic Testing for Coronary Artery Disease in the Hypertensive Patient: Potential Advantages of a Risk Estimation-Based Algorithm," *Am. J. Hypertens.*, vol. 25, no. 12, pp. 1226–1235, Dec. 2012, doi: 10.1038/ajh.2012.90.
[28] P. Ungprasert, E. L. Matteson, and C. S. Crowson, "Reliability of Cardiovascular Risk Calculators to Estimate Accurately the Risk of Cardiovascular Disease in Patients With Sarcoidosis," *Am. J. Cardiol.*, vol. 120, no. 5, pp. 868–873, Sep. 2017, doi: 10.1016/j.amjcard.2017.05.060.
[29] "Risk calculators for heart disease may be flawed," *Washington Post*. Accessed: Sep. 26, 2022. [Online]. Available: https://www.washingtonpost.com/national/health-science/risk-calculators-for-heart-disease-may-be-flawed/2018/08/31/f1845ee4-7a14-11e8-93cc-6d3beccdd7a3_story.html
[30] V. Kaul, S. Enslin, and S. A. Gross, "History of artificial intelligence in medicine," *Gastrointest. Endosc.*, vol. 92, no. 4, pp. 807–812, Oct. 2020, doi: 10.1016/j.gie.2020.06.040.
[31] A. Anaya-Isaza, L. Mera-Jiménez, and M. Zequera-Diaz, "An overview of deep learning in medical imaging," *Inform. Med. Unlocked*, vol. 26, p. 100723, Jan. 2021, doi: 10.1016/j.imu.2021.100723.
[32] Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, and M. S. Nasrin, "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches," p. 39.
[33] "Medical Definition of Retinal fundus," *MedicineNet*. https://www.medicinenet.com/retinal_fundus/definition.htm (accessed May 17, 2022).
[34] U. Iqbal, "Smartphone fundus photography: a narrative review," *Int. J. Retina Vitr.*, vol. 7, no. 1, p. 44, Jun. 2021, doi: 10.1186/s40942-021-00313-9.
[35] "Introduction: - Human Eye." https://sites.google.com/site/humaneye12/introduction (accessed Sep. 29, 2022).
[36] "Structure and Functions of Human Eye with labelled Diagram," *BYJUS*. https://byjus.com/biology/structure-of-eye/ (accessed Sep. 29, 2022).

[37] “Human eye,” *Wikipedia*. Sep. 12, 2022. Accessed: Sep. 29, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Human_eye&oldid=1109850979
[38] Cleveland Clinic, “Eyes: How They Work, Anatomy & Common Conditions,” *Cleveland Clinic*. https://my.clevelandclinic.org/health/body/21823-eyes (accessed Sep. 29, 2022).
[39] “How do eyes work? - Human Eye.” https://sites.google.com/site/humaneye12/resume (accessed May 18, 2022).
[40] “Eye vs. Camera,” *Let’s Talk Science*. https://letstalkscience.ca/educational-resources/stem-in-context/eye-vs-camera (accessed May 18, 2022).
[41] eOphtha, “Anatomy of Retina.” https://www.eophtha.com/posts/anatomy-of-retina (accessed May 20, 2022).
[42] N. Mahabadi and Y. Al Khalili, “Neuroanatomy, Retina,” in *StatPearls*, Treasure Island (FL): StatPearls Publishing, 2022. Accessed: Sep. 29, 2022. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK545310/
[43] “Deep Learning (Ian Goodfellow, Yoshua Bengio, Aaron Courville) (z-lib.org).pdf.”
[44] L. Deng and D. Yu, “Deep Learning: Methods and Applications,” *Found. Trends® Signal Process.*, vol. 7, no. 3–4, pp. 197–387, Jun. 2014, doi: 10.1561/2000000039.
[45] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, Art. no. 7553, May 2015, doi: 10.1038/nature14539.
[46] K. D. Foote, “A Brief History of Deep Learning,” *DATAVERSITY*, Feb. 04, 2022. https://www.dataversity.net/brief-history-deep-learning/ (accessed Sep. 30, 2022).
[47] “An Easy Guide to Neuron Diagrams and Types,” *Healthline*, Feb. 28, 2022. https://www.healthline.com/health/neurons (accessed Jun. 27, 2022).
[48] N. T. Carnevale and M. L. Hines, *The NEURON Book*. Cambridge University Press, 2006.
[49] J.-W. Lin, “Artificial neural network related to biological neuron network: a review,” *Adv. Stud. Med. Sci.*, vol. 5, pp. 55–62, 2017, doi: 10.12988/asms.2017.753.
[50] “Neural Smithing Supervised Learning in Feedforward Artificial Neural Networks (Russell Reed, Robert J MarksII) (z-lib.org).pdf.”
[51] D. Stathakis, “How many hidden layers and nodes?,” *Int. J. Remote Sens.*, vol. 30, no. 8, pp. 2133–2147, Apr. 2009, doi: 10.1080/01431160802549278.
[52] “Neural Network Systems Techniques and Applications. Volume 2. Optimization Techniques ( PDFDrive ).pdf.”
[53] S. Karagiannakos, “Regularization techniques for training deep neural networks,” *AI Summer*, May 27, 2021. https://theaisummer.com/regularization/ (accessed Jan. 15, 2023).
[54] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014.
[55] L. Prechelt, “Early Stopping - But When?,” in *Neural Networks: Tricks of the Trade*, G. B. Orr and K.-R. Müller, Eds. Berlin, Heidelberg: Springer, 1998, pp. 55–69. doi: 10.1007/3-540-49430-8_3.
[56] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning requires rethinking generalization.” arXiv, Feb. 26, 2017. doi: 10.48550/arXiv.1611.03530.
[57] J. Sum, C.-S. Leung, and K. Ho, “A Limitation of Gradient Descent Learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2227–2232, Jun. 2020, doi: 10.1109/TNNLS.2019.2927689.
[58] A. Defazio, “Momentum via Primal Averaging: Theoretical Insights and Learning Rate Schedules for Non-Convex Optimization.” arXiv, Jun. 01, 2021. Accessed: Nov. 20, 2022. [Online]. Available: http://arxiv.org/abs/2010.00406
[59] X. Liu, W. Tao, and Z. Pan, “A convergence analysis of Nesterov’s accelerated gradient method in training deep linear neural networks,” *Inf. Sci.*, vol. 612, pp. 898–925, Oct. 2022, doi: 10.1016/j.ins.2022.08.090.

[60] J. Duchi, E. Hazan, and Y. Singer, “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,” *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Jul. 2011.
[61] M. D. Zeiler, “ADADELTA: An Adaptive Learning Rate Method.” arXiv, Dec. 22, 2012. Accessed: Nov. 20, 2022. [Online]. Available: http://arxiv.org/abs/1212.5701
[62] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization.” arXiv, Jan. 29, 2017. doi: 10.48550/arXiv.1412.6980.
[63] K. You, M. Long, J. Wang, and M. I. Jordan, “How Does Learning Rate Decay Help Modern Neural Networks?” arXiv, Sep. 26, 2019. doi: 10.48550/arXiv.1908.01878.
[64] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.” arXiv, Mar. 02, 2015. doi: 10.48550/arXiv.1502.03167.
[65] D. H. Hubel and T. N. Wiesel, “Receptive fields of single neurones in the cat’s striate cortex,” *J. Physiol.*, vol. 148, no. 3, pp. 574–591, Oct. 1959.
[66] D. Hutchison *et al.*, “Blocks World Revisited: Image Understanding Using Qualitative Geometry and Mechanics,” in *Computer Vision – ECCV 2010*, vol. 6314, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 482–496. doi: 10.1007/978-3-642-15561-1_35.
[67] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, Apr. 1980, doi: 10.1007/BF00344251.
[68] Y. LeCun, K. Kavukcuoglu, and C. Farabet, “Convolutional networks and applications in vision,” in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, Paris, France, May 2010, pp. 253–256. doi: 10.1109/ISCAS.2010.5537907.
[69] Y. LeCun, “Gradient-Based Learning Applied to Document Recognition,” *Proc. IEEE*, Jan. 1998, Accessed: Oct. 13, 2022. [Online]. Available: https://www.academia.edu/8932919/Gradient_Based_Learning_Applied_to_Document_Recognition
[70] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Neural Inf. Process. Syst.*, vol. 25, Jan. 2012, doi: 10.1145/3065386.
[71] L. Alzubaidi *et al.*, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *J. Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.
[72] G. Yao, T. Lei, and J. Zhong, “A review of Convolutional-Neural-Network-based action recognition,” *Pattern Recognit. Lett.*, vol. 118, pp. 14–22, Feb. 2019, doi: 10.1016/j.patrec.2018.05.018.
[73] A. Dhillon and G. K. Verma, “Convolutional neural network: a review of models, methodologies and applications to object detection,” *Prog. Artif. Intell.*, vol. 9, no. 2, pp. 85–112, Jun. 2020, doi: 10.1007/s13748-019-00203-0.
[74] S. R. Kheradpisheh, M. Ghodrati, M. Ganjtabesh, and T. Masquelier, “Deep Networks Can Resemble Human Feed-forward Vision in Invariant Object Recognition,” *Sci. Rep.*, vol. 6, no. 1, Art. no. 1, Sep. 2016, doi: 10.1038/srep32672.
[75] A. De Cesarei, S. Cavicchi, G. Cristadoro, and M. Lippi, “Do Humans and Deep Convolutional Neural Networks Use Visual Information Similarly for the Categorization of Natural Scenes?,” *Cogn. Sci.*, vol. 45, no. 6, p. e13009, Jun. 2021, doi: 10.1111/cogs.13009.
[76] D. R. Sarvamangala and R. V. Kulkarni, “Convolutional neural networks in medical image understanding: a survey,” *Evol. Intell.*, vol. 15, no. 1, pp. 1–22, Mar. 2022, doi: 10.1007/s12065-020-00540-3.
[77] S. S. Yadav and S. M. Jadhav, “Deep convolutional neural network based medical image classification for disease diagnosis,” *J. Big Data*, vol. 6, no. 1, p. 113, Dec. 2019, doi: 10.1186/s40537-019-0276-2.

[78] H. El-Rewaidy *et al.*, "Multi-domain convolutional neural network (MD-CNN) for radial reconstruction of dynamic cardiac MRI," *Magn. Reson. Med.*, vol. 85, no. 3, pp. 1195–1208, Mar. 2021, doi: 10.1002/mrm.28485.
[79] S. Sakib, N. Ahmed, A. J. Kabir, and H. Ahmed, "An Overview of Convolutional Neural Network: Its Architecture and Applications." Preprints, Feb. 14, 2019. doi: 10.20944/preprints201811.0546.v4.
[80] X. Hou *et al.*, "Learning Based Image Transformation Using Convolutional Neural Networks," *IEEE Access*, vol. 6, pp. 49779–49792, 2018, doi: 10.1109/ACCESS.2018.2868733.
[81] O. Semih Kayhan and J. C. van Gemert, "On Translation Invariance in CNNs: Convolutional Layers Can Exploit Absolute Spatial Location," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 14262–14273. doi: 10.1109/CVPR42600.2020.01428.
[82] A. Bietti and J. Mairal, "Invariance and Stability of Deep Convolutional Representations," in *Advances in Neural Information Processing Systems*, 2017, vol. 30. Accessed: Oct. 17, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/38ed162a0dbef7b3fe0f628aa08b90e7-Abstract.html
[83] R. Blything, V. Biscione, I. I. Vankov, C. J. H. Ludwig, and J. S. Bowers, "The human visual system and CNNs can both support robust online translation tolerance following extreme displacements," *J. Vis.*, vol. 21, no. 2, p. 9, Feb. 2021, doi: 10.1167/jov.21.2.9.
[84] "Output Feature Map - an overview | ScienceDirect Topics." https://www.sciencedirect.com/topics/computer-science/output-feature-map (accessed Oct. 15, 2022).
[85] E. W. Weisstein, "Convolution." https://mathworld.wolfram.com/ (accessed Oct. 15, 2022).
[86] S. Kim and R. Casper, "Applications of convolution in image processing with MATLAB," *Univ. Wash.*, pp. 1–20, 2013.
[87] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Penksy, "Sparse Convolutional Neural Networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 806–814. doi: 10.1109/CVPR.2015.7298681.
[88] P. Savarese and M. Maire, "Learning Implicitly Recurrent CNNs Through Parameter Sharing." arXiv, Mar. 13, 2019. doi: 10.48550/arXiv.1902.09701.
[89] T. S. Cohen and M. Welling, "Group Equivariant Convolutional Networks." arXiv, Jun. 03, 2016. Accessed: Oct. 17, 2022. [Online]. Available: http://arxiv.org/abs/1602.07576
[90] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, Dec. 2020, doi: 10.1007/s10462-020-09825-6.
[91] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition." arXiv, Apr. 10, 2015. doi: 10.48550/arXiv.1409.1556.
[92] C. Szegedy *et al.*, "Going Deeper with Convolutions." arXiv, Sep. 16, 2014. doi: 10.48550/arXiv.1409.4842.
[93] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition." arXiv, Dec. 10, 2015. doi: 10.48550/arXiv.1512.03385.
[94] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." arXiv, Apr. 16, 2017. doi: 10.48550/arXiv.1704.04861.
[95] F. Chollet, "Xception: Deep Learning With Depthwise Separable Convolutions," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251–1258. Accessed: Oct. 24, 2022. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Chollet_Xception_Deep_Learning_CVPR_2017_paper.html

[96] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks." arXiv, Mar. 21, 2019. doi: 10.48550/arXiv.1801.04381.
[97] H. Hosseini, B. Xiao, M. Jaiswal, and R. Poovendran, "On the Limitation of Convolutional Neural Networks in Recognizing Negative Images," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2017, pp. 352–358. doi: 10.1109/ICMLA.2017.0-136.
[98] F. H. Malik, F. Batool, A. Rubab, N. A. Chaudhary, K. B. Khan, and M. A. Qureshi, "Retinal disorder as a biomarker for detection of human diseases," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, Nov. 2020, pp. 1–6. doi: 10.1109/INMIC50486.2020.9318059.
[99] S. Ktori, "Using AI to Assess Eye Scans, Scientists Find 'Retinal Age Gap' Predicts Mortality Risk," *GEN - Genetic Engineering and Biotechnology News*, Jan. 19, 2022. https://www.genengnews.com/artificial-intelligence/using-ai-to-assess-eye-scans-scientists-find-retinal-age-gap-predicts-mortality-risk/ (accessed May 21, 2022).
[100] E. Korot *et al.*, "Predicting sex from retinal fundus photographs using automated deep learning," *Sci. Rep.*, vol. 11, no. 1, Art. no. 1, May 2021, doi: 10.1038/s41598-021-89743-x.
[101] M. Bhaskaranand *et al.*, "Automated Diabetic Retinopathy Screening and Monitoring Using Retinal Fundus Image Analysis," *J. Diabetes Sci. Technol.*, vol. 10, no. 2, pp. 254–261, Mar. 2016, doi: 10.1177/1932296816628546.
[102] R. Raman, S. Srinivasan, S. Virmani, S. Sivaprasad, C. Rao, and R. Rajalakshmi, "Fundus photograph-based deep learning algorithms in detecting diabetic retinopathy," *Eye*, vol. 33, no. 1, Art. no. 1, Jan. 2019, doi: 10.1038/s41433-018-0269-y.
[103] A. V. Stanton, P. Mullaney, F. Mee, E. T. O'Brien, and K. O'Malley, "A method of quantifying retinal microvascular alterations associated with blood pressure and age," *J. Hypertens.*, vol. 13, no. 1, pp. 41–48, Jan. 1995.
[104] M. Bhargava, M. K. Ikram, and T. Y. Wong, "How does hypertension affect your eyes?," *J. Hum. Hypertens.*, vol. 26, no. 2, pp. 71–83, Feb. 2012, doi: 10.1038/jhh.2011.37.
[105] I. A. Pikuleva and C. A. Curcio, "Cholesterol in the retina: the best is yet to come," *Prog. Retin. Eye Res.*, vol. 41, pp. 64–89, Jul. 2014, doi: 10.1016/j.preteyeres.2014.03.002.
[106] N. Cheung and T. Y. Wong, "Obesity and Eye Diseases," *Surv. Ophthalmol.*, vol. 52, no. 2, pp. 180–195, 2007, doi: 10.1016/j.survophthal.2006.12.003.
[107] A. Agarwal *et al.*, "Effect of weight loss on the retinochoroidal structural alterations among patients with exogenous obesity," *PLOS ONE*, vol. 15, no. 7, p. e0235926, Jul. 2020, doi: 10.1371/journal.pone.0235926.
[108] E. Vaghefi, S. Yang, S. Hill, G. Humphrey, N. Walker, and D. Squirrell, "Detection of smoking status from retinal images; a Convolutional Neural Network study," *Sci. Rep.*, vol. 9, no. 1, Art. no. 1, May 2019, doi: 10.1038/s41598-019-43670-0.
[109] T.-K. Yang, X.-G. Huang, and J.-Y. Yao, "Effects of Cigarette Smoking on Retinal and Choroidal Thickness: A Systematic Review and Meta-Analysis," *J. Ophthalmol.*, vol. 2019, p. e8079127, Sep. 2019, doi: 10.1155/2019/8079127.
[110] K. Teberik, "The Effect of Smoking on Macular, Choroidal, and Retina Nerve Fiber Layer Thickness," *Turk. J. Ophthalmol.*, vol. 49, no. 1, pp. 20–24, Feb. 2019, doi: 10.4274/tjo.galenos.2018.80588.
[111] "TensorFlow." https://www.tensorflow.org/ (accessed Jan. 01, 2023).
[112] J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken, "Ridge-based vessel segmentation in color images of the retina," *IEEE Trans. Med. Imaging*, vol. 23, no. 4, pp. 501–509, Apr. 2004, doi: 10.1109/TMI.2004.825627.
[113] A. D. Hoover, V. Kouznetsova, and M. Goldbaum, "Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response," *IEEE Trans. Med. Imaging*, vol. 19, no. 3, pp. 203–210, Mar. 2000, doi: 10.1109/42.845178.

[114] P. Porwal *et al.*, "Indian Diabetic Retinopathy Image Dataset (IDRiD): A Database for Diabetic Retinopathy Screening Research," *Data*, vol. 3, no. 3, Art. no. 3, Sep. 2018, doi: 10.3390/data3030025.
[115] P. Porwal *et al.*, "IDRiD: Diabetic Retinopathy – Segmentation and Grading Challenge," *Med. Image Anal.*, vol. 59, p. 101561, Jan. 2020, doi: 10.1016/j.media.2019.101561.
[116] M. M. Fraz, A. R. Rudnicka, C. G. Owen, and S. A. Barman, "Delineation of blood vessels in pediatric retinal images using decision trees-based ensemble classification," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 9, no. 5, pp. 795–811, Sep. 2014, doi: 10.1007/s11548-013-0965-9.
[117] J. Cuadros and G. Bresnick, "EyePACS: An Adaptable Telemedicine System for Diabetic Retinopathy Screening," *J. Diabetes Sci. Technol. Online*, vol. 3, no. 3, pp. 509–516, May 2009.
[118] E. Decencière *et al.*, "FEEDBACK ON A PUBLICLY DISTRIBUTED IMAGE DATABASE: THE MESSIDOR DATABASE," *Image Anal. Stereol.*, vol. 33, no. 3, Art. no. 3, Aug. 2014, doi: 10.5566/ias.1155.
[119] Y. Xu and R. Goodacre, "On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning," *J. Anal. Test.*, vol. 2, no. 3, pp. 249–262, Jul. 2018, doi: 10.1007/s41664-018-0068-2.
[120] O. A. Montesinos López, A. Montesinos López, and J. Crossa, "Overfitting, Model Tuning, and Evaluation of Prediction Performance," in *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, O. A. Montesinos López, A. Montesinos López, and J. Crossa, Eds. Cham: Springer International Publishing, 2022, pp. 109–139. doi: 10.1007/978-3-030-89010-0_4.
[121] T. Viering and M. Loog, "The Shape of Learning Curves: A Review," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–20, 2022, doi: 10.1109/TPAMI.2022.3220744.
[122] X. Guan and H. Burton, "Bias-variance tradeoff in machine learning: Theoretical formulation and implications to structural engineering applications," *Structures*, vol. 46, pp. 17–30, Dec. 2022, doi: 10.1016/j.istruc.2022.10.004.
[123] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias–variance trade-off," *Proc. Natl. Acad. Sci.*, vol. 116, no. 32, pp. 15849–15854, Aug. 2019, doi: 10.1073/pnas.1903070116.
[124] M. R. Reddy, B. N. Kumar, N. M. Rao, and B. Karthikeyan, "A New Approach for Bias–Variance Analysis Using Regularized Linear Regression," in *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Singapore, 2020, pp. 35–46. doi: 10.1007/978-981-15-0339-9_4.
[125] S. Masiha, A. Gohari, M. H. Yassaee, and M. R. Aref, "Learning under Distribution Mismatch and Model Misspecification." arXiv, Aug. 10, 2022. Accessed: Jan. 05, 2023. [Online]. Available: http://arxiv.org/abs/2102.05695
[126] H. K. Jabbar and R. Z. Khan, "Methods to Avoid Over-Fitting and Under-Fitting in Supervised Machine Learning (Comparative Study)," in *Computer Science, Communication and Instrumentation Devices*, 2014, pp. 163–172. doi: 10.3850/978-981-09-5247-1_017.
[127] D. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," *Mach Learn Technol*, vol. 2, Jan. 2008.