POPULAR DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH



University of Yahia Fares - Médéa
Faculty of Sciences
Department of Mathematics and Computer Science



Graduation Project

Submitted for the Master Degree in Computer Science

Option: System Engineering and Web Technology

Deep Learning and Transfer Learning-based Detection of COVID-19 using Radiography Images

---

Proposed and directed by :

Dr. Ahmed Cherif Mazari

Presented by :

Miss. Nouha KADIK
Miss. Maroua BENDOU

2021/2022


بسم الله الرحمن الرحيم

Acknowledgements

In the name of **ALLAH**, the Most Gracious and the Most Merciful.

All praises to **ALLAH** and His blessing for the completion of this work. who we thank for all the opportunities,trials and strength that have been showered on us to complete this work.

Our humblest gratitude to the holy Prophet **Muhammad** (Peace be upon him) whose way of life has been a continuous guidance for us.

We would like to express our heartfelt gratitude to our supervisor, **Dr. Ahmed Cherif Mazari**, for his constant support, general insight, and direction during the course of this project, and acknowledge that it was a real privilege and honor to work and study under his supervision.

We want to additionally express our heartfelt gratitude to all of our dear classmates and friend group who stood by us and supported us through thick and thin.

We thank the honourable jury who agreed to judge our work for their time and commitment .
Lastly, we thank whoever crossed our path and helped us in our journey to complete this work.

This thesis is dedicated to:

My mother **Yamina**, who gave me the encouragement and made my life much easier.

My sister **Sihem** and her kids **Wassim** and **Tassnim** who I love a lot.

My two brothers **Seif el Islem** and **Abd El Basset** who supported me morally.

My partner and best friend **Nouha** who shared the good and bad with me in my journey.

To all my friends **Yasmine Benlakehal, Fethia Belkacem, Houssem Mihoubi** my family and everyone who knows me and helped me in this thesis.

Maroua Bendou

This thesis is dedicated to:

My mother **Nabila** and my father **Mustapha** ,Words can not express how grateful I am for all of the sacrifices that you've made on my behalf. Your prayers for me were what sustained me thus far.

My Brother **Madjid** for his support throughout the years.

My Brother **Ahmed** for being the kind soul that he is and his moral and financial support.

My soul sister **Nabila**, my amethyst **Mouna**, my joy **Farah**, you have been here since the beginning and there is no one i would rather share this with other than you .

My partner and best friend **Maroua**, for being such an amazing partner and best friend.

Nouha KADIK

Abstract

Covid-19 has been fought on several fronts and in various forms. Many aspects of coping with the pandemic may be dealt with using computerized methods; Physical sampling, vital sign monitoring, and disinfection are just a few examples of robotic solutions. Furthermore, image recognition and artificial intelligence are being utilized to confirm cases. In this work, we present a deep learning and transfer learning method for automatically diagnosing COVID-19-related pneumonia from X-ray pictures.
We used these approaches in our work, Convolutional neural network (CNN), MobileNetV2, and Residual Network(ResNet50) are three deep learning architectures we presented . We also look at which one is the most accurate.
This research focuses on the Covid-19 viral identification work, which is concerned with finding the virus in chest X-ray pictures.
The CNN model(96.58%) outperformed the MobileNetV2 (94%) and ResNet50 (92.83%) models in every performance metric.

Key words

Deep learning, Machine learning, CNN, ResNet50, MobileNetV2,Covid-19.

Résumé

Le Covid-19 a été combattu sur plusieurs fronts et sous diverses formes. De nombreux aspects de la lutte contre la pandémie peuvent être traités à l'aide de méthodes informatisées; l'échantillonnage physique, la surveillance des signes vitaux et la désinfection ne sont que quelques exemples de solutions robotiques.
De plus, la reconnaissance d'images et l'intelligence artificielle sont utilisées pour confirmer les cas. Dans ce travail, nous présentons une méthode d'apprentissage en profondeur et d'apprentissage par transfert pour diagnostiquer automatiquement la pneumonie liée au COVID-19 à partir d'images radiographiques.
Le modèle CNN (96.58%) a surpassé les modèles MobileNetV2 (94%) et ResNet50 (92.83%) dans toutes les mesures de performance.

Mot clés

Apprentissage en profondeur, Apprentissage automatique, CNN, ResNet50, MobileNetV2, COVID-19.

الملخص

تم مكافحة كوفيد ١٩ على جبهات عديدة وبأشكال مختلفة. يمكن التعامل مع العديد من جوانب التعامل مع الوباء باستخدام الأساليب المحوسبة. أخذ العينات المادية ومراقبة العلامات الحيوية والتطهير ليست سوى أمثلة قليلة للحلول الروبوتية. علاوة على ذلك ، يتم استخدام التعرف على الصور والذكاء الاصطناعي لتأكيد الحالات. في هذا العمل ، نقدم طريقة التعلم العميق ونقل التعلم لتشخيص الالتهاب الرئوي المرتبط بـ كوفيد ١٩ تلقائيًا من صور الأشعة السينية. يركز هذا البحث على وظيفة التعرف على فيروس Covid-19 ، نستخدم هذه الأساليب في عملنا ، الشبكة العصبية التلافيفية MobileNetV2 والشبكة المتبقية ResNet50 هي ثلاث بنيات وننظر أيضًا في أي منها هو الأكثر دقة. تفوق نموذج CNN(96.58%) على طرازي MobileNetV2(94%) والشبكة المتبقية ResNet50(92.83%) في كل مقياس أداء.

الكلمات المفتاحية

Covid-19 ، ResNet50 ، CNN ، MobileNetV2 ، تعلم عميق ، نقل التعلم

Table of content

Abstract . . . ii
List of Figures . . . ix
List of Tables . . . x
Abbreviations . . . xi
General Introduction . . . 1

1 Digital and Covid-19 image **3**
1.1 Introduction . . . 4
1.2 Images . . . 4
1.2.1 Basic image definition . . . 4
1.3 Digital image definition . . . 5
1.3.1 Types of digital images . . . 6
1.4 Pixels . . . 7
1.4.1 The binary image . . . 7
1.4.2 Gray-scale image . . . 8
1.4.3 Color image . . . 9
1.5 Image file format . . . 9
1.6 Image resolution . . . 11
1.6.1 Spatial Resolution . . . 11
1.6.2 Gray level resolution . . . 12
1.7 Image segmentation . . . 12
1.7.1 Definition . . . 12
1.7.2 Groups of image segmentation . . . 13
1.7.3 Region approach . . . 13
1.7.4 Edge detection approach . . . 13
1.7.5 Filter . . . 13
1.7.6 Image smoothing . . . 13
1.7.7 Image noise . . . 15
1.7.8 Convolution . . . 15
1.7.9 Contour . . . 16
1.8 Medical imaging . . . 16
1.8.1 Medical imaging technologies . . . 16
1.8.2 Medical image segmentation . . . 18
1.9 Computer vision . . . 19
1.10 Covid-19 . . . 19
1.10.1 Covid-19 symptoms . . . 19
1.10.2 Covid-19 Timeline : . . . 19
1.10.3 Covid-19 Images . . . 20
1.10.4 Covid-19 detection . . . 22
1.11 Conclusion . . . 24

2 Deep learning and transfer learning-based models 25
2.1 Introduction . . . 26
2.2 Machine learning . . . 26
2.2.1 Supervised Learning . . . 26
2.2.2 Unsupervised Learning . . . 26
2.3 Deep learning . . . 27
2.3.1 Definition . . . 27
2.3.2 Deep Learning Applications . . . 27
2.3.3 Deep learning techniques . . . 28
2.4 Difference between machine learning and deep learning . . . 33
2.5 Network Architectures . . . 34
2.5.1 Convolutional Neural Network . . . 34
2.5.2 VGG Network . . . 34
2.5.3 ResNet . . . 35
2.5.4 AlexNet . . . 37
2.5.5 GoogleNet . . . 37
2.5.6 MobileNet . . . 38
2.6 Transfer learning basics . . . 39
2.6.1 Transfer learning . . . 39
2.6.2 Definition . . . 39
2.6.3 Question to answer before transferring . . . 39
2.7 Transfer learning Approaches . . . 40
2.7.1 Develop model approach . . . 40
2.7.2 Pre-trained model approach . . . 40
2.8 The difference between machine learning and transfer learning . . . 41
2.9 Data processing . . . 41
2.9.1 Definition . . . 41
2.9.2 Data normalization . . . 42
2.9.3 Data augmentation . . . 42
2.10 Feature extraction . . . 43
2.11 Model evaluation . . . 43
2.11.1 Metrics . . . 43
2.12 Previous studies . . . 45

3 Proposed Approach 51
3.1 Introduction . . . 52
3.2 Methodology . . . 52
3.3 System architecture . . . 52
3.4 Proposed approaches . . . 53
3.4.1 Convolutional Neural Network CNN . . . 53
3.4.2 Resnet50 . . . 55
3.4.3 MobileNetV2 . . . 57
3.5 Conclusion . . . 57

4 Test and evaluation 58
4.1 Introduction . . . 59
4.2 Environment and development tools . . . 59
4.2.1 Google Co-laboratory . . . 59
4.2.2 Python Modules Used . . . 60

4.3 Data used for training and test 60
4.3.1 Kaggel COVID-QU-Ex Dataset 60
4.4 Experimentation Steps 61
4.4.1 Data Preprocessing 61
4.4.2 Splitting the data 61
4.4.3 Train, test and validation 62
4.4.4 Over-fitting and under-fitting 62
4.5 Results and discussion 62
4.5.1 CNN implementation 63
4.5.2 ResNet50 implementation 65
4.5.3 MobileNetV2 implementation 67
4.6 Discussion 68
4.6.1 Performance comparison between CNN and transfer learning architectures 69
4.6.2 Performance comparison between the transfer learning architectures 70
4.6.3 Performance comparison with the previous studies 70
4.7 Conclusion 71
General Conclusion 72
Bibliography 75

List of Figures

1.1 Basic Image. . . . 4
1.2 Representation of image sampling using graphs[1]. . . . 5
1.3 An example of a quantization process[1]. . . . 6
1.4 Vector Graphics.[2] 6
1.5 Raster Graphics [2]. . . . 7
1.6 Binary Image. . . . 8
1.7 Gray-scale image. . . . 8
1.8 Color image. . . . 9
1.9 A comparison of screen resolutions for 72 PPI, 150 PPI, 300 PPI [4]. . . . 11
1.10 A comparison of screen resolutions for 300 dpi, 72 dpi, 10 dpi[4] 12
1.11 Gray-level resolution. . . . 12
1.12 Mean filter application. . . . 14
1.13 Median filter. . . . 15
1.14 Different contour types. . . . 16
1.15 Typical IVUS image. . . . 18
1.16 Covid-19 timeline. . . . 20
1.17 An X-ray of lungs of patient with Covid-19. . . . 21
1.18 Ultra sound of the lungs of a patient with COVID-19. . . . 21
1.19 Baseline CT images (a-d) of a 50 year-old man admitted for symptoms of fever for 1 day.There were multiple patchy ground-glass opacities in the left upper lobe and right lower lobe (a, b). Air bronchogram and crazy paving pattern can be seen (c). Lesions are located in the peripheral area of the lung (d). . . . 22
1.20 PCR and molecular biology fundamental principals. . . . 23
1.21 RDT. . . . 24

2.1 The structure of a deep learning network 27
2.2 DNN architecture. . . . 28
2.3 simple CNN architecture of five layers 29
2.4 Convonlutional layer [12]. . . . 29
2.5 Pooling layer. . . . 30
2.6 Fully connected layer. . . . 31
2.7 a simple RNN architecture. . . . 32
2.8 A LSTM architecture 32
2.9 An Auto Encoder architecture. . . . 33
2.10 The difference between AI,machine learning and deep learning. . . . 34
2.11 VGG Layers. . . . 35

2.12 Example network architectures. Left: the VGG-19 model as a reference. Middle: a plain network with 34 parameter layers . Right: a residual network with 34 parameter layers . The dotted shortcuts increase dimensions[14]. 36
2.13 AlexNet architecture diagram [17]. 37
2.14 GoogleNet architecture diagram.[18] 38
2.15 Structure of inceptions[19] 38
2.16 MobileNet architecture diagram.[22] 39
2.17 Data pre-processing cycle. 42
2.18 Data normalization. 42
2.19 Example of Data Augmentation. 43
2.20 Shallow CNN models and their achievements [28]. 49

3.1 the general structure of the process. 53
3.2 CNN Model Summary. 54
3.3 ResNet50 Architecture. 56
3.4 Added layers to ResNet50. 57

4.1 Samples from dataset. 61
4.2 the compile and fit functions for the The three models 63
4.3 the last epochs of the training (CNN). 63
4.4 Training and validation: loss and accuracy. 64
4.5 Testing labels. 64
4.6 The last epochs of the training (ResNet50). 65
4.7 Training and validation: loss and accuracy (ResNet50). 66
4.8 the training and validation: loss and accuracy (MobileNetV2). 67
4.9 Training and validation: loss and accuracy (MobileNetV2). 67
4.10 Preformance comparaison between ResNet50 and MobileNetV2. 70
4.11 Accuracy comparison between previous work and our work. 71

List of Tables

1.1 Common Image file Formats. 10

2.1 The differences between Machine Learning and Transfer Learning. 41
2.2 Training results for XCeption Net. 45
2.3 Testing results for XCeption Net[23]. 45
2.4 Training results for Inception Net V3. 45
2.5 Testing results for Inception Net V3. 46
2.6 Training results for ResNeXt. 46
2.7 Testing results for ResNeXt. 46
2.8 Details of data splitting with and without augmentation. 46
2.9 Classification performance (in %) comparison among different optimizers. . 47
2.10 Classification results comparison of all eight CNN models (in %). 47
2.11 Achievements of pretrained deep CNN models and SVM classifiers on COVID-19 detection (Accuracy %). 48
2.12 Achievements of fine-tuning of pretrained deep CNN models on COVID-19 classification. 49
2.13 Results of first experiment using local feature descriptors and SVM classifiers (accuracy %). 50

4.1 organization of training, testing and validation in the Resnet, CNN,MobileNetV2 proposed models. 62
4.2 Accuracy results 64
4.3 CNN Confusion matrix 65
4.4 CNN Test results. 65
4.5 Accuracy results (Resnet50). 66
4.6 ResNet50 Confusion matrix 66
4.7 ResNet50 Test results. 67
4.8 Accuracy results (MobileNetV2). 68
4.9 MobileNetV2 Confusion matrix 68
4.10 MobileNetV2 Test results. 68
4.11 Performance comparison between the three model. 69
4.12 Performance comparison with the previous studies. 70

Abbreviations

Ag-RDT Antigen Rapid Diagnostic.

AI Artificial Intelligence.

ANNs Artificial neural networks.

CNN Convolutional neural network.

Covid-19 Coronavirus disease 2019.

CPU Central processing unit.

CXR Chest X-Ray.

DL Deep Learning.

DNN Deep neural network.

FC Fully connected.

IVUS Intravascular Ultrasound.

MRI Magnetic resonance images.

PCR Polymerase Chain Reaction.

PPI Pixels Per Inch.

ResNet Residual Network.

RGB Red Green Blue.

RNN Recurrent neural network.

SNNs Simulated neural networks.

TL Transfer Learning.

VGG Visual Geometry Group.

WHO World Health Organization.

General Introduction

In the past three years, the daily life of human beings, their health and the economy of all countries has been affected because of the deadly virus Covid-19 that claimed the lives of a large number of people all over the world.

Coronavirus disease 2019 (Covid-19) is an infectious disease caused by the SARS-CoV-2 virus, according to a report provided by the World Health Organization (WHO). There were 532 million infected cases from 2019 till now, with 6.3 million deaths .
Typically, such disease is detected using human-based detection, which entails obtaining a nasal pharyngeal swab or a pharyngeal swab to see if it is infected or not, using Antigen Rapid Diagnostic (Ag-RDT) tests or Polymerase Chain Reaction (PCR) tests. However, this traditional process is time and human resource consuming.
That's why, Covid-19 detection should be aimed toward automation, which can help with case identification and treatment. Artificial Intelligence (AI) is an automation technology that can be utilized to identify Covid-19 illness at a low cost and with high accuracy.

In several fields, the globe has seen remarkable technological advancement in the recent decade. AI and deep learning have demonstrated their usefulness in other sectors, such as biomedical engineering, where biomedical engineers build and develop biomedical prototypes as medical instruments that aid in the diagnosis of a certain condition. The early detection of any disease is critical for providing the appropriate treatment, as the outcomes save human lives. As a result, doctors and physicians require such equipment and technologies.
To use Deep Learning to automate the Covid-19 diagnosis, the system requires a dataset of Chest X-Ray (CXR) pictures with a specific resolution. Pixels are organized in rows and columns to create these images. When these images are an ocular or other depiction of a real thing, a graphic or a picture that has all of the information needed to detect the case is used.

The goal of our work is to create a deep learning-based system for Covid-19 identification that is reliable and fast. As a result, we experiment we new methods : a Convolutional neural network (CNN) build from scratch, a pre-trained MobileNetV2 model, and a pre-trained ResNet50 model .

We divided our thesis into the following sections:
The first chapter introduces the concept of pictures and contrasts it with the problem of Covid-19 infection. It first and foremost covers information on the various components of images as well as approaches for segmentation in medical imaging.

The second chapter discusses the many forms of Deep learning and Transfer Learning, its functions, applications, and architectures.we also go through an overview of the previous researches made about detecting Covid-19 using Deep Learning and Transfer Learning.

The third chapter introduces the basic concept of our proposed techniques, as well as the components and functions that are used in the three models mentioned.

Our implementation outcomes for the three proposed options are shown in the fourth chapter.
First and foremost, it is divided into two parts: the first is concerned with the data, such as pre-processing, data augmentation, and data splitting into sets, which are validation, training, and test.
The second portion is about the model, such as training, compiling, and finally testing and obtaining results.

Chapter 1

Digital and Covid-19 image

1.1 Introduction

An image is a representation (ocular or others) of an actual item, a graphic, or a picture that contains all of the information necessary to create a live functional replica.

In the last two decades or so, low-cost, high-performance digital computers have become widely available and have been used to perform a wide range of tasks. Very sophisticated systems for creating, analyzing, and modifying imagery have been built and are being used in a variety of fields by coupling computers to imaging detectors and displays. For instance, they now are used to reconstruct X-ray and Magnetic resonance images (MRI) in medicine.

In this chapter, we will introduce images and some basic concepts concerning image treatment, processing and segmentation and their use in the medical field and relating it to the detection of Covid-19 and its methods.

1.2 Images

1.2.1 Basic image definition

An image is a visual representation of something. In information technology, the term has several usages. An image is a picture that has been created or copied and stored in electronic form. An image can be described in terms of vector graphics or raster graphics. Figure 1.1 is an example of basic color image.



Figure 1.1: Basic Image.

1.3 Digital image definition

A digital image is a computer file that can be viewed as an array of numbers. Its definition is the total number of dots (pixels) that make it up, and is given as the number of pixels in a row X number in a column.
In order to create an image which is digital, it is necessary to covert continuous data into digital form and in order to become suitable for digital processing, an image function f(x,y) must be digitized both spatially and in amplitude[1]. There are two steps in which it is done:

a) Sampling

In sampling the noise is reduced by taking samples on the coordinates (x). The more samples taken, the better the quality of the image is going to be, the noise would be more removed and same happens vice versa. Figure 1.2 represent image sampling .

Sampling has a relationship with image pixels. The total number of pixels in an image can be calculated as :

$$Pixels = \text{Total number of rows} \times \text{Total number of columns}$$

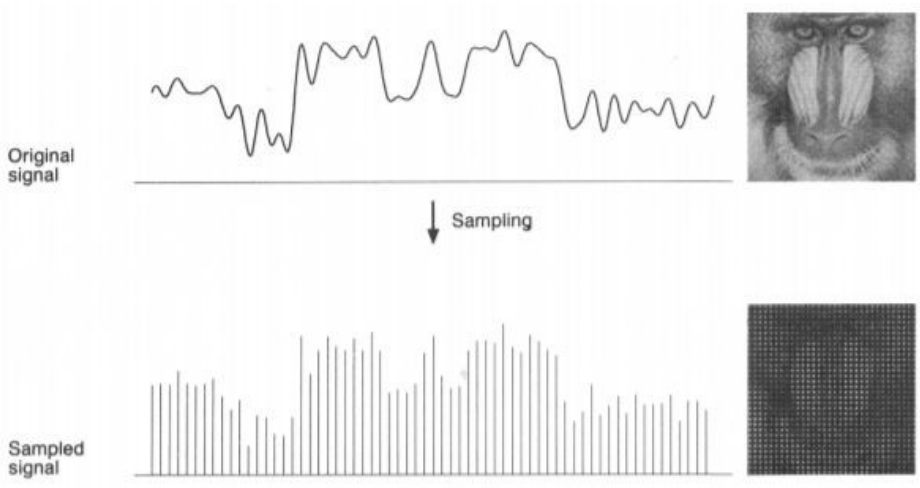


Figure 1.2: Representation of image sampling using graphs[1].

b) Quantization

It is the process of transforming a real valued sampled image to one, taking only a finite number of distinct values, the quantization level determines the number of grey levels in the digitized image. As represented in Figure 1.3.

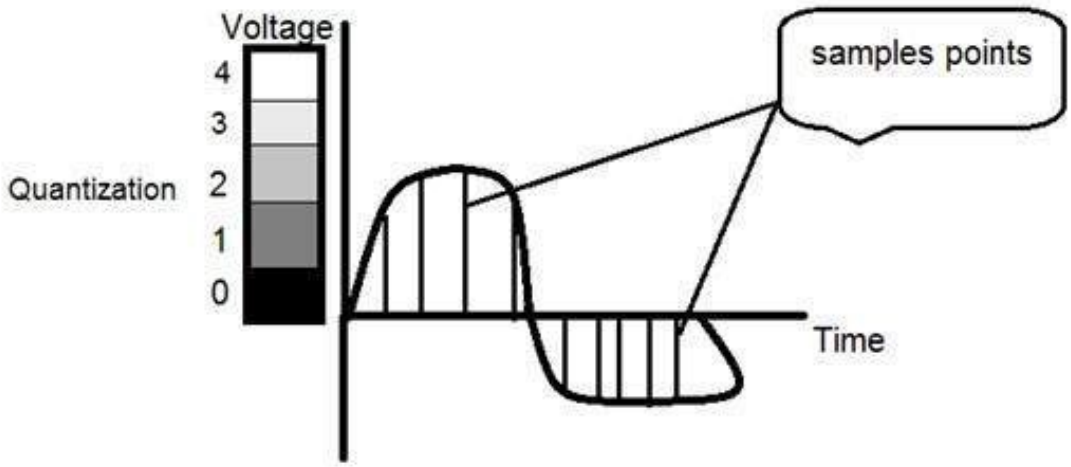


Figure 1.3: An example of a quantization process[1].

In the image shown above, these vertically ranging values have been quantized into five different levels or partitions, ranging from 0 (black) to 4 (white). This level could vary according to the type of image you want.

1.3.1 Types of digital images

All digital image files fall into one of two categories: **vector** or **raster**. Each format has advantages and disadvantages in different situations, so knowing the properties of each can help determine which format is the best choice for any project.

a) Vector images

A vector image -named a line mode image- is composed of individual mathematical geometry and primitive geometry that defines different characteristics like the form, position or colors [2]. Figure 1.4 represent a vector graphics.

Vector files are most useful when handling resizing, as they retain crisp detail when scaled to any size.

The Vector format is an excellent choice for digital illustrations such as logos.Common vector file extensions: SVG, EPS, and EMF.

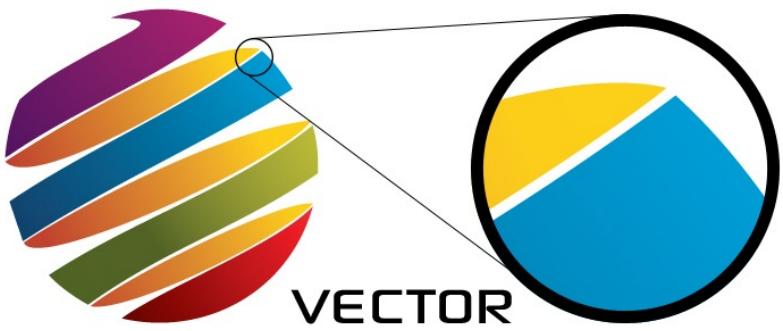


Figure 1.4: Vector Graphics.[2]

b) Raster image

A raster image (bitmap) it's an image build by a matrix or network table, each case has one color (colored pixels) [2]. it has a point of a juxtaposition of colors gathered in an image. Figure 1.5 represent a raster graphics.
Raster images are best for complex images like photographs, and are more likely to be a widely compatible file format.Common raster file extensions: JPG, PNG, and TIFF.

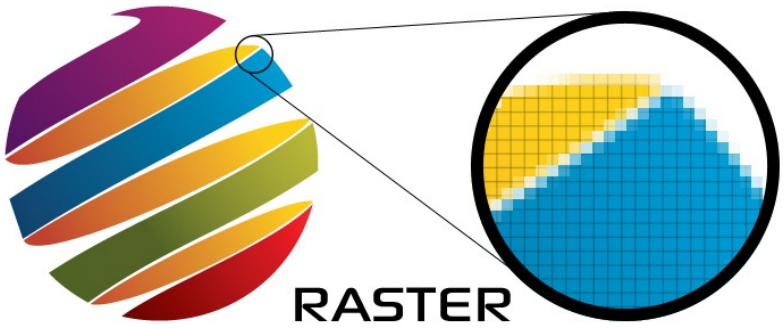


Figure 1.5: Raster Graphics [2].

The difference between vector graphics and raster graphics are:

- Raster graphics can reveal a multitude of colors in a single image and allow for more precise color editing. They showcase finer nuances in shading and light at higher resolutions. However, raster files, especially high resolution files, are very large.

- Vector-based graphics are more malleable than raster images — thus, they are much more versatile, flexible and easy to use. The most obvious advantage of vector images over raster graphics is that vector images are quickly and perfectly scalable.

1.4 Pixels

In digital imaging, a pixel (or picture element) is the smallest item of information in an image.
Pixels (abbreviated as "px") are also a unit of measurement commonly used in graphic and web design, equivalent to roughly 0.26 mm . This measurement is used to make sure a given element will display as the same size no matter what screen resolution views it.

1.4.1 The binary image

Binary images are images whose pixels have only two possible intensity values. They are normally displayed as black and white. Figure 1.6 represent a binary image.
the black color is represented by '0' and the white color is presented by '1'.

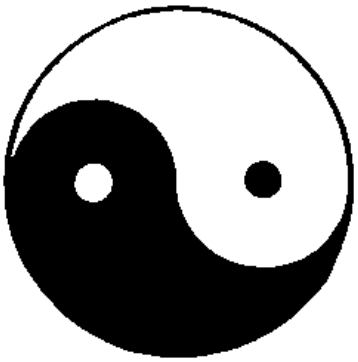


Figure 1.6: Binary Image.

1.4.2 Gray-scale image

Gray-scale Images are images whose pixels - similarly to binary images - have only two possible intensity values, they are normally displayed as black and white.
the most commonly used format is 8 bit and the range of this shade varies from 0 to 255. '0' stands for black while '255' stands for white and in-between for the different shades.

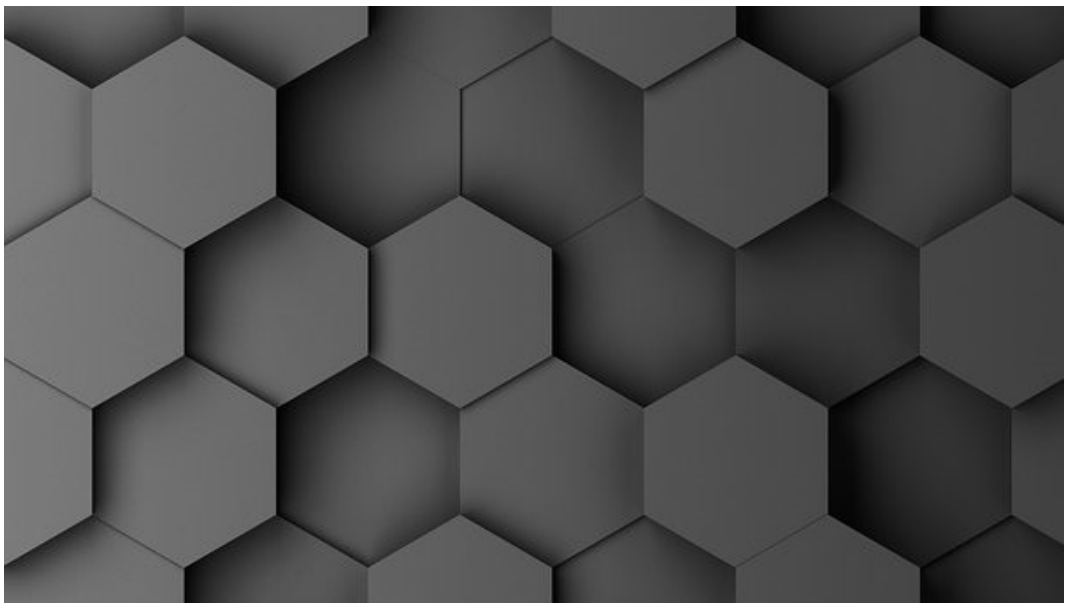


Figure 1.7: Gray-scale image.

1.4.3 Color image



Figure 1.8: Color image.

In a color image, each pixel has color information and is composed of three channels which are most commonly regarded as RGB mainly representing Red, Green, and Blue colors.
Each of these channels needs 8 bit for storage and hence, it would become 24 bits for each pixel. therefore, RGB together constitutes a pixel in a color image.

1.5 Image file format

Image file formats are a set of standards for organizing and storing digital images. It can store data in an uncompressed format, a compressed format (which may be lossless), or a vector format[3].
The list below outlines some of the more common file types and provides a brief description, how the file is best used.

Image Format	**Description**	**Use**
TIFF	Tagged Image File Format(.tif, .tiff), lossless image files(do not need to compress or lose any image quality or information).	High quality prints, professional publications, archival copies.
Bitmap	Developed by Microsoft for Windows(.bmp),no compression or information loss.	High quality scans, archival copies
JPEG	Joint Photographic Experts Groups(.jpg, .jpeg), a lossy format(compressed to make a smaller file), unnoticeable loss in quality.	Web Images, Non-Professional Printing, E-Mail, Powerpoint
GIF	Graphics Interchange Format files(.gif), limited only to 256 colors , allow transparency, can be animated and small in size and portable.	Web Images
EPS	Encapsulated PostScript file (.eps), a vector file type, can be opened in many illustration applications(Adobe Illustrator).	Vector artwork, illustrations
PNG	Portable Network Graphics files(.png), lossless image format and can handle up to 16 million colors.	Web Images
SVG	Calable vector graphics(.svg), resolution independent.	shapes, text, sketches, logos.
WebP	Lossless and lossy compression, supports transparency.(.webp)	Web Images
RAW	Unprocessed images created by a camera or scanner.(.raw, .cr2, .nef, .orf, .sr2)	Photography

Table 1.1: Common Image file Formats.

1.6 Image resolution

Image resolution is typically described in Pixels Per Inch (PPI), which refers to how many pixels are displayed per inch of an image.
Higher resolutions mean that there more PPI, resulting in more pixel information and creating a high-quality, crisp image [4]. lower resolutions mean that there are fewer pixels, and those few pixels are might be too large (usually when an image is stretched). Figure 1.9 shows the comparison between these three resolutions.

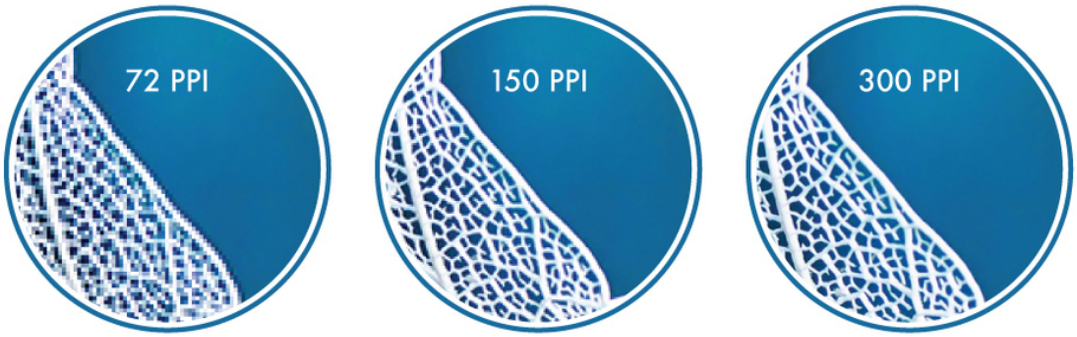


Figure 1.9: A comparison of screen resolutions for 72 PPI, 150 PPI, 300 PPI [4].

1.6.1 Spatial Resolution

It is a measure of the smallest object that can be resolved by the sensor and the ability to see fine detail in an image; it can be quantified by separation of two distinguishable points.
this level of resolution and optical instrument is dependent on the size of the aperture.
A spatial resolution of output devices, such as monitors or printers, is usually given in dots per inch (DPI) and a Line per inch (LPI) is usually used in laser printers.
Figure 1.10 shows the comparison between these three screen resolutions.

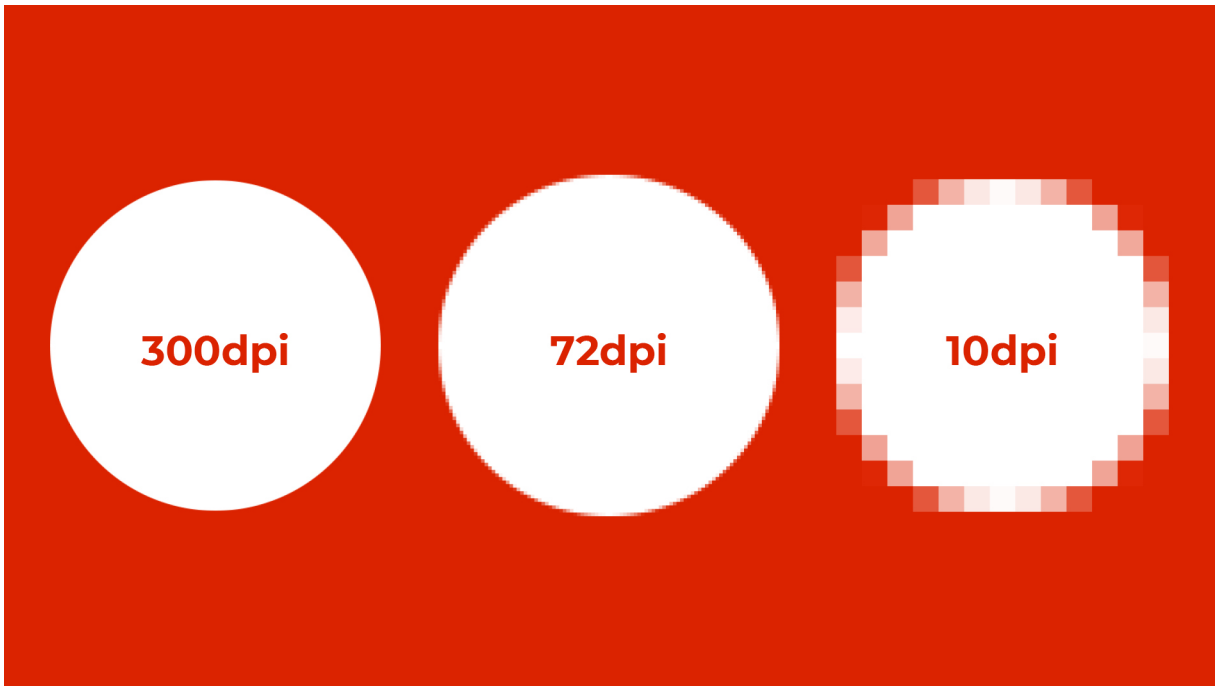


Figure 1.10: A comparison of screen resolutions for 300 dpi, 72 dpi, 10 dpi[4]

1.6.2 Gray level resolution

Gray-level resolution is a term that refers to the number of shades of gray utilized in preparing the image for display. Digital images having higher gray-level resolution are composed with a larger number of gray shades and are displayed at a greater bit depth than those of lower gray-level resolution .
Figure 1.11 shows the comparison between three Grey-level resolution.

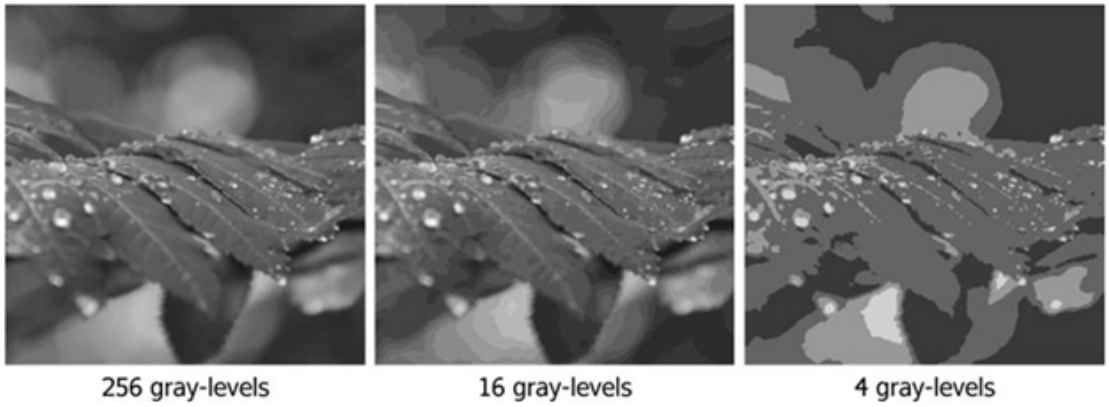


Figure 1.11: Gray-level resolution.

1.7 Image segmentation

1.7.1 Definition

Image segmentation is a method in which a digital image is broken down into various subgroups called Image segments which helps in reducing the complexity of the image to make further processing or analysis of the image simpler.

1.7.2 Groups of image segmentation

-Semantic segmentation is an approach detecting for every pixel, this is in stark contrast to classification, where a single label is assigned to the entire picture,it treats multiple objects of the same class as a single entity [5].

-Instance segmentation is the task of detecting and delineating each distinct object of interest appearing in an image.

1.7.3 Region approach

In this type of segmentation, some predefined rules are present which have to be obeyed by a pixel in order to be classified into similar pixel regions, related points with similar properties: gray intensity, color, texture, will be combined in the same set.

1.7.4 Edge detection approach

Edge detection method has the advantage of not necessarily using closed boundaries even though it does not have a significant improvement for multi-spectral images. It works by detecting discontinuities in brightness. Edge detection is used for image segmentation and data extraction in areas such as image processing, computer vision, and machine vision.

1.7.5 Filter

The filter is represented by a dimensioned matrix with coefficients that determine the filter properties. The image matrix is multiplied with the filter, resulting in a new matrix that corresponds to the treated image.

1.7.6 Image smoothing

Smoothing image, it is a filtering operation to eliminate image noise. there are filters used to smooth images, these filters are:

a- Linear filter
It is used to convert a set of input data to a set of output data according to convolution operation.

b- Medium filter
It is a filter used to average the value of the neighboring pixels. The result of this filter is a more blurred image.

c- Gaussian filter
It depends on the standard deviation of the image(distribution) and assumes the mean is zero (a mean different can be defined from zero as well). Gaussian filters do not take care of the edges.[6]

d- Mean Filter

It is the simplest low pass linear filter. Figure 1.12 describes how mean filter work.

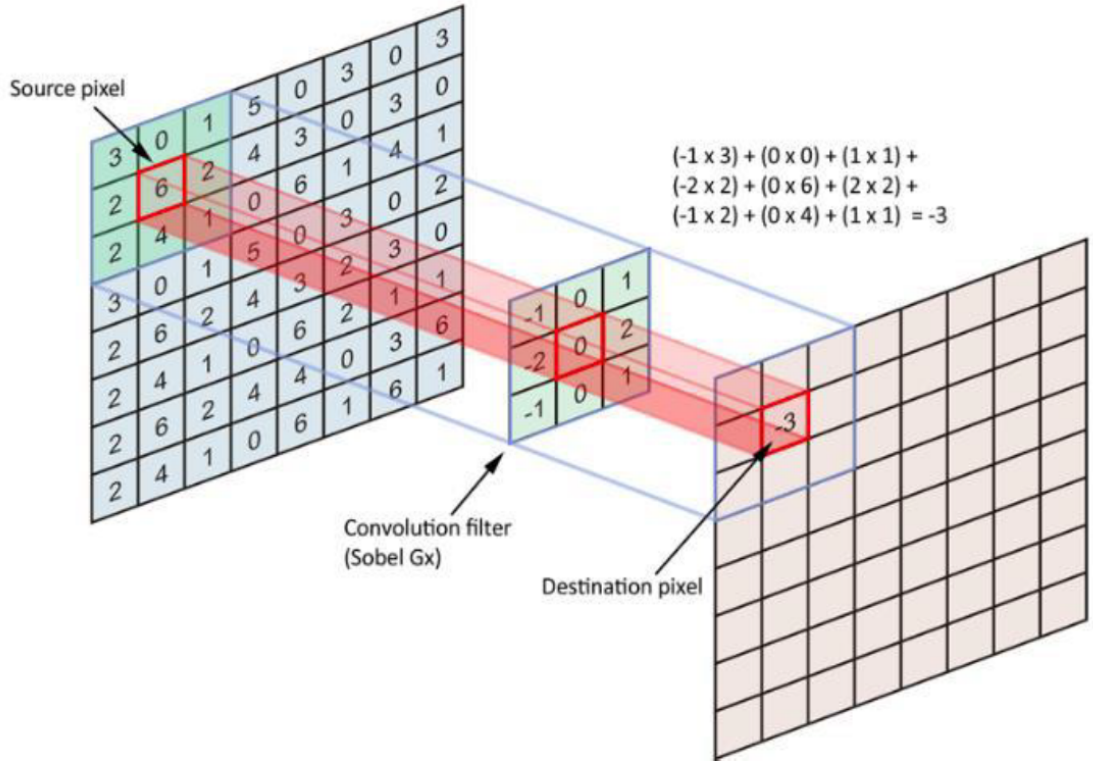


Figure 1.12: Mean filter application.

e- Median filter

The median filter is one of the most basic image-smoothing filters. It's a nonlinear filter that removes black-and-white noise present in an image by finding the median using neighboring pixels.[7]. This filter is used to attenuate isolated pixels, of a very different value from their neighborhood. This filter does not introduce blurring on the contours of particles unlike smoothing. Figure 1.13 shows the median filter application.

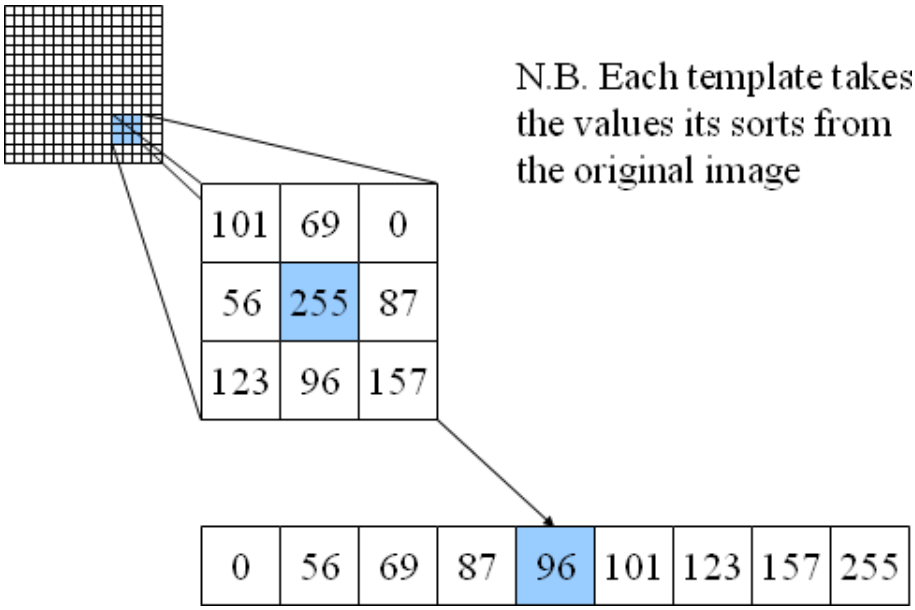


Figure 1.13: Median filter.

Working of 3x3 median filter; in the filtering process the figure pixel value 255 is replaced by 96 which is the median value of the 8 neighborhood pixels.

1.7.7 Image noise

Image noise is random variation of brightness or color information in images, and is usually an aspect of electronic noise. It can be produced by the image sensor and circuitry of a scanner or digital camera. Image noise can also originate in film grain and in the unavoidable shot noise of an ideal photon detector. For example, some noise forms because of how light arrives at the sensor while some is generated from converting the charger of each pixel into digital information.

1.7.8 Convolution

The act of adding each element of the picture to its nearby neighbors, weighted by the kernel, is known as convolution. This has something to do with a type of mathematical convolution. Despite being similarly characterized by the term "convolution," the matrix operation being performed is not typical matrix multiplication. Convolution is a technique for extracting picture properties such as form and edge, as well as for other purposes.
There are two types of convolution:

- **Valid convolution**
  When padding == "VALID", the input image is not padded. This means that the filter window always stays inside the input image, there can be a loss of information. Generally, elements on the right and the bottom of the image tend to be ignored.

- **Same convolution**
  When padding == "SAME", the input is half padded. The padding type is called

SAME because the output size is the same as the input size(when stride=1).
Using 'SAME' ensures that the filter is applied to all the elements of the input. Normally, padding is set to "SAME" while training the model. Output size is mathematically convenient for further computation.

1.7.9 Contour

Contour detection is a curve that connects all continuous points and has colors and intensities. The contours are a useful tool for analyzing shapes, identifying objects, and recognizing objects. The contours can detect the edges of photos, lowering data quality and obliterating some information.
There are three contours type:
-Step: the outline is sharp (ideal contours).
-Ramp: the contours are more blurred.
-roof: it is a line on a uniform background.
Figure 1.14 shows the different contour types.

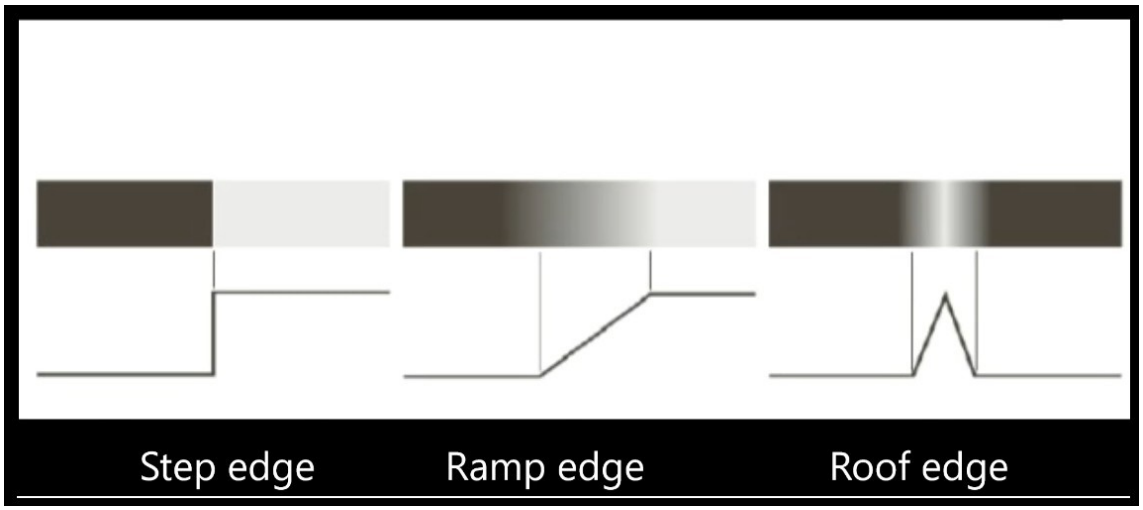


Figure 1.14: Different contour types.

1.8 Medical imaging

Medical imaging refers to several different technologies that are used to view the human body in order to diagnose, monitor, or treat medical conditions.
The development of technology in medical imaging has helped doctors to make easy and less life threatening therapeutic decisions, medical imaging techniques have many advantages, especially the pathological visualization entity itself, allowing a better understanding and orientation of the diagnosis.

1.8.1 Medical imaging technologies

a- History of medical imaging The first Medical imaging was elaborated by Wilhelm Conrad Rontgen (1845-1923) called X-ray. Working with an early cathode ray tube called a Crooke's tube, he noticed that the invisible rays were able to penetrate

some solids (like human flesh) better than others (like bone or metal).
After that there was a lot of development like X-ray tomography that was introduced in the 1940s, allowing "tomograms" or slices to be obtained through tissues without the over or under-lying tissue's being seen.
In 1972 the British radiologists Allan Mc Kornack (1924-1998) and Godfrey N. Hounsfield (1918), developed a scanner with digital image processing.
Doctors can use medical imaging to peek into your body from the outside. We take it for granted now, but detecting injuries and diseases before imaging was sometimes difficult or impossible. People all across the world benefit from various imaging technologies, from the first X-rays obtained in 1896 to today's advanced radiology treatments.

b- Imaging modalities There are many techniques of medical imaging processing used like magnetic resonance imaging (MRI), X-ray, digital mammography and others on . For clinical usage, images are now almost collected and stored in a digital representation, Medical imaging classify be into these categories:

- **X-Ray**
  X-ray is a snapshot of a specific parts of the body whether it's the chest or the abdomen or a specific joint like the knee or the thigh. It basically a beam of X-rays (highly charged electrons) that is passed through the body.
  the high-density object will appear white (e.g. bone)
  -the low-density objects will appear black (e.g. lung)

- **CT Scan** It is a series of cross-sectional X-rays taken from a 360 degree view around the patient, which helps create slice by slice maps of the patient. It also helps for looking to bone soft tissue and vessels all at the same time without separating them. The high-density object will appear white while the low-density objects will appear black.

- **Intravascular Ultrasound (IVUS)**
  High-frequency waves are bounced off of tissues in the human body, used to create a cross-sectional image of a specific part of the body, it is most effective in visualizing body cavities, tissues, and blood vessels.
  A growing number of cardiologists think that new information yielded by IVUS can make a significant difference in how the patient is treated and can provide more accurate information which will reduce complications and incidence of heart diseases.
  The figure 1.15 below generated in real-time, yields information that is not possible with routine imaging methods or even non-invasive Multi-slice CT scans.

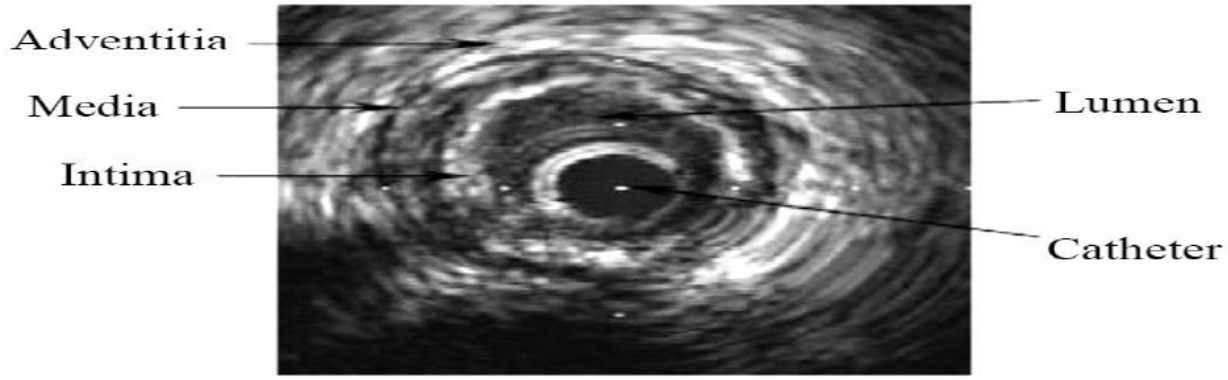


Figure 1.15: Typical IVUS image.

- **Magnetic resonance imaging (MRI)**
  Magnetic resonance imaging, invented in 1970, used for human body visualization and can be adapted to brain, spinal cord and muscles images. Clinical MRI is based on the hydrogen nucleus due to its abundance in the human body and its magnetic resonance sensitivity.
  For image formation, a large static magnetic field is used to perturb magnetically moments of the proton that exist in the hydrogen nucleus from their equilibrium and observing how perturbed moments relaxes back to their equilibrium. Naturally, the protons are oriented randomly. But in the existence of a static magnetic field, they line up with the field and the net magnetization of protons tends toward the direction of the field. In the existence of enough energy, it is possible to make the net magnetization zero. In the relaxation process, and the induced electronic signal is recorded. The strength and duration of the signal depend on three quantities:
  1. ρ (proton density)
  2. Spin-lattice relaxation time: the time which describes how fast the net magnetization
  takes to relax back to its equilibrium (T1).
  3. Spin-spin relaxation time: with this time, magnetization components decrease to zero (T2).

1.8.2 Medical image segmentation

Medical image segmentation is the isolation of scenes into the component regions, and there are segmentation techniques used to extract an attribute from an image by partition it into multiple regions. Segmentation is achieved by subjective or objective methods that associate and aggregate locally homogeneous image elements based on some measurable feature (or features) such as intensity, texture, gradient or many others. segmentation techniques are very important in discovering the region of interest.

1.9 Computer vision

Computer vision is an important technique to provide the required information in diagnosis and treatment. the first thing we need to consider for medical images is to use image processing techniques including image enhancement, color image processing, wavelet and multi-resolution image processing, image compression, morphological image processing ,and image segmentation.
Both image enhancement and segmentation can be considered as low-level computer vision. For mid-level computer vision, computers can organize the knowledge (information) acquired in the low-level vision to make some useful decisions. For high-level vision, computers must further provide some "thinking" capabilities like humans do.

1.10 Covid-19

The coronavirus disease (Covid-19) is a global pandemic and an infectious disease caused by the SARS-CoV-2 virus that was discovered by a Chinese physician in Wuhan, the capital city of Hubei province in mainland China, in December 2019.
It infects the lungs as most people experience mild to moderate respiratory illness and recover without requiring special treatment. However, some become seriously ill and require medical attention and potentially suffer of deadly respiratory syndromes.[8]

1.10.1 Covid-19 symptoms

People with COVID-19 have had a wide range of symptoms reported – ranging from mild symptoms to severe illness. Symptoms may appear 2-14 days after exposure to the virus. Anyone can have mild to severe symptoms. People with these symptoms may have COVID-19[9]:
Fever or chills, Cough, Shortness of breath or difficulty breathing, Fatigue, Muscle or body aches, Headache, New loss of taste or smell, Sore throat, Congestion or runny nose, Nausea or vomiting and Diarrhea.

1.10.2 Covid-19 Timeline :

Figure 1.16 represent Covid-19 timeline from it's first appearance in December 2019 till March 2022. Also mentioning the different places in the world according to the most worldwide appearing cases.

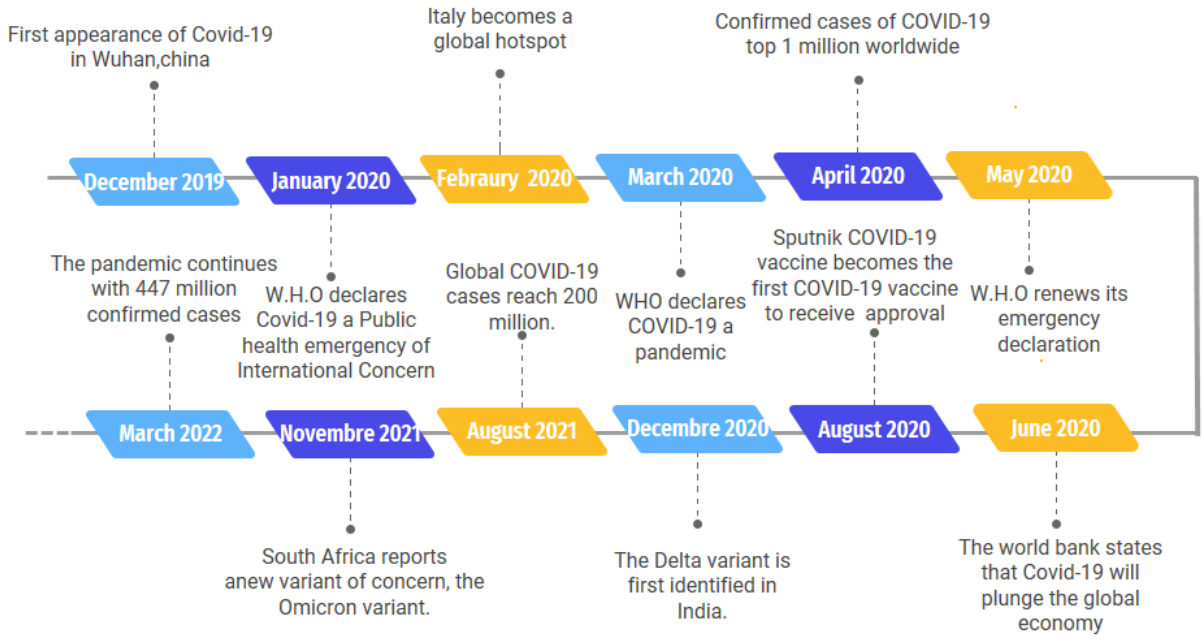


Figure 1.16: Covid-19 timeline.

1.10.3 Covid-19 Images

Different medical imaging methods are used to identify Covid-19 infections like X-ray, ultrasound images, MRI and abdominal computed tomography (CT).

- **X-Ray images**

  Radiation is usually used in X-rays (radiography) to create a two-dimensional picture. Usually, when a patient has indications of Covid-19, like fever, cough, or shortness of breath, they are recommended to get a Chest X-Ray. The most unusual finding is "ground-glass opacities", which suggests that some portion of lungs resembles a hazy shade of grey instead of being black with excellent white lung markings for blood vessels.
  In critical type COVID-19 patients, multifocal or diffuse consolidation can be seen in both lungs, showing as "white lung". Though the CXR is not very sensitive for mild types of patients, it has been proven effective previously in other types of corona viruses, such as severe acute respiratory syndrome (SARS) and the middle east respiratory syndrome (MERS)[10].
  They are usually performed in hospitals by a radiographer using fixed equipment, although they can also be performed using portable devices. Pooled results showed that chest X-ray correctly diagnosed COVID-19 in 80.6% of people who had COVID-19. However, it incorrectly identified COVID-19 in 28.5% of people who did not have COVID-19.
  Figure 1.17 shows a chest X-ray of a patient with Covid-19.



Figure 1.17: An X-ray of lungs of patient with Covid-19.

- **Ultra-sound**

  Ultrasound scans use high-frequency sound waves to produce an image. Pooled results showed that lung ultrasound correctly diagnosed COVID-19 in 86.4% of people with COVID-19. However, it incorrectly diagnosed COVID-19 in 45% of people who did not have COVID-19.
  Figure 1.18 mention the 4 different ultra-sound scores.

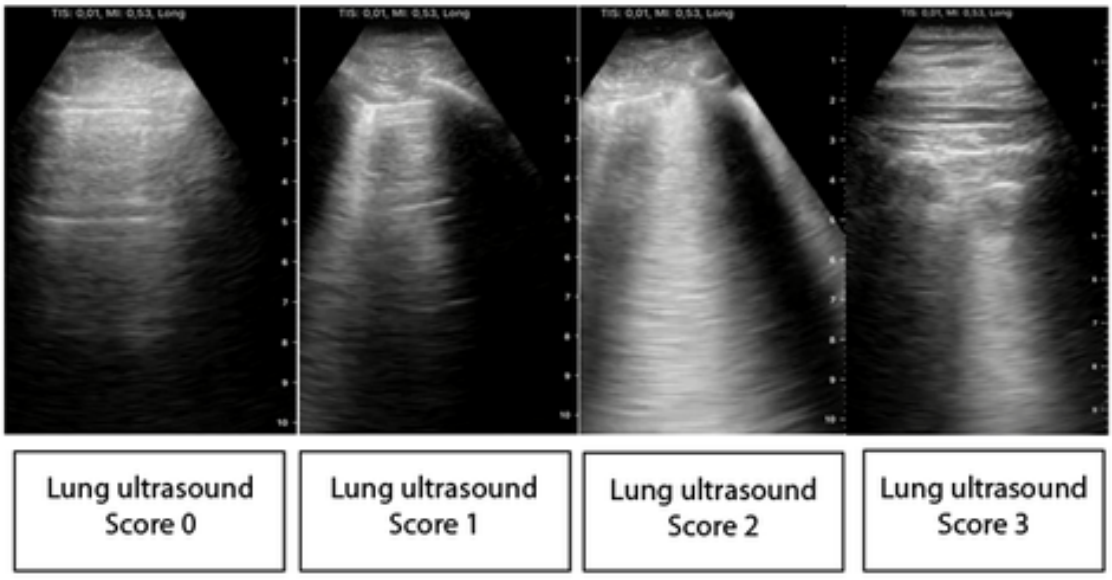


Figure 1.18: Ultra sound of the lungs of a patient with COVID-19.

- **Abdominal computed tomography**

  CT scans combine two-dimensional X-ray pictures into a three-dimensional image using a computer. They require specialized equipment and are performed by a trained radiographer in a hospital setting.
  Pooled results showed that chest CT correctly diagnosed COVID-19 in 87.9% of people who had COVID-19. However, it incorrectly identified COVID-19 in 20% of people who did not have COVID-19.
  Figure 1.19 represent a baseline ct images of a 50 years old patient.

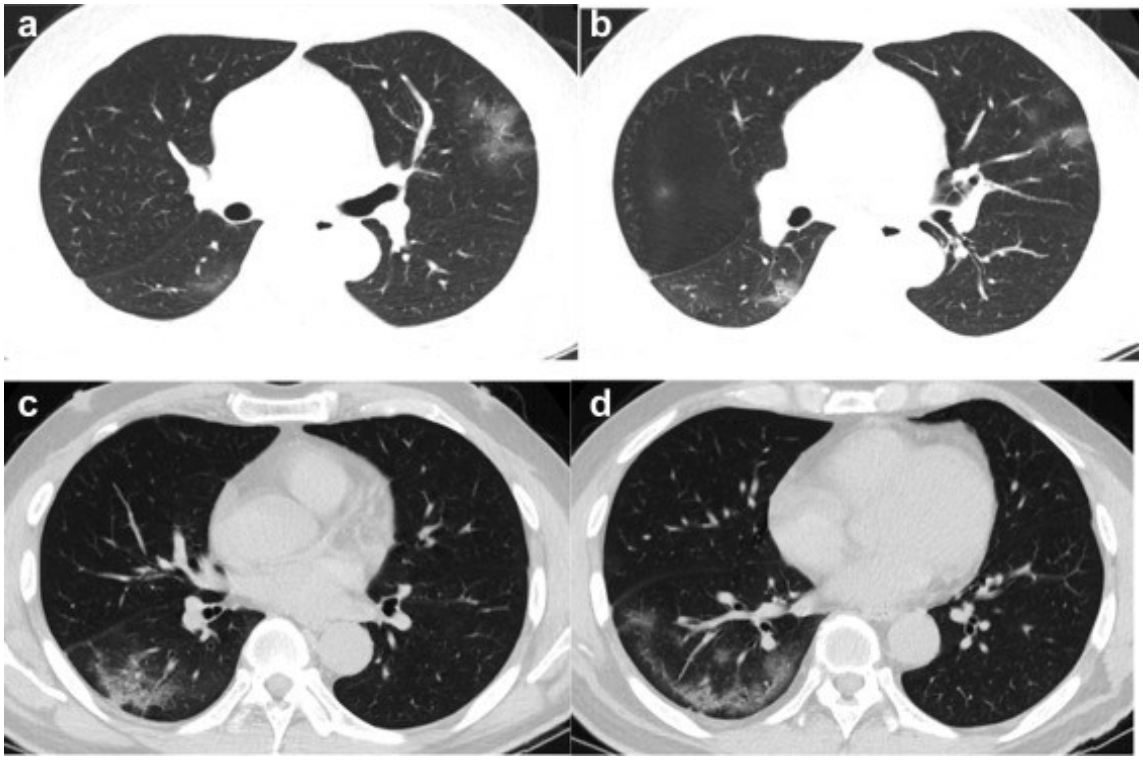


Figure 1.19: Baseline CT images (a-d) of a 50 year-old man admitted for symptoms of fever for 1 day.There were multiple patchy ground-glass opacities in the left upper lobe and right lower lobe (a, b). Air bronchogram and crazy paving pattern can be seen (c). Lesions are located in the peripheral area of the lung (d).

1.10.4 Covid-19 detection

Currently, several performance evaluations of SARS-CoV-2 IVDs are being carried out by regulatory authorities, reference laboratories and other stakeholders in various regions. Manufacturers are strongly encouraged to participate in initiatives which generate evidence that can be used to support their Emergency Use Listing (EUL) submission.

Polymerase chain reaction

The polymerase chain reaction (PCR) test for COVID-19 is a molecular test that analyzes your upper respiratory specimen, looking for genetic material (ribonucleic acid or RNA) of SARS-CoV-2, the virus that causes COVID-19. Scientists use the PCR technology to amplify small amounts of RNA from specimens into deoxy ribonucleic acid (DNA),

which is replicated until SARS-CoV-2 is detectable if present. The PCR test has been the gold standard test for diagnosing COVID-19 since authorized for use in February 2020.which is accurate and reliable, as shown in figure 1.20.

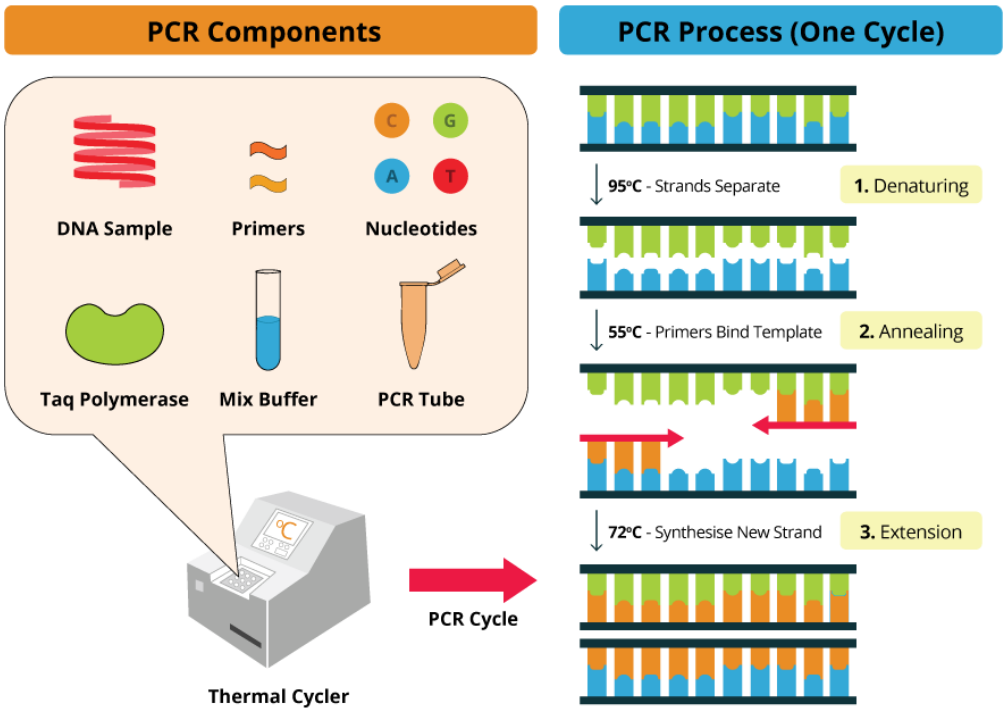


Figure 1.20: PCR and molecular biology fundamental principals.

Antigen rapid diagnostic tests

Unlike PCR, Ag-RDT are:
• less expensive compared to PCR, no additional reagents or consumables required;
• able to produce patient results in 15 to 30 minutes;
• easy to use - no complex manipulations required;
• suitable for any setting where biosafety measures are ensured and storage conditions for tests are suitable. Figure 1.21 below is an exemple of RDT detection technique.



Figure 1.21: RDT.

1.11 Conclusion

The fundamental concept of pictures, digital images, and medical imaging are presented in this chapter, along with the problem of the COVID-19 virus.
It first and mainly covers information on the various components of pictures as well as approaches for segmentation in medical imaging.
Furthermore, next chapters considers many deep learning algorithms for categorizing COVID-19 identification, with the methods of classification and other neural networks presented and discussed in the following chapters.

Chapter 2

Deep learning and transfer learning-based models

2.1 Introduction

Due to the vast amount of data that specialists must process, a person may make multiple errors and waste a substantial amount of time diagnosing a specific case in the pathology sector. To identify disease at a low cost, with high efficiency, and in a short amount of time, we opted to use deep learning for pathological field-related diagnostics. The goal of this chapter is to introduce machine learning, deep learning, and their many applications, as well as Covid-19 detection.

2.2 Machine learning

Machine learning is a type of data analysis that automates the creation of analytical models. It's a field of artificial intelligence based on the premise that computers can learn from data, recognize patterns, and make judgments with little or no human input.
Researchers interested in artificial intelligence wanted to see if computers could learn from data. The iterative aspect of machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results.
there are two basic approaches in AI: supervised learning and unsupervised learning.

2.2.1 Supervised Learning

Supervised learning is a machine learning approach that uses datasets designed to train or "supervise" algorithms into classifying data or predicting outcomes accurately. Using labeled inputs and outputs, the model can measure its accuracy and learn over time.
Supervised learning can be further divided into two types:

- **Classification**
  Classification is a process of finding a function which helps in dividing the dataset into classes based on different parameters.

- **Regression**
  Regression is a process of finding the correlations between dependent and independent variables. It helps in predicting the continuous variables.

2.2.2 Unsupervised Learning

Unsupervised learning uses machine learning algorithms to analyze and cluster unlabeled data sets. These algorithms discover hidden patterns in data without the need for human intervention.
Unsupervised learning can be further grouped into types:

- **Clustering**
  Clustering is the method of dividing the objects into clusters that are similar between them and are dissimilar to the objects belonging to another cluster.

- **Association**
  Association is a rule-based machine learning to discover the probability of the co-occurrence of items in a collection.

2.3 Deep learning

2.3.1 Definition

Deep learning is an artificial intelligence discipline that focuses on building huge neural network models that can make accurate data-driven decisions simulating the behavior of the human brain [11]. It is particularly suited when the data is complex and where there are large datasets available.
Many artificial intelligence (AI) apps and services rely on deep learning to increase automation by executing analytical and physical tasks without the need for human participation.

Deep Learning uses layers of algorithms to process data, understand human speech, and visually recognize objects. Information is passed through each layer, with the output of the previous layer providing input for the next layer.
The first layer in a network is called the input layer, while the last is called an output layer. All the layers between the two are referred to as hidden layers.Figure 2.1 shows The structure of a deep learning network .

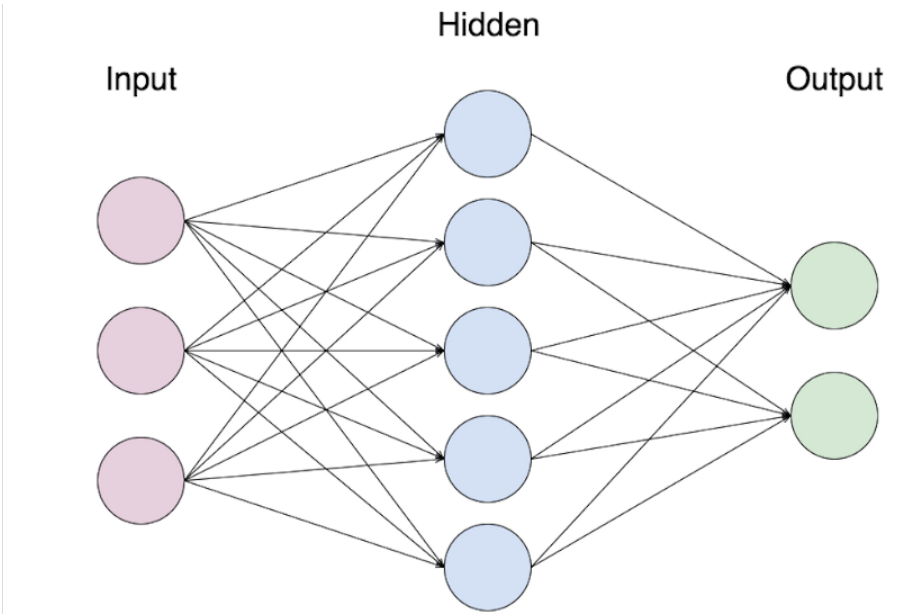


Figure 2.1: The structure of a deep learning network .

2.3.2 Deep Learning Applications

Deep learning technology lies behind everyday products and services. Among other things, Microsoft utilizes deep learning for image search and also for machine translation. Smart phones as well have deep learning systems running on them; for example, deep learning is now the standard technology for speech recognition, and also for face detection on digital cameras.
In the healthcare sector, deep learning is used to process medical images (X-rays, CT, and MRI scans) and diagnose health conditions.

2.3.3 Deep learning techniques

Six techniques we are going to talk about :

Artificial Neural Networks

Neural networks, also known as Artificial neural networks (ANNs) or Simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.

Deep Neural Networks

A Deep neural network (DNN) is an ANN with multiple layers between the input and output layers.No matter the various types of DNN , its components are : neurons, synapses, weights, biases, and functions. Figure 2.2 illustrate it's architecture.

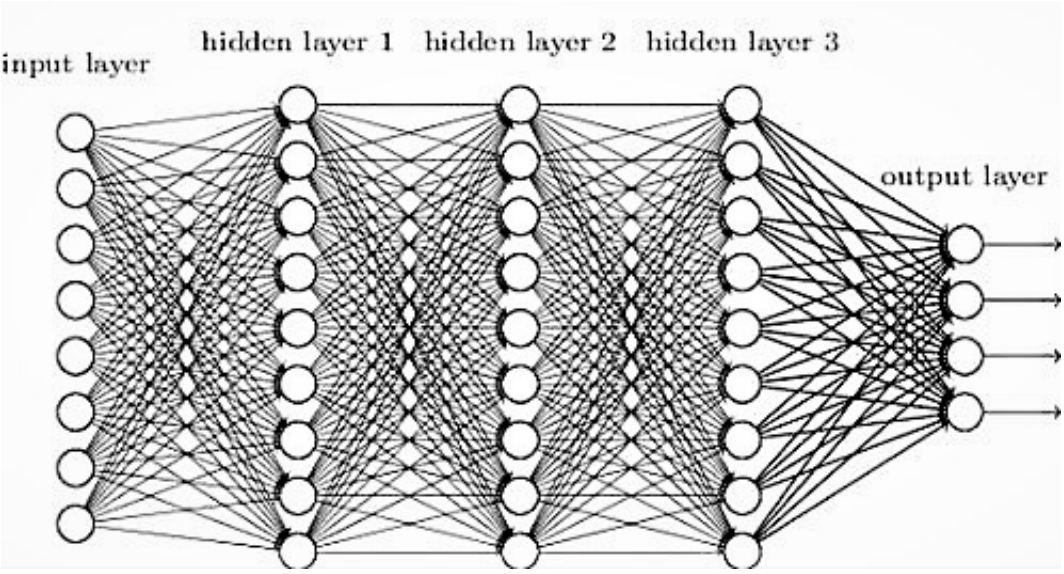


Figure 2.2: DNN architecture.

Convolutional Neural Networks

Convolutional neural networks are set apart from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are: convolutional layers, pooling layers and fully-connected layers. When these layers are stacked, a CNN architecture is formed.
A simplified CNN architecture for MNIST classification is illustrated in the figure 2.3 below

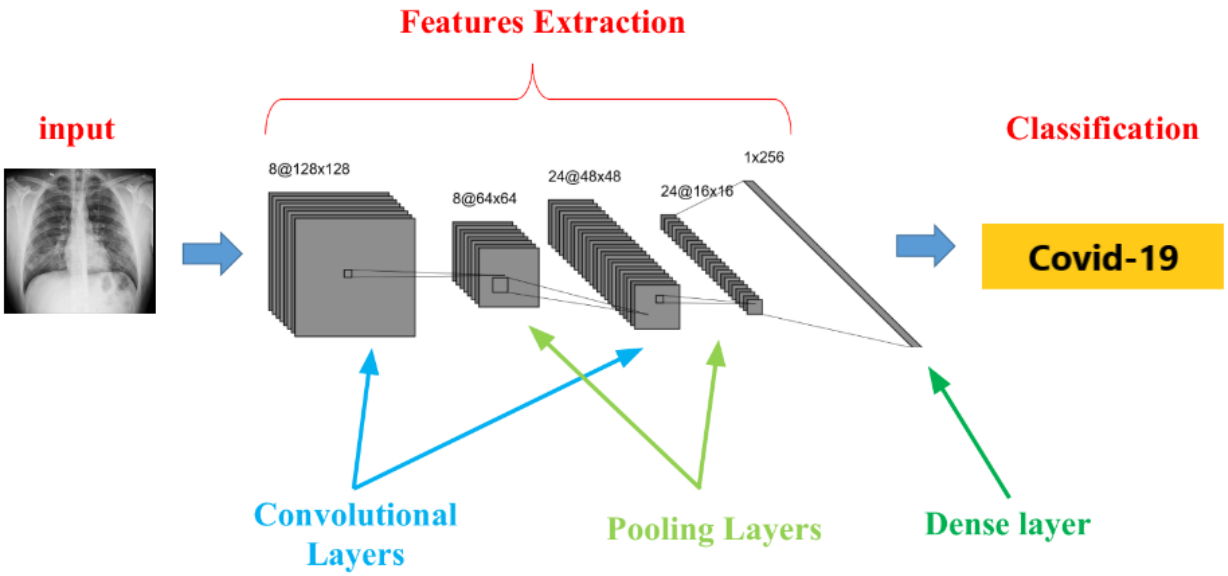


Figure 2.3: simple CNN architecture of five layers

- **convolutional layer** The convolutional layer is the core of CNN, and it is where most of computation occurs. It requires a few components, which are input data, a filter, and a feature map. Convolution is a specialized type of linear operation used for feature extraction, where a small array of numbers, called a kernel, is applied across the input, which is an array of numbers, called a tensor.As mentions in figure 2.4 below.

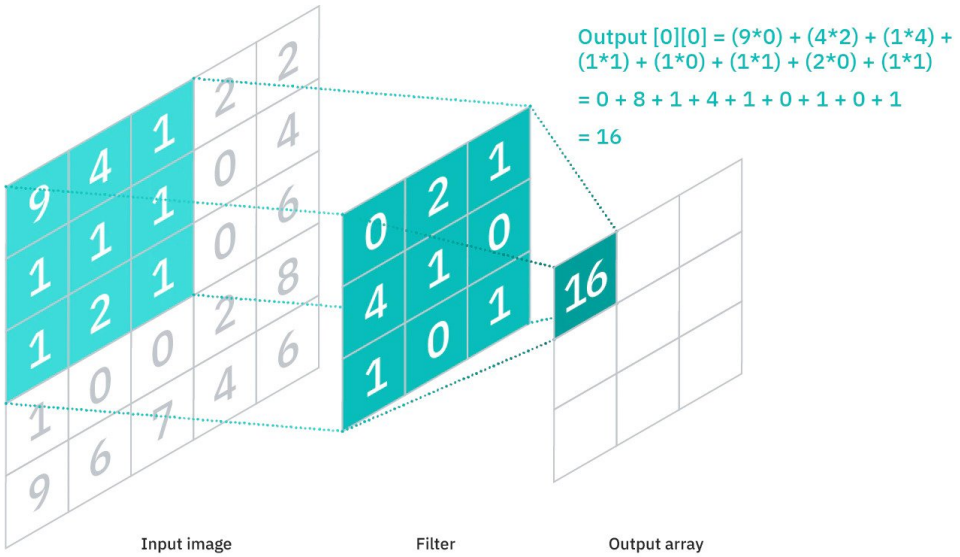


Figure 2.4: Convonlutional layer [12].

- **pooling layer** Pooling layers (or down-sampling) is a dimensional reduction technique that reduces the number of parameters in the input. the pooling process sweeps a filter across the whole input, but unlike the convolutional layer this filter does not have any weights.
  There are two main types of pooling:
  Max pooling: As the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an aside, this approach tends to be

used more often compared to average pooling.
Average pooling: As the filter moves across the input, it calculates the average value within the receptive field to send to the output array.Figure 2.5 illustrate the difference.

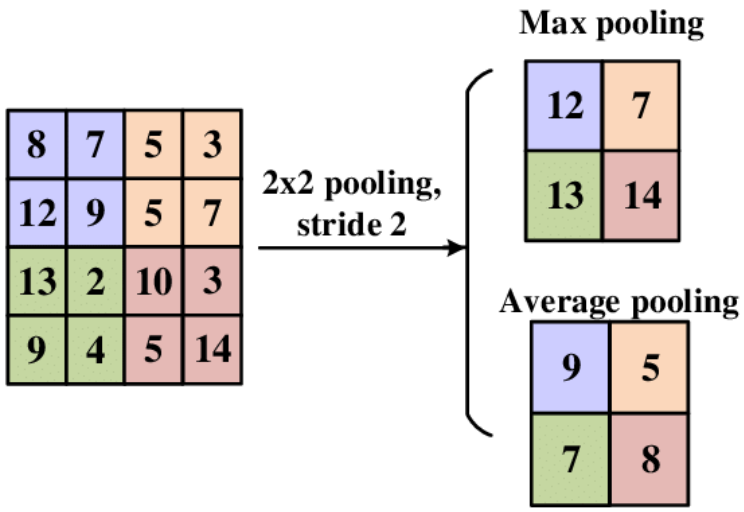


Figure 2.5: Pooling layer.

- **Fully-connected layer** in the fully-connected layer, each node in the output layer connects directly to a node in the previous layer.
  This layer performs the task of classification based on the features extracted through the previous layers and their different filters. While convolutional and pooling layers tend to use ReLu functions, FC layers usually leverage a softmax activation function to classify inputs appropriately, producing a probability from 0 to 1. a shown in Figure 2.6 below.

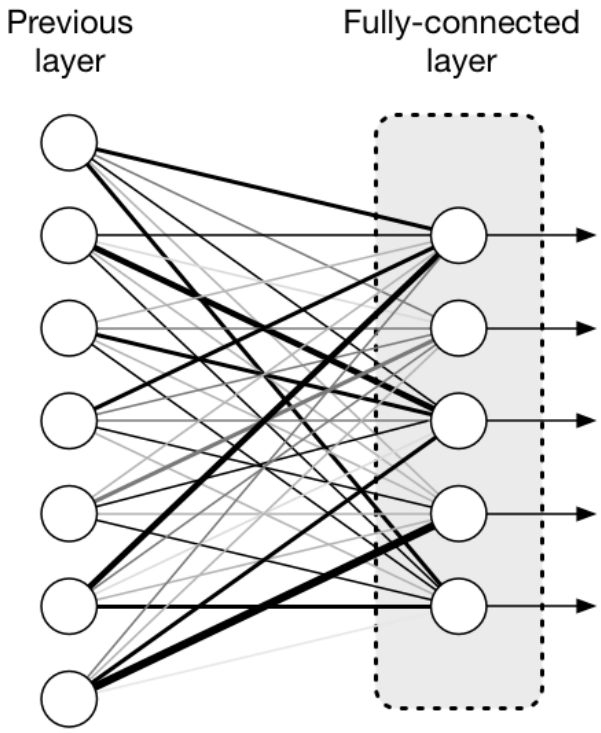


Figure 2.6: Fully connected layer.

Recurrent Neural Networks

A Recurrent neural network (RNN) is a class of neural networks naturally suited to work for time series data or data that involves sequences. RNNs have the concept of 'memory' that helps them store the states or information of previous inputs to generate the next output of the sequence.
In some of the applications like text processing, speech recognition and DNA sequences, the output depends on the previous computations. Figure 2.7 illustrate a simple RNN architecture.

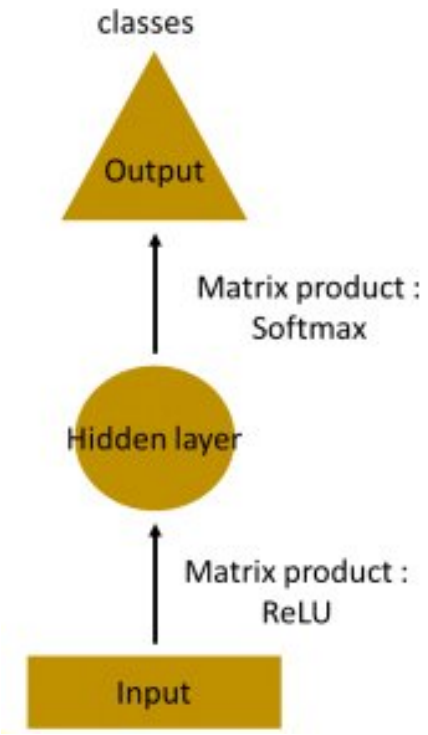


Figure 2.7: a simple RNN architecture.

Long Short-Term memory

Long short-term memory (LSTM) is an advanced form of RNN architecture that uses special units to deal with the vanishing gradient problem, which was introduced by Hochreiter et al[13]. A memory cell in an LSTM unit can store data for long periods and the flow of information into and out of the cell is managed by three gates.Figure 2.8 describes it architecture.

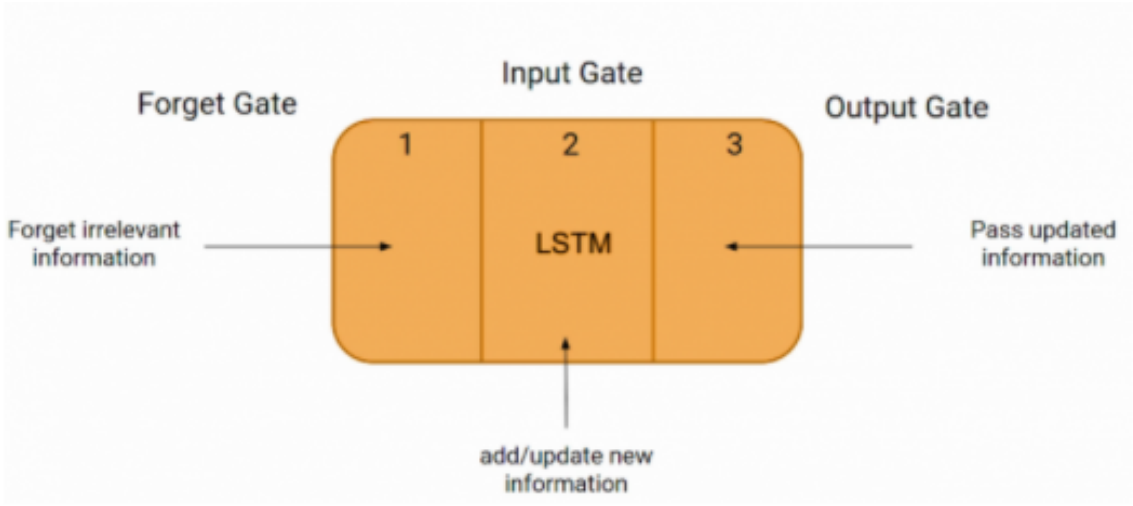


Figure 2.8: A LSTM architecture

AutoEncoder

Autoencoders are a specific type of feedforward neural networks where the input is the same as the output. They compress the input into a lower-dimensional code and then

reconstruct the output from this representation. The code is a compact "summary" or "compression" of the input, also called the latent-space representation.
An autoencoder consists of 3 components: encoder, code and decoder. The encoder compresses the input and produces the code, the decoder then reconstructs the input only using this code.as mentioned in Figure 2.9 .

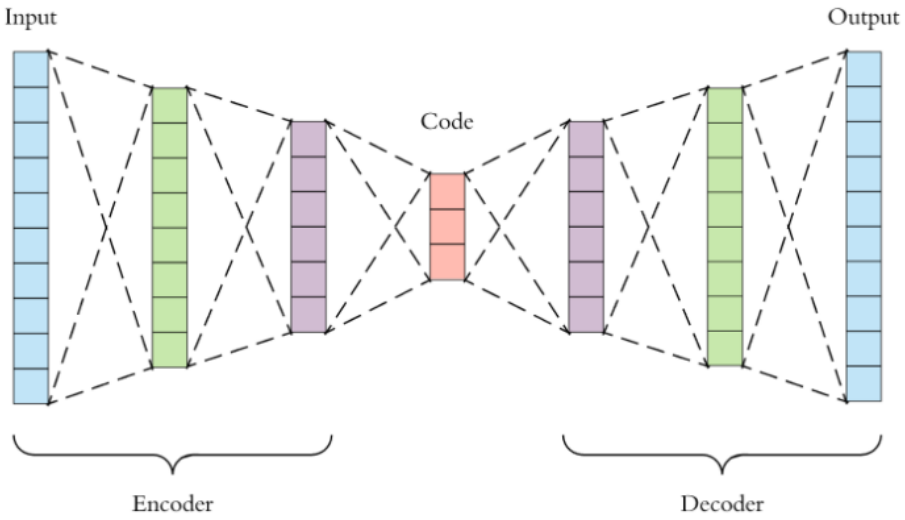


Figure 2.9: An Auto Encoder architecture.

2.4 Difference between machine learning and deep learning

First and foremost, whereas typical machine learning methods, such as linear regression or a decision tree, have a relatively simple structure, deep learning is based on an artificial neural network. This multi-layered ANN is complicated and interwoven, just like a human brain.

Deep learning algorithms, on the other hand, require far less human intervention. For example : if a STOP sign image recognition technique was based on a typical machine learning algorithm, a software engineer would manually select features and a classifier to classify photographs, assess whether the output is as expected, and change the algorithm if necessary. However, in a deep learning system, the features are automatically extracted, and the algorithm learns from its own mistakes.

Thirdly, deep learning takes far more data to work successfully than a standard machine learning algorithm. Machine learning uses a thousand data points, but deep learning uses millions. A deep learning system requires a huge dataset to minimize fluctuations and produce high-quality interpretations due to its complicated multi-layer structure.Shown in Figure 2.10 below.

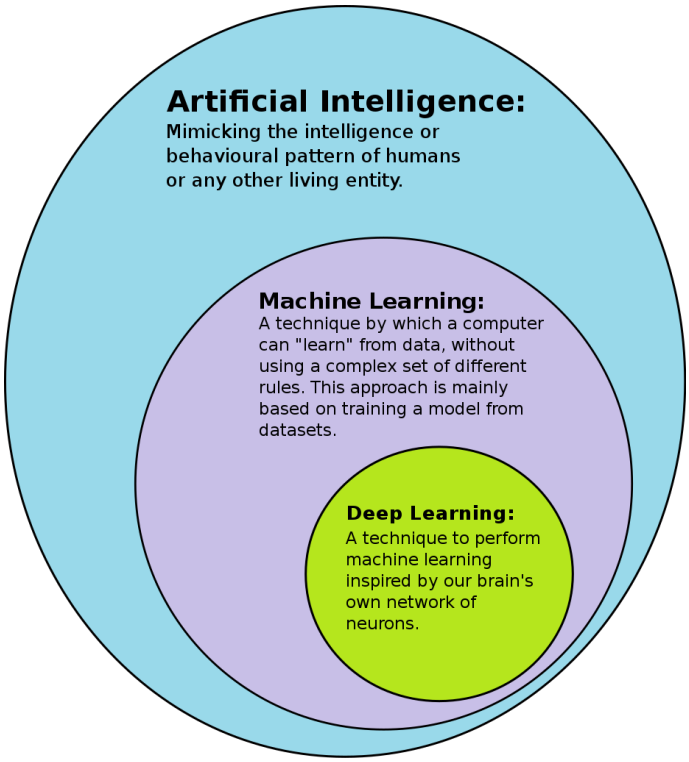


Figure 2.10: The difference between AI,machine learning and deep learning.

2.5 Network Architectures

Deep neural networks generally extract high-level features for classifying the image. but these networks have some limitation to classify image classification and object detection, because they use small sets of data for training and classification.
Each network is designed for a given input modality, which encodes assumptions about the data's structure.

2.5.1 Convolutional Neural Network

It refers to a special type of multilayer neural network whose design is intended to directly recognize the visual patterns in pixel images with preprocessing that is minimal.It will separate features, allowing for faster and more precise grouping. The convolution, non-linearity (ReLU), Pooling, and classification layers are the four fundamental layers of the CNN.

2.5.2 VGG Network

VGG stands for Visual Geometry Group, it is a multi-layer deep CNN architecture. The term "deep" refers to the number of layers in VGG-16 or VGG-19, which have 16 or 19 convolutional layers respectively.
It constructed with 13 convolutional layers and three fully connected layers. It uses an effortless way to build the system. The VGG network is designed considering the appropriate setting of layers depth that do not increase the complexity of the network.Shown in Figure 2.11 below.

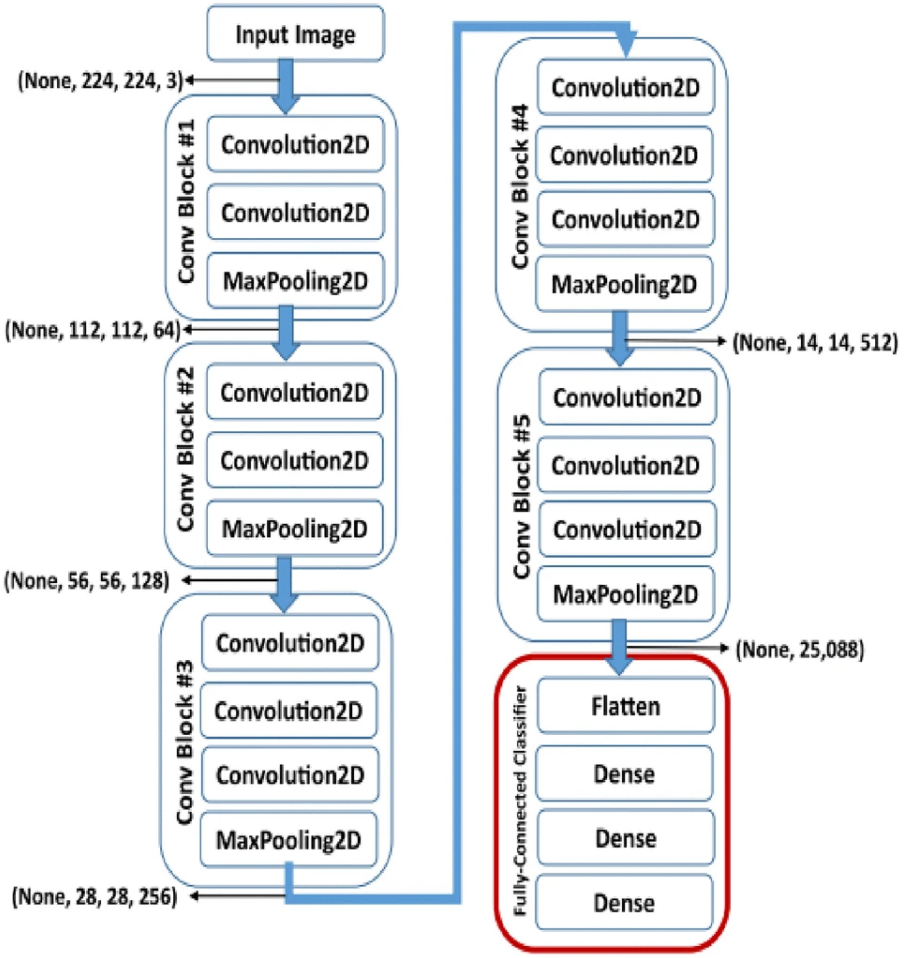


Figure 2.11: VGG Layers.

2.5.3 ResNet

Residual Network (ResNet) is one of the famous deep learning models that was introduced by Shaoqing Ren, Kaiming He, Jian Sun, and Xiangyu Zhang in their paper. The paper was named "Deep Residual Learning for Image Recognition" [14] in 2015.
When it comes to picture categorization and segmentation, deep residual networks(ResNet) perform well. Deep residual networks function similarly to a bank of filters.utilize skip connections, or shortcuts to jump over some layers.Shown in Figure 2.12 below.
There are two primary reasons for adding skip connections: to minimize disappearing gradients and to mitigate degradation (accuracy saturation).Skipping effectively simplifies the network, using fewer layers in the initial training stage. residual connections.

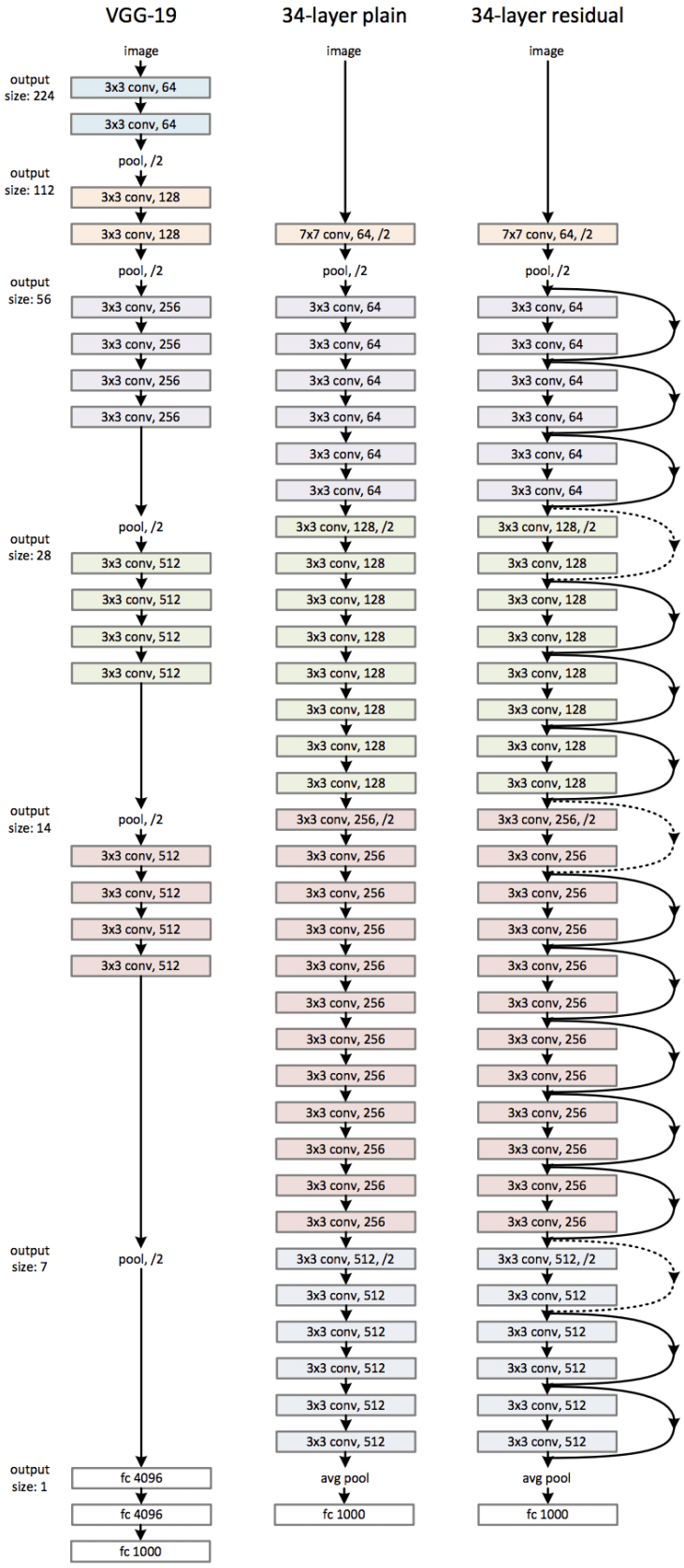


Figure 2.12: Example network architectures. Left: the VGG-19 model as a reference. Middle: a plain network with 34 parameter layers . Right: a residual network with 34 parameter layers . The dotted shortcuts increase dimensions[14].

.

2.5.4 AlexNet

AlexNet is a convolutional neural network that consists of eight layers: Five convolutional layers and three fully connected layers. designed by Alex Krizhevsky in collaboration with Ilya Sutskever and Geoffrey Hinton, who was Krizhevsky's Ph.D. advisor[15]. Two new concepts: Maxpooling and ReLU activation are introduced in this architecture [16].
The input RGB images for AlexNet is of the size $227 \times 227 \times 3$.
The convolutional layers use 11×11 filters with a stride of 4 and the maxpooling uses 3×3 filters with a stride of 2.
it was built to accommodate coloured images of size $(224 \times 224 \times 3)$. It boasted a total of over 62 million trainable parameters. Shown in Figure 2.13 below.

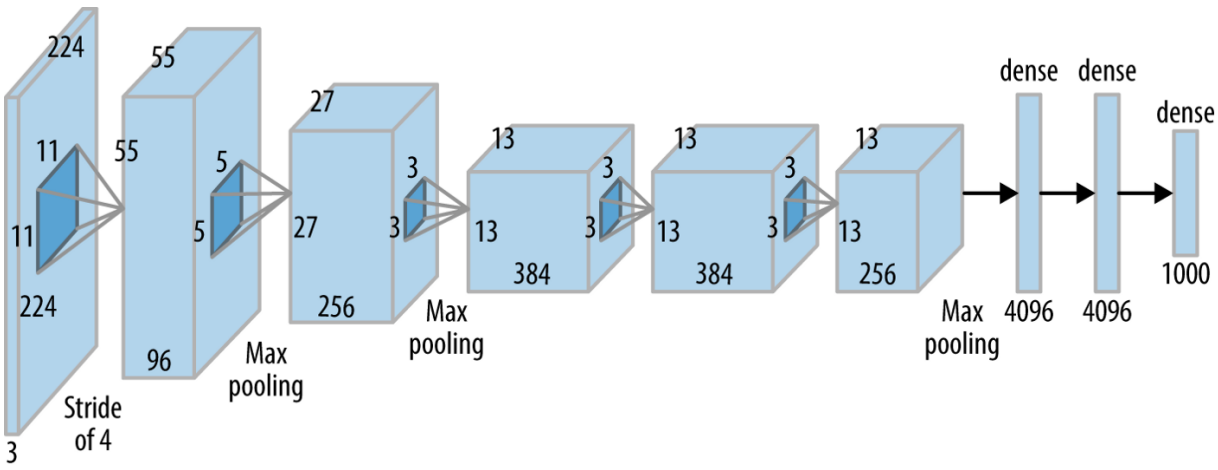


Figure 2.13: AlexNet architecture diagram [17].

2.5.5 GoogleNet

GoogLeNet is a pretrained convolutional neural network that is 22 layers deep. A pretrained version of the network can be loaded on either the ImageNet or Places365 data sets. The network trained on ImageNet classifies images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.This architecture uses 3 different size filters.(1×1, 3×3,5×5) for the same image and combines the features to get a robust output.

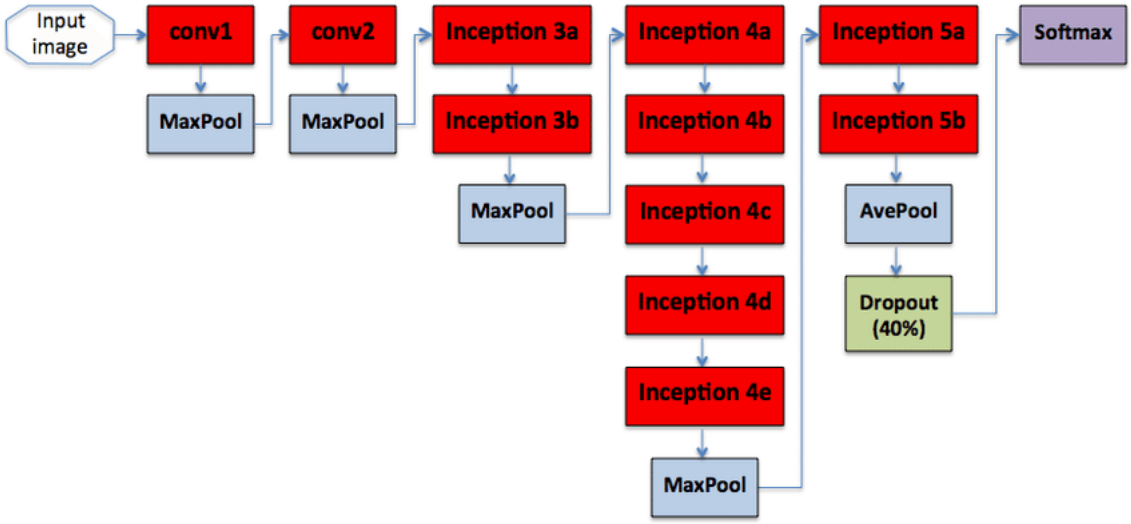


Figure 2.14: GoogleNet architecture diagram.[18]

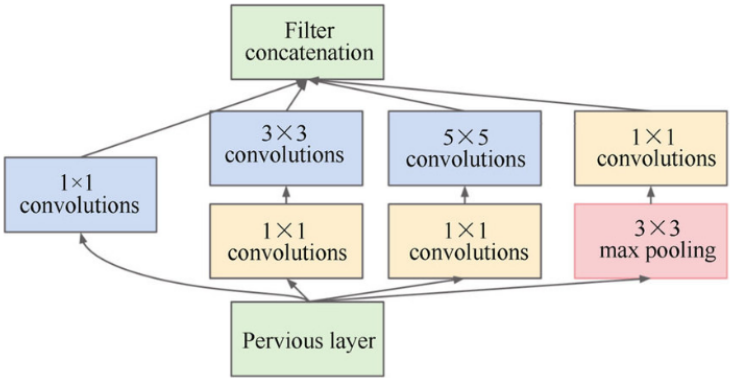


Figure 2.15: Structure of inceptions[19]

2.5.6 MobileNet

MobileNet is a lightweight architecture developed by Howard et al [20] from Google. The MobileNet model is designed for use in mobile applications, this architecture uses separable convolutions in depth. It significantly reduces the number of parameters compared to the network with convolutions regular with the same depth in the threads. This results in lightweight deep neural networks.

Howard et al introduce two simple global hyperparameters that effectively trade off latency and precision. These hyperparameters allow the model builder to choose the model of the right size for their application on the constraints of the problem.
The second version of MobileNet proposed by Sandler et al [21]. It is based on a residual structure reversed where the residual connections are between the bottleneck layers. The intermediate expansion layer uses lightweight convolutions in depth to filter features as a source of non-linearity.
MobileNet uses 3×3 depth-wise separable convolutions which uses between 8 to 9 times less computation than standard convolutions at only a small reduction in accuracy.As shown in Figure 2.16 below.

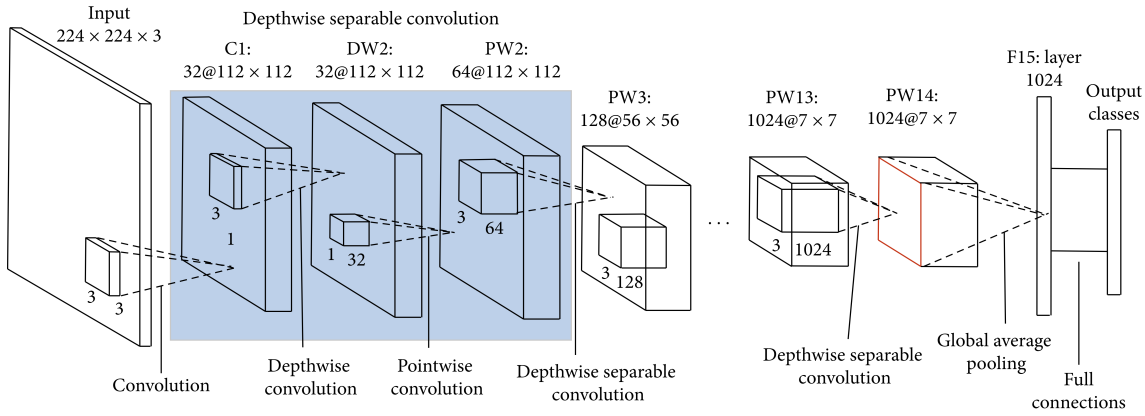


Figure 2.16: MobileNet architecture diagram.[22]

2.6 Transfer learning basics

Deep learning experts introduced transfer learning to overcome the limitations of traditional machine learning models.

2.6.1 Transfer learning

The reuse of a previously learned model on a new issue is known as transfer learning. It's particularly popular in deep learning since it can train deep neural networks with a small amount of data. This is particularly valuable in the field of data science, as most real-world situations do not require millions of labeled data points to train complicated models.

2.6.2 Definition

Transfer learning is a machine learning technique in which a model created for one job is utilized as the basis for a model on a different task.We basically reuse a pre-trained model as the starting point for a model on a new task.

2.6.3 Question to answer before transferring

What to transfer

Identifying whether parts of information could be transferred from the source model to the target model in order to better accomplish the target task and measuring the likeness between the source and target knowledge are the goals of this research.
or by asking the following question: Which aspects of knowledge may be transmitted from the source to the target in order to improve the target task's performance.

When to transfer

Because there may be circumstances and settings when transferring lowers task outcomes and performance, this stage seeks to determine whether to transfer or not. This type of transfer is known as a negative transfer. As a result, discretion is essential while deciding whether or not to transfer.

How to transfer

This stage tries to identify several approaches for transferring information from the source task to the target task, which will be explored further down.

2.7 Transfer learning Approaches

Two common approaches are as follows: develop model approach and pre-trained model approach.

2.7.1 Develop model approach

- **Selecting the source task option**
  the first step of the develop model approach is choosing a relevant predictive modeling issue with a large amount of data in which the input data, output data, and/or ideas learnt throughout the mapping from input to output data have some link.

- **Creating a source model**
  The next step is creating a skilled model for this initial assignment. To ensure that any feature learning has occurred, the model must be better than a naive model.

- **Model for reuse**
  The model fit on the source job can then be utilized to build a model for the second task of interest. Depending on the modeling approach employed, this may include utilizing all or sections of the model.

- **Adjusting the model**
  On the input-output pair data available for the job of interest, the model may need to be altered or enhanced.

2.7.2 Pre-trained model approach

- **Choosing a source model**
  From the available models, a pre-trained source model is picked. Many research institutes provide models based on vast and difficult datasets, which may be included in the pool of candidate models.

- **Model for reuse**
  The pre-trained model may then be utilized to build a model for the second job of interest. Depending on the modeling approach employed, this may include utilizing all or sections of the model.

- **Adjusting the model**
  On the input-output pair data available for the job of interest, the model may need to be altered or enhanced.

2.8 The difference between machine learning and transfer learning

The difference between machine learning and transfer learning is illustrated in the table 2.1 below.

Machine Learning	Transfer Learning
Starts from scratch which is computationally costly and necessitates a huge quantity of data in order to attain a good performance	Computationally efficient and aids in achieving better results with a small data set
Uses an isolated training technique, in which each model is trained separately for a given goal without relying on prior information	Leverages information from the pre-trained model to complete the job
Models with lower preformance	High preformance because models that employ previously trained models knowledge (features, weights, etc.) already know what the characteristics are. It's faster than building neural networks from the ground up

Table 2.1: The differences between Machine Learning and Transfer Learning.

2.9 Data processing

Computer's manipulation of data entails the translation of raw data into machine-readable form, data flow via the CPU, memory to output devices, and output formatting or modification. Data processing encompasses the use of computers to execute prescribed operations on data.
Data processing is a term used in the business sector to describe the processing of data necessary to manage organizations and enterprises.

2.9.1 Definition

The process of converting raw data into a comprehensible format is known as data preparation. We can't deal with raw data, thus this is a key stage in data mining. Before using machine learning or data mining methods, the data has to be of good quality.
The purpose of data preprocessing is to ensure that the data is of good quality.
The following criteria can be used to assess quality: Accuracy, Completeness, Consistency, Timeliness, Believability and Interpretability. Major Tasks in Data Preprocessing: Data cleaning, Data integration, Data reduction and Data transformation.Figure 2.17 describes it.



Figure 2.17: Data pre-processing cycle.

2.9.2 Data normalization

The act of structuring data in a database is known as normalization. This entails constructing tables and defining relationships between them according to rules aimed to secure data while also making the database more adaptable by removing duplication and inconsistent dependencies.As shown in Figure 2.18.

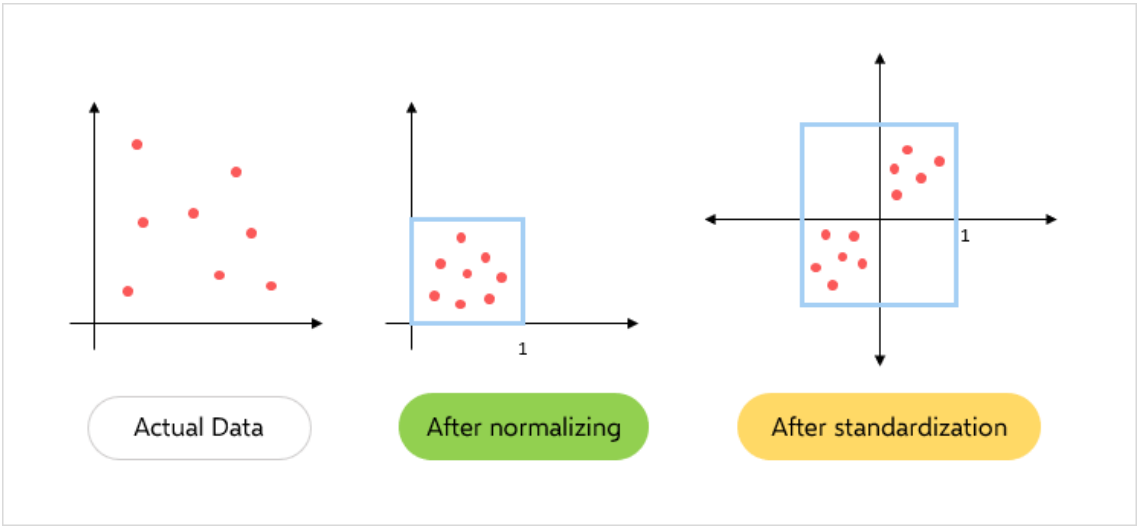


Figure 2.18: Data normalization.

2.9.3 Data augmentation

Data augmentation is a set of techniques to artificially increase the amount of data by generating new data points from existing data. This includes making small changes to data or using deep learning models to generate new data points.As shown in Figure 2.19 below. The following are some of the advantages of data augmentation:

- Improving the accuracy of model prediction.
- Reduces the cost of gathering and classifying data.

- Allowing for the prediction of unusual events.
- Prevents issues with data privacy.

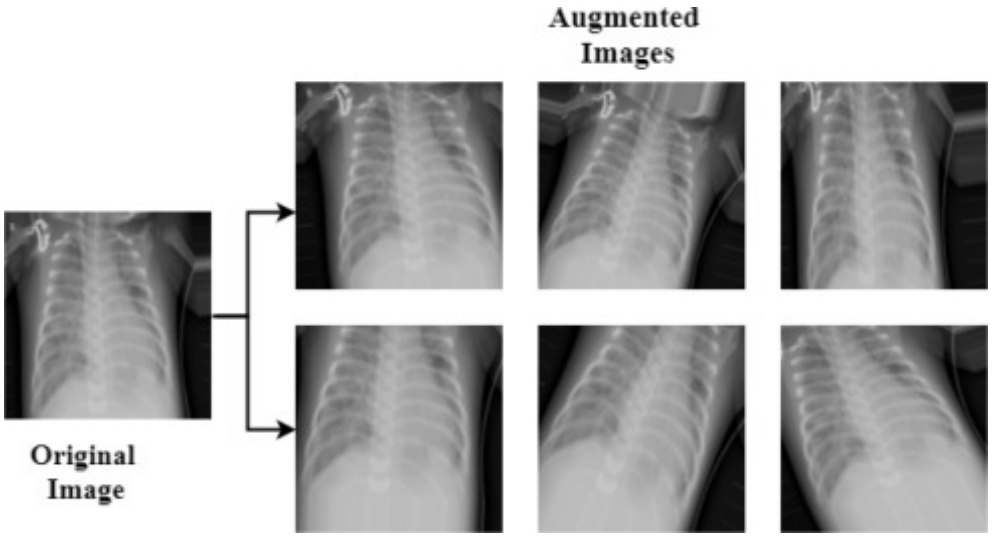


Figure 2.19: Example of Data Augmentation.

2.10 Feature extraction

Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. It yields better results than applying machine learning directly to the raw data,some traditional feature exacators are :
Independent component analysis , Autoencoder ,Resnet50 ..ect
Traditional feature extractors can be replaced with CNNs, which have a greater capacity to extract complex features.

2.11 Model evaluation

Model evaluation is the process of using different evaluation metrics to understand a machine learning model's performance, as well as its strengths and weaknesses. Model evaluation is important to assess the efficacy of a model during initial research phases, and it also plays a role in model monitoring.

2.11.1 Metrics

1. **Classification metrics**
   Classification applications rely on four main outcomes to generate this data:

   - True positive: The ground truth is positive and the predicted class is also positive.
   - False positive: The ground truth is negative and the predicted class is positive.
   - True negative: The ground truth is negative and the predicted class is negative.

- False negative: The ground truth is positive and the predicted class is negative.

2. **Accuracy**
   Accuracy is an essential metric for the evaluation of the results of DL classifiers. It is calculated as the ratio between the number of correct predictions to the total number of predictions.

$$Accuracy = \frac{True_{positive} + True_{negative}}{True_{positive} + True_{negative} + False_{positive} + False_{negative}}$$

3. **Precision**
   The precision is calculated as the ratio between the number of Positive samples correctly classified to the total number of samples classified as Positive (either correctly or incorrectly).

$$Precision = \frac{True_{positive}}{True_{positive} + False_{positive}}$$

4. **Sensitivity**
   Sensitivity(or recall) is measured as the number of accurate positive predictions divided by the sum of positive.The higher the recall, the more positive samples detected.

$$Sensitivity = \frac{True_{positive}}{True_{positive} + False_{negative}}$$

5. **Specificity**
   Specificity, also referred to as the true negative rate (TNR) is measured as the number of accurate negative prediction divided by the sum of the accurate negative and positive predictions .

$$Specificity = \frac{True_{negative}}{True_{Negative} + False_{positive}}$$

6. **F1-score**
   The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean. It is primarily used to compare the performance of two classifiers. Suppose that classifier A has a higher recall, and classifier B has higher precision.

$$F1Score = \frac{2(Precision * Recall)}{Precision + Recall}$$

7. **Regression metrics**
   Regression predictive modeling are those problems that involve predicting a numeric value. Metrics for regression involve calculating an error score to summarize the predictive skill of a model.

8. **Validation curves**
   The most used curves to understand the progress of neural networks is an accuracy curve.The gap between training and validation accuracy is a clear indication of overfitting. The larger the gap, the higher the overfitting.

9. **Learning curves**
   Learning curves are plots that show changes in learning performance over time in terms of experience.

2.12 Previous studies

As quickly as the outbreak of the pandemic , the effort to collect data about the new virus rose and many researchers attempted to automatically diagnose COVID-19 using x-ray images.
CXR image classification is not a new problem in artificial intelligence. CNNs have already reached very high performances in the diagnosis of lung diseases. And the availability of COVID-19 x-ray and CT images datasets encouraged many researchers to apply the same techniques using these data .

In [23], the authors compared three CNN models : Inception V3, Xception and ResNeXt . The database used was collected from Kaggle [24] that contains Chest X-ray scans (6432 in total) of Covid-19 cases , normal cases and pneumonia cases. that is divided into a training set and a validation set as : 1345 is normal, 490 are Covid-19, and 3632 is pneumonia for the training , and 238 samples of a normal case, 86 Covid-19, and 641 of pneumonia were considered for the validation phase. In the three models implemented the researchers used the activation LeakyReLU and used Categorical cross-entropy and the adam optimizer with a learning rate of 0.001 for the training , the results are shown on the tables below and as the they concluded Xception net had the best preformance .

Label	**Precision**	**Recall**	**F1-score**
Normal	1	0.99	1
Covid-19	1	1	1
Pneumonia	1	1	1
Accuracy			1

Table 2.2: Training results for XCeption Net.

Label	**Precision**	**Recall**	**F1-score**
Normal	0.98	0.93	0.95
Covid-19	0.99	0.92	0.95
Pneumonia	0.97	0.99	0.98
Accuracy			0.97

Table 2.3: Testing results for XCeption Net[23].

Label	**Precision**	**Recall**	**F1-score**
Normal	1	0.97	0.99
Covid-19	1	1	1
Pneumonia	0.99	1	1
Accuracy			0.99

Table 2.4: Training results for Inception Net V3.

Label	**Precision**	**Recall**	**F1-score**
Normal	0.98	0.87	0.92
Covid-19	0.96	0.95	0.96
Pneumonia	0.95	0.99	0.97
Accuracy			0.96

Table 2.5: Testing results for Inception Net V3.

Label	**Precision**	**Recall**	**F1-score**
Normal	0.99	0.98	0.98
Covid-19	1	0.90	0.95
Pneumonia	0.98	1	0.99
Accuracy			0.98

Table 2.6: Training results for ResNeXt.

Label	**Precision**	**Recall**	**F1-score**
Normal	0.91	0.99	0.90
Covid-19	0.97	0.78	0.86
Pneumonia	0.94	0.97	0.95
Accuracy			0.93

Table 2.7: Testing results for ResNeXt.

In a similar approach, The study [25] proposed a DL empowered method using pre-trained CNN models such as : AlexNet, VGG-16, GoogleNet, MobileNet-V2,SqueezeNet, ResNet-34, ResNet-50 and Inception-V3 . those models were used for a binary classification of Normal X-ray images from Covid-19 X-ray images . The data used has been collected from two sources : Covid-19 CXR images from the dataset covid-chestxray-dataset[26] and Normal CXR from a GITHub repository [27] with a splitting of 70% for training (286 total images , 143 Covid-19 and 143 Normal cases) and 30% for testing (120 images, 60 COVID-19 and 60 Normal cases). The researchers used an input layer size of (224,224,3) for all the models expect Inception-V3 (299,299,3). In pre-processing phase the data was normalized and augmented (table 2.8) .

Class	**Original dataset**		**Augmented dataset**		
	Train	**Test**	**Train**	**Validation**	**Test**
Covid-19	143	60	501	214	60
Normal	143	60	501	214	60
Total	286	120	1002	428	120

Table 2.8: Details of data splitting with and without augmentation.

The batch size was set to 32 and the model trained for 50 epochs , the optimizer used was Adam optimizer after evaluating the effectiveness of it against 3 other optimizers (SGD, Adadelta, RMSProp)(table 2.) and the learning rate was empirically decided . the evaluation metrics used were : F1-score , Specificity , Sensitivity , precision , accuracy and Area Under the ROC curve (AUc).

as the table of results below show, the ResNet-34 model achieved the highest performance . It is also noticeable that the higher sensitivities (i.e., 100%) were the two ResNet models.

Model	**Optimizer**	**Precision**	**Sensitivity**	**specificity**	**F1-score**	**Accuracy**
ResNet-34	**SGD**	95	100	95.24	0.9740	97.50
	Adadelta	95	93.44	94.92	0.9420	94.17
	RSMProp	96.67	98.31	96.72	0.9748	97.50
	Adam	96.77	100	96.67	0.9836	98.33
AexNet	**SGD**	98.33	93.65	98.25	0.9593	95.83
	Adadelta	91.67	93.22	91.80	0.9240	92.50
	RMSProp	95	98.28	95.16	0.9661	96.67
	Adam	96.72	98.33	96.67	0.9752	97.50

Table 2.9: Classification performance (in %) comparison among different optimizers.

Model	**Precision**	**Sensitivity**	**Specificity**	**F1-score**	**Accuracy**	**AUC**
ResNet-34	96.77	100	96.67	0.9836	98.33	0.9836
ResNet-50	95.24	100	95	0.9756	97.50	0.9731
GoogleNet	96.67	96.67	96.67	0.9667	96.67	0.9696
VGG-16	95.08	96.67	95	0.9587	95.83	0.9487
AlexNet	96.72	98.33	96.67	0.9752	97.50	0.9642
MobileNet-V2	98.24	93.33	98.33	0.9573	95.83	0.9506
Inception-V3	96.36	88.33	96.67	0.9217	92.50	0.9342
SqueezeNet	98.27	95	98.33	0.9661	96.67	0.9705

Table 2.10: Classification results comparison of all eight CNN models (in %).

In the paper [28], deep-learning-based approaches, namely deep feature extraction, fine-tuning of pretrained CNN, and end-to-end training of a developed CNN model, have been used in order to classify COVID-19 and normal CXR images. For deep feature extraction, pretrained deep CNN models (ResNet18, ResNet50, ResNet101, VGG16, and VGG19) were used. For classification of the deep features, the Support Vector Machines (SVM) classifier was used with various kernel functions, namely Linear, Quadratic, Cubic, and Gaussian. The aforementioned pretrained deep CNN models were also used for the finetuning procedure.
The CXR images were collected from Three different sources([29],[30],[31]):180 COVID-19 and 200 Normal . 75% of that data was selected for training and 25% for testing . The images were resized to 224×224.
For the pretrained CNN models , the SVM method was used for the purposes of classification with four kernel functions, namely Linear, Quadratic, Cubic, and Gaussian. The epsilon value of the SVMs with the Linear and Quadratic kernel functions were set to 0.04 and 0.02, respectively, whilst both the Cubic kernel and Gaussian functions epsilon value was set as 0.01 .As results show in table 2.11 ,it can be seen that the ResNet50 model produced the highest average accuracy score,it can also be observed that the ResNet50 features and Linear kernel SVM classifier produced a 94.7% accuracy score, which was the highest individual accuracy score overall.

Method : SVM	**ResNet18**	**ResNet50**	**ResNet101**	**VGG16**	**VGG19**	**Average**
Linear Kernel	86.3	94.7	88.4	89.5	88.4	89.5
Quadratic Kernel	87.4	91.6	89.5	89.5	87.4	89.1
Cubic Kernel	89.5	90.5	91.6	90.5	89.5	90.3
Guassian Kernel	86.3	93.7	88.4	89.5	87.4	89.1
Average	87.4	92.6	89.5	89.8	88.1	

Table 2.11: Achievements of pretrained deep CNN models and SVM classifiers on COVID-19 detection (Accuracy %).

Data augmentation was conducted both for fine-tuning and for end-to-end training, and was carried out by randomly rotating, shifting, and flipping the training images.Details the achievements of the fine-tuned pretrained deep CNN models on COVID-19 classification are in the table below that show that the highest accuracy score was 92.63%, produced by the ResNet50 model.

Fine-Tuning	Accuracy (%)
VGG-16	85.26
ResNet-18	88.42
ResNet-50	92.63
ResNet-101	87.37
VGG-19	89.47

Table 2.12: Achievements of fine-tuning of pretrained deep CNN models on COVID-19 classification.

In the final experiments of the current study, a novel CNN model was constructed and end-to-end trained for the purposes of COVID-19 classification that is composed of 21 layers.The network started with an input layer, followed by five convolutions layers, namely $conv_1$, $conv_2$, $conv_3$, $conv_4$, and $conv_5$, with batch normalization and ReLU layers following each convolution layer. There were also two pooling layers, $pool_1$ and $pool_2$, which followed after the $ReLu_1$ and $ReLu_2$ layers, respectively. A fully-connected layer, softmax layer, and classification layer were also used for the purposes of classification. The $conv_1$, $conv_2$, and $conv_3$ layers contained 64, 32, 16, 8, and 4 filters that were size 33 pixels. The max operator function was used in the pooling layers. The training of this CNN was carried using two different SGDM and Adam. The obtained accuracy score was 91.58% and the training procedure was completed in 11,400 iterations. the rate of correct classification of COVID-19 samples was 88.89%, whilst it was 94.0% for the normal (healthy) cases.
Additional shallow CNN models were then used to investigate their potential in terms of COVID-19 classification accuracy. The shallow networks were also end-to-end trained, the obtained results are presented in the table 2. and it can be seen that the deep CNN model (model one, with 21 layers) produced the highest accuracy score.

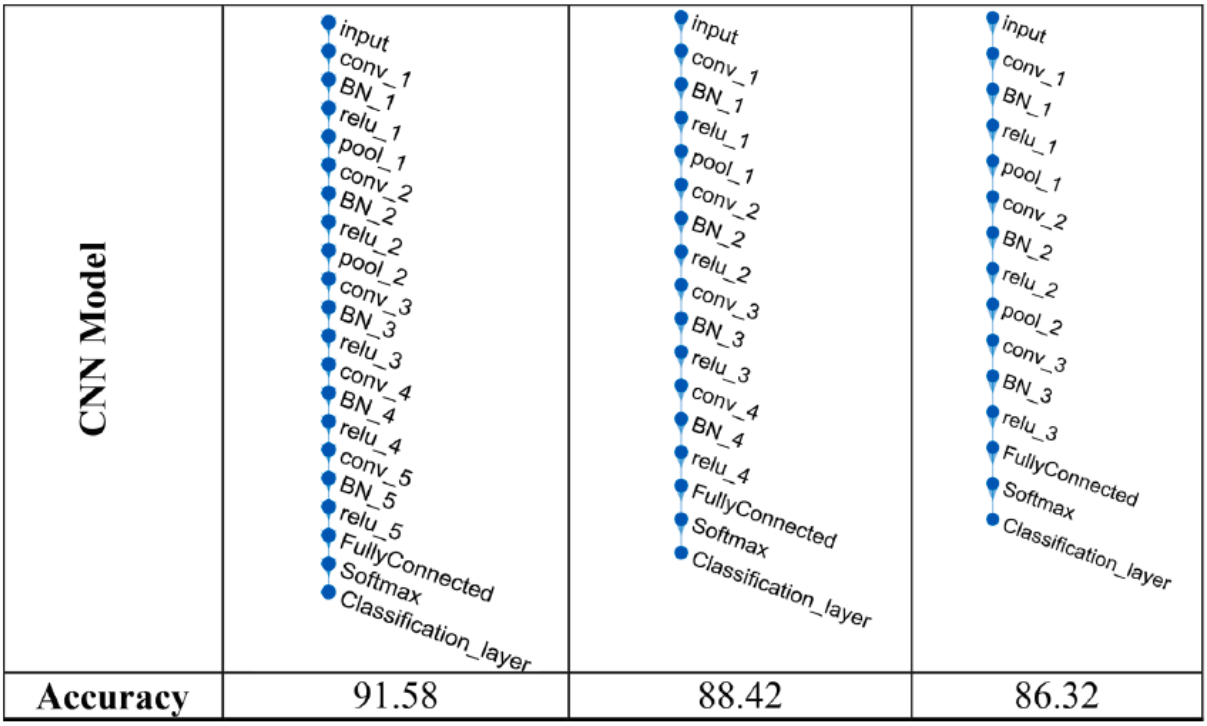


Figure 2.20: Shallow CNN models and their achievements [28].

As can be seen from these results, the deep feature extraction and deep transfer learning approach produced better accuracy scores than the end-to-end training of the proposed CNN models with the SVM classifiers.
the authors also used various local descriptors to extract features from the chest X-ray images for the purposes of COVID-19 detection. the local descriptors along with the results are summarized in the table below.

Local descriptors	**Linear SVM**	**Quadratic SVM**	**Cubic SVM**	**Gaussian SVM**	**Average Accuracy(%)**
LBP	81.1	77.9	70.5	81.1	77.1
BGP	84.2	83.2	80	83.2	82.7
BSIF	90.5	90.5	90.5	88.4	90
CENTRIST	84.2	84.2	81.1	84.2	83.4
PHOG	85.3	83.2	83.2	84.2	84
LPQ	85	84.2	82.6	85	84.2
FDLBP	87.4	87.4	87.4	87.4	87.4
QLRBP	82.1	82.1	82.1	82.1	82.1
Average Accuracy(%)	85	84.1	82.2	84.5	

Table 2.13: Results of first experiment using local feature descriptors and SVM classifiers (accuracy %).

Chapter 3

Proposed Approach

3.1 Introduction

Based on previous research using deep learning to detect Covid-19, CNN models are the most used for the detection. Accuracy and f1 score are widely used to evaluate the performance of pretrained models from scratch, we use f-1 score and accuracy besides sensitivity and precision as evaluation and validation metrics for our approaches.
In this chapter we discussed the three architectures we chose to work with in our work.

3.2 Methodology

The first step we took is downloading the dataset from the official website Kaggel where the researchers of Qatar University have compiled the COVID-QU-Ex dataset [32]. The dataset should be ready to insert into the input for the second phase.
Data preparation is one of the important procedures to do before introducing a dataset into training by cleaning it, ignoring repeated pictures, and employing other approaches that might aid in decreasing underfitting and overfitting.
Next, We resize the pictures and place them in the shape of (224x224x3), which is ideal for achieving good results owing to the enormous number of parameters that each model processes.
And after creating the models ,lastly we input the data into our three models to train, test and evaluate. We shuffle and split the data into training , testing and validation data.

3.3 System architecture

CNN, a pretrained model Residual Network (ResNet50) and a lightweight architecture MobileNetV2 are among the proposed models in the system architectures. Each model in the presented approaches has a number of layers, each of which has a number of parameters that we may tweak to achieve the optimal model for our needs.
We establish twenty two (22) layers in the initial proposed model of the CNN, five of which are convolutional layers followed by a Batch Normalization layers then a Moxpooling. And one for classification with sigmoid activation function, as well a fully connected layer .
For the second proposed model, which is based on the Residual Network (ResNet50) pretrained model with 50 layers, we used feature extraction (freezing the layers expect the last layer).
The MobileNetV2's third proposed model, a lightweight CNN model, has 53 layers, including a deep convolution layer, a one-point convolution layer, batchnorm, ReLU, average collecting layer, and softmax.
In the next sections, we'll look at why particular architectures and types of parameters were chosen, along with all the details.Figure 3.1 shows the general tructure of the process.

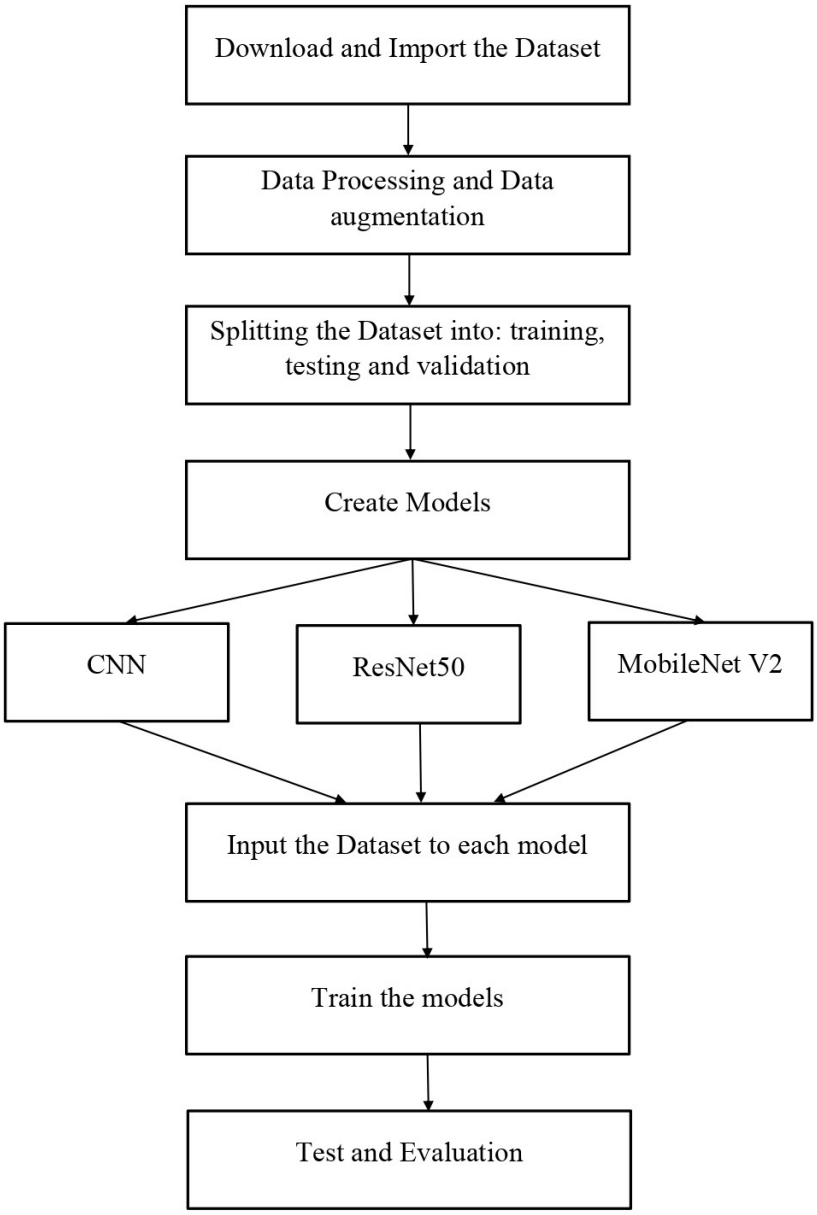


Figure 3.1: the general structure of the process.

3.4 Proposed approaches

3.4.1 Convolutional Neural Network CNN

As already stated in the previous chapter, Convolutional neural network (CNN) is a deep neural network component that contains one or more convolutional layers that automatically recognize, divide, and bind images and classifications as well as specialize in processing data with a network-like topology such as an image. It also includes Max-Pooling, AveragePooling, Dropout, Batch Normalization, and fully connected layers with activation functions.

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 1)]	0
conv2d_5 (Conv2D)	(None, 224, 224, 64)	640
batch_normalization_8 (BatchNormalization)	(None, 224, 224, 64)	256
max_pooling2d_5 (MaxPooling2D)	(None, 112, 112, 64)	0
conv2d_6 (Conv2D)	(None, 112, 112, 128)	73856
batch_normalization_9 (BatchNormalization)	(None, 112, 112, 128)	512
max_pooling2d_6 (MaxPooling2D)	(None, 56, 56, 128)	0
conv2d_7 (Conv2D)	(None, 56, 56, 256)	295168
batch_normalization_10 (BatchNormalization)	(None, 56, 56, 256)	1024
max_pooling2d_7 (MaxPooling2D)	(None, 28, 28, 256)	0
conv2d_8 (Conv2D)	(None, 28, 28, 512)	1180160
batch_normalization_11 (Bat	(None, 28, 28, 512)	2048
max_pooling2d_8 (MaxPooling2D)	(None, 14, 14, 512)	0
conv2d_9 (Conv2D)	(None, 14, 14, 512)	2359808
batch_normalization_12 (BatchNormalization)	(None, 14, 14, 512)	2048
max_pooling2d_9 (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense_2 (Dense)	(None, 64)	1605696
batch_normalization_13 (BatchNormalization)	(None, 64)	256
dropout_1 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 1)	65

Figure 3.2: CNN Model Summary.

As shown in figure 3.2 , we begin our model with an image of shape (224x224), and we employ a batch normalization before starting our first convolutional layer.
We utilized the ReLU activation function after the input layer since it is used in practically all convolutional neural networks and is the most used activation function in the world because it is half corrected. ReLU replaces negative values with zero and keeps

the same values when equal to or greater than zero, lowering the computational cost of training the network.Except for the output layer, we employ this activation function for all of the network's convolutional, maxpooling, and dense layers.

We employ a Batch Normalization after in each convoutional block and a max pooling of the size (2×2)

In the Fc Layer we use a flatten then a dense layer of a relu activation , a batch Normalization and a dropout layer to remove 0.5 of the convolutional layer's output at random. The dropout layers have values ranging from 0 to 1. (0-1). And we used (0.5) to avoid dropping a significant number of nodes that produced the desired results.
Dropout layers are used to prevent Neural Networks from overfitting .

We used The padding type same because the output size is the same as the input size. Using 'same' ensures that the filter is applied to all the elements of the input. Normally, padding is set to "same" while training the model. Output size is mathematically convenient for further computation.
The final output layer is a dense layer with a sigmoid activation and 1 units.

3.4.2 Resnet50

The Residual Network (ResNet) is a brand-new concept. It aids in the completion of complicated jobs and improves detection accuracy. ResNet aims to address the challenges of deep CNN training, like as saturation and accuracy loss. We used the ResNet50 architecture in this work. There are 50 layers of residual networks in ResNet50. The ResNet50 architecture is shown in Figure 3.5 .

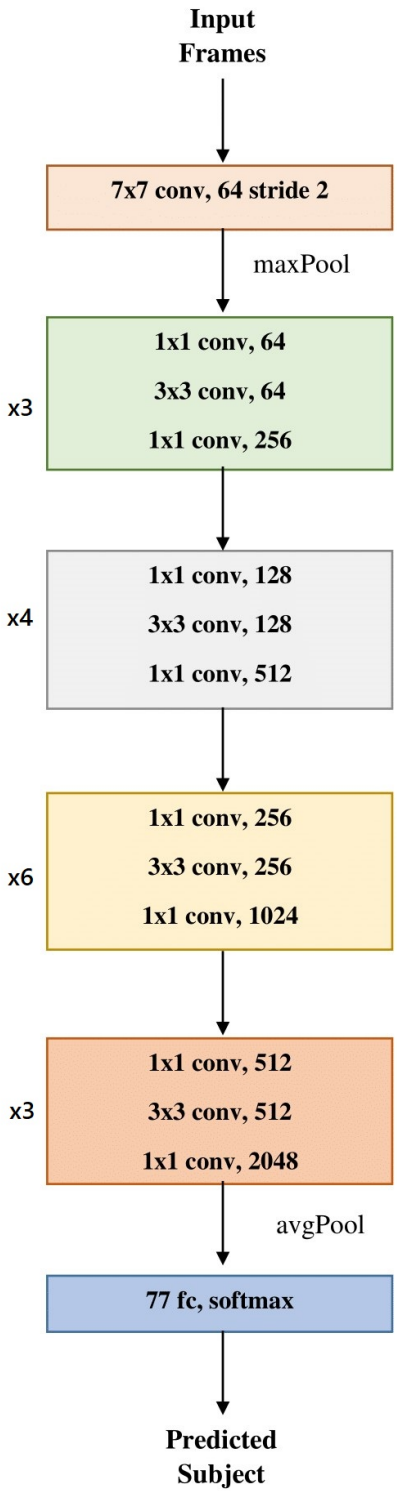


Figure 3.3: ResNet50 Architecture.

We used the pre-trained model with feature extraction by freezing the weights and then adding a flatten, a dense of 128 units and a relu activation , a dropout of 0.25 , another dense of 64 layer with a relu activation and finally an output dense layer of a sigmoid activation. Figure 3.4 shows the added the layers.

```
x=base_model.output
x = Flatten(name="flatten")(x)
x = Dense(128, activation='relu')(x)
x = Dropout(0.25)(x)
x = Dense(64, activation='relu')(x)
out=Dense(1,activation='sigmoid')(x)
```

Figure 3.4: Added layers to ResNet50.

3.4.3 MobileNetV2

As mentioned in the second chapter, Except that it employs inverted residual blocks with bottlenecking features, MobileNet Version2 is quite identical to the original MobileNet. It has a significantly fewer number of parameters V2(3.5M) than the original MobileNetV1 (4.3M). Any image size bigger than 32 × 32 is supported by MobileNets, with larger image sizes providing better performance.
The reason we chose the MobileNet V2 model is that it has a highly efficient network architecture that is easily adaptable to the needs of mobile and embedded applications [21].

we used the pre-trained model with feature extraction by freezing the weights and an output dense layer of a sigmoid activtion.

3.5 Conclusion

In this chapter we went through the proposed approach for ourwork to employ the three architectures (CNN , ResNet50 and MobileNetV2) with specific details about the models and function .
In the Next chapter we will see the employment of the models with the results given .

Chapter 4

Test and evaluation

4.1 Introduction

This chapter, we will go through everything concerning the execution of the approaches mentioned in chapter 3 . starting from the environment used for the development and the coding , the dataset chosen , the pre-processing and of course the training ,validation and testing of the implementation of our architecture . Finally, we will go through the results and the evaluation and discussion of those results.

4.2 Environment and development tools

We programmed our models using python[33] through the Google Colab [34] online plate-form and the free components provided on it.

4.2.1 Google Co-laboratory

Colaboratory, or "Colab" for short, is a is a product from Google Research which is web-based interactive environment that lets you to write and run Python code with no setup required, free access to GPUs and TPU, and easy sharing. To take advantage of Google Colab, you must have a G-mail account in order to create a Colab notebook where you can write and execute code[34].

Google Colab includes the frameworks and Python modules needed to construct, train, and evaluate machine and deep learning models. Additionally, the "pip install" command can be used to install new modules. Data and notebooks can also be imported from the local file system, Google Drive, and kaggle. Furthermore, colab notebooks and other types of data can be stored using Google cloud storage services .

Session Time

Google colab provides 12 hours of continuous execution for one session.

RAM Available Space

The available space in Random Access Memory is 12 GB, with no option to increase it.

Storage available space

In addition to Google Drive storage space, the Google colab environment allows you to use more than 30 GB of storage space for storage during a session, which implies that all python modules and files stored on this drive will be destroyed once the session is finished.

GPUs available

For execution on colab, you can use the CPU, GPU, or TPU. We picked GPUs for model training in our actual work, because they are simple to use and speed up the execution of sophisticated models such as BERT.

4.2.2 Python Modules Used

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It is easy to learn and has efficient high-level data structures and an object-oriented programming technique that is simple but effective. Python's beautiful syntax and dynamic typing, as well as its interpreted nature, make it an excellent language for scripting and quick application development across a wide range of platforms[35].

Numpy

Is the fundamental package for scientific computing in Python.In general scientific computing, linear algebra is used to reshape vectors into matrices, normalize data, randomize data partitioning, and perform other mathematical operations.

tensorflow

TensorFlow is an open source framework developed by Google researchers to run machine learning, deep learning and other statistical and predictive analytics workloads[36], we used the Version 2.0 .

Keras

Keras is one of the leading high-level neural networks APIs. It is written in Python and supports multiple back-end neural network computation engines[37].TensorFlow has adopted Keras as The high-level API for TensorFlow 2.0 .
we used it to create our models and load the pre-trained models (ResNet50 and MobileNetV2).

Matplotlib

It is used for visualization.it is a collection of command style functions that make matplotlib work like MATLAB.
e.g., creates a figure, creates a plotting area in a figure.. etc.

Sklearn

It is utilized for determining precision and recall, as well as evaluating and calculating the confusion matrix , Precision , sensitivity , specificity and F1-score.

4.3 Data used for training and test

4.3.1 Kaggel COVID-QU-Ex Dataset

We gathered the data from a well-known research group [38] at Qatar University's offered in Kaggel repository [32]. The dataset gathered consists of 6000 chest X-ray (CXR) images including:
-3000 COVID-19
-3000 Normal. 477 CXR images were also gathered and put aside for the testing phase (223 Normal images and 224 Covid-19 images) Figure 4.1 represents 6 samples from the dataset.

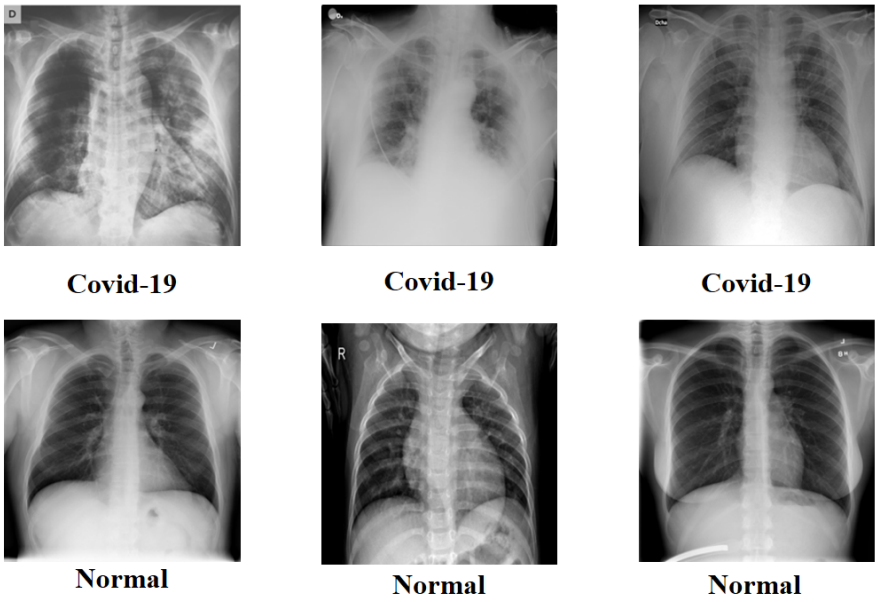


Figure 4.1: Samples from dataset.

4.4 Experimentation Steps

4.4.1 Data Preprocessing

First, we uploaded the data to Google Drive and shuffled it into two groups, COVID-19 and Normal CXR,then using a python code we imported it In Google Colab. Following that, we tackled one of the most essential tasks in data preprocessing: resizing CXR images. The images were reduced to 224 by 224 pixels, which struck a balance between giving a high enough resolution for the model to detect Covid-19 and quick training.
All images were normalized with the Keras BatchNormalization layer.

4.4.2 Splitting the data

Dataset was divided into two parts (80% training and 20% validation) for all three models (Resnet50, CNN and MobileNetV2) nonetheless. There are 6000 images (PNG) divided to two categories as follows , plus 447 images (PNG) for testing. Table 4.1 shows the data organization in the proposed models.

Categories	**Number of Training Images**	**Number of Testing Images**	**Number of Validation Images**	**Image size**
Covid-19	2400	224	600	224 x 224 pixels
Normal	2400	224	600	224 x 224 pixels
Total	4800	448	1200	

Table 4.1: organization of training, testing and validation in the Resnet, CNN,MobileNetV2 proposed models.

4.4.3 Train, test and validation

The train dataset is used to train the model. In the CNN architecture,the model learns its weights and biases while the pre-trained models (ResNet50 and MobileNetV2) learned it's weights from imageNet.

After each set of predictions, the model evaluates the validation dataset, which also aids in tuning the model's hyper-parameters.

4.4.4 Over-fitting and under-fitting

Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data.
Underfitting refers to a model that can neither model the training data nor generalize to new data.

4.5 Results and discussion

We start with a 224x224 picture, 32 shape filters (3x3), and the Adaptive Moment Estimation (Adam) optimization strategy with a learning rate of 0.001 as shown in the figure 4.2 , which gave us decent results by utilizing 20 epochs and 10 steps per epoch, which proved to be adequate to produce the best model.
Since we are training a binary classifier we chose the loss function as a binary cross-entropy.

```
print("----Training the network----")
model.compile(optimizer= Adam(0.001),
          loss='binary_crossentropy',
          metrics=['accuracy'])
#model_checkpoint
mc = ModelCheckpoint('cnn_model.h5', monitor='val_loss', mode='min', verbose=1, save_best_only=True)
r = model.fit(train_data,
              validation_data = val_data,
              epochs = epochs,
              verbose = 2,
              batch_size = 32,
              callbacks = [mc])
print("Train score:", model.evaluate(train_data))
print("Test score:", model.evaluate(val_data))
n_epochs = len(r.history['loss'])
```

Figure 4.2: the compile and fit functions for the The three models

Matrices used for result evaluation

The models are evaluated using the help of different parameters such as : accuracy, sensitivity, Precision and F1-score.

4.5.1 CNN implementation

Training the model

The model has twenty-two layers (input layer , 5 convolutional blocks , a fully connected layer and an output layert) in total, including one input layer, ten convolutional layers with a relu activation and a kernel size of (3 , 3), five MaxPooling layers with a pool size of (2 , 2), a batch Normalization in each block ,a flatten and one dense layer with a relu activation and an output dense layer with a sigmoid activation .
The figure 4.3 below shows the last 4 epochs of the training and validation.

```
Epoch 17: val_loss did not improve from 0.10422
151/151 - 28s - loss: 0.0684 - accuracy: 0.9819 - val_loss: 0.1062 - val_accuracy: 0.9550 - 28s/epoch - 184ms/step
Epoch 18/20

Epoch 18: val_loss did not improve from 0.10422
151/151 - 28s - loss: 0.0726 - accuracy: 0.9781 - val_loss: 0.1083 - val_accuracy: 0.9575 - 28s/epoch - 184ms/step
Epoch 19/20

Epoch 19: val_loss did not improve from 0.10422
151/151 - 28s - loss: 0.0644 - accuracy: 0.9850 - val_loss: 0.1063 - val_accuracy: 0.9592 - 28s/epoch - 183ms/step
Epoch 20/20

Epoch 20: val_loss improved from 0.10422 to 0.10164, saving model to cnn_model.h5
```

Figure 4.3: the last epochs of the training (CNN).

Model evaluation

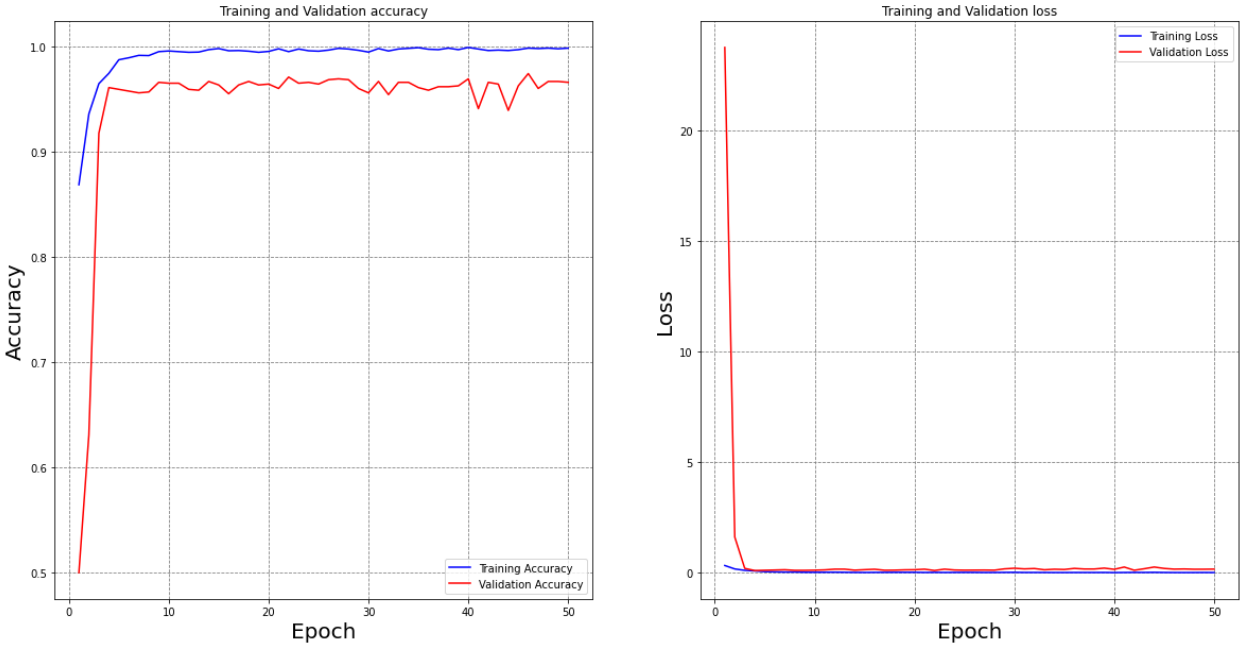


Figure 4.4: Training and validation: loss and accuracy.

In figure 4.4 the network's learning curve during training was represented in the graph. The validation and training losses both decreased as the number of iterations increased. The training and validation accuracy are shown in table 4.2.

	Validation	**Training**
Accuracy	96.58%	100%

Table 4.2: Accuracy results .

Testing the model

We moved on to the test phase after training and verifying the model, which contained 447 Covid19 imaging samples, 224 of which were diseased and the remaining 223 were not.

Using the test data we utilize the predict function with the labels and names showed in figure 4.6 to calculate the probability of each picture belonging to one of the two classes and thus creating the confusion matrix (table 4.3) and calculating the evaluation metrics mentioned earlier in this chapter (table 4.4).

```
target_names = ['Covid-19', 'Normal']
label_names = [0,1]
```

Figure 4.5: Testing labels.

	Covid19	Normal
Covid-19	219	5
Normal	13	211

Table 4.3: CNN Confusion matrix .

	Precision	sensitivity	F1-Score
Covid-19	0.94	0.97	0.95
Normal	0.97	0.94	095
Accuracy			0.96

Table 4.4: CNN Test results.

4.5.2 ResNet50 implementation

Training the model

we used and loaded the pre-trained ResNet50 consisting of 50 layers deep with weights from imagenet and freezing the layers up to the dense layer (feature extraction) . with a flatten layer, two dense layers with 128 and 64 filters respectively of a relu activation , a dropout layer of 0.25 and the final output layer as a dense layer of a sigmoid activation . The figure 4.7 below shows the last 4 epochs of the training and validation.

```
Epoch 18/20
10/10 [==============================] - ETA: 0s - loss: 0.2798 - accuracy: 0.8719
Epoch 18: val_loss did not improve from 0.14284
10/10 [==============================] - 3s 350ms/step - loss: 0.2798 - accuracy: 0.8719 - val_loss: 0.1586 - val_accuracy: 0.9469
Epoch 19/20
10/10 [==============================] - ETA: 0s - loss: 0.2614 - accuracy: 0.9000
Epoch 19: val_loss did not improve from 0.14284
10/10 [==============================] - 3s 350ms/step - loss: 0.2614 - accuracy: 0.9000 - val_loss: 0.1995 - val_accuracy: 0.9187
Epoch 20/20
10/10 [==============================] - ETA: 0s - loss: 0.3322 - accuracy: 0.8625
Epoch 20: val_loss did not improve from 0.14284
10/10 [==============================] - 3s 346ms/step - loss: 0.3322 - accuracy: 0.8625 - val_loss: 0.1752 - val_accuracy: 0.9219
```

Figure 4.6: The last epochs of the training (ResNet50).

Model evaluation

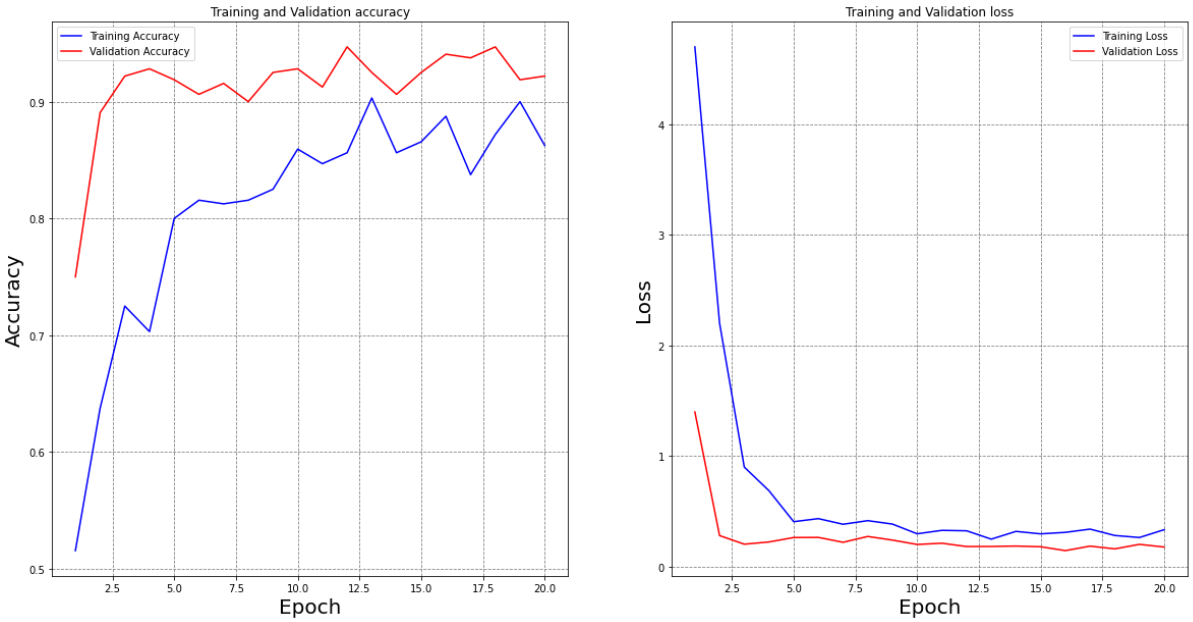


Figure 4.7: Training and validation: loss and accuracy (ResNet50).

In figure 4.8 The network's learning curve during training was represented in the graph. The validation and training losses both decreased as the number of iterations increased, but relatively a little unstable.
The training and validation accuracy are shown in table 4.5.

	Validation	**Training**
Accuracy	92.83%	90.21%

Table 4.5: Accuracy results (Resnet50).

Testing the model

Using the test data we utilize the predict function with the labels and names showed previously to calculate the probability of each picture belonging to one of the two classes and thus creating the confusion matrix (table 4.6) and calculating the evaluation metrics mentioned earlier in this chapter (table 4.7).

	Covid19	Normal
Covid-19	210	11
Normal	29	195

Table 4.6: ResNet50 Confusion matrix .

	Precision	sensitivity	F1-Score
Covid-19	0.88	0.95	0.91
Normal	0.94	0.87	0.90
Accuracy			0.91

Table 4.7: ResNet50 Test results.

4.5.3 MobileNetV2 implementation

Training the model

we used and loaded the pre-trained MobileNetV2 consisting of 53 layers deep with weights from imagenet and pooling set to average , also freezing the layers up to the dense layer (feature extraction) .the final output layer as a dense layer of a sigmoid activation . The figure 4.9 below shows the last 4 epochs of the training and validation.

```
Epoch 17: val_loss did not improve from 0.14229
10/10 - 21s - loss: 0.2701 - accuracy: 0.8844 - val_loss: 0.2146 - val_accuracy: 0.9156 - 21s/epoch - 2s/step
Epoch 18/20

Epoch 18: val_loss did not improve from 0.14229
10/10 - 22s - loss: 0.2505 - accuracy: 0.9156 - val_loss: 0.1539 - val_accuracy: 0.9375 - 22s/epoch - 2s/step
Epoch 19/20

Epoch 19: val_loss did not improve from 0.14229
10/10 - 20s - loss: 0.2488 - accuracy: 0.9031 - val_loss: 0.1887 - val_accuracy: 0.9156 - 20s/epoch - 2s/step
Epoch 20/20

Epoch 20: val_loss did not improve from 0.14229
10/10 - 22s - loss: 0.2399 - accuracy: 0.9000 - val_loss: 0.1796 - val_accuracy: 0.9281 - 22s/epoch - 2s/step
```

Figure 4.8: the training and validation: loss and accuracy (MobileNetV2).

Model evaluation

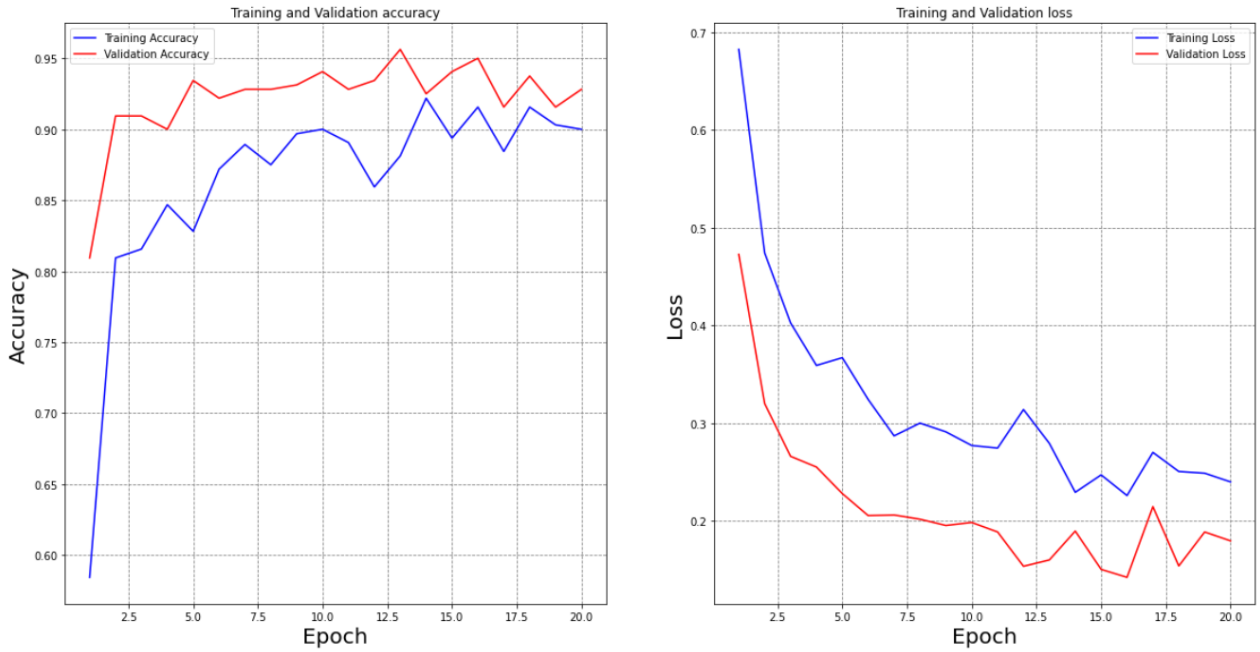


Figure 4.9: Training and validation: loss and accuracy (MobileNetV2).

In figure 4.10 The network's learning curve during training was represented in the graph. The validation and training losses both decreased as the number of iterations increased, but noticeably unstable.
The training and validation accuracy are shown in table 4.8.

	Validation	**Training**
Accuracy	94%	89.84%

Table 4.8: Accuracy results (MobileNetV2).

Testing the model

Using the test data we utilize the predict function with the labels and names showed previously to calculate the probability of each picture belonging to one of the two classes and thus creating the confusion matrix (table 4.9) and calculating the evaluation metrics mentioned earlier in this chapter (table 4.10).

	Covid19	Normal
Covid-19	200	24
Normal	11	210

Table 4.9: MobileNetV2 Confusion matrix .

	Precision	sensitivity	F1-Score
Covid-19	0.94	0.89	0.92
Normal	0.90	0.95	0.93
Accuracy			0.93

Table 4.10: MobileNetV2 Test results.

4.6 Discussion

We applied three deep CNN approaches in the detection of Covid-19 based on CXR images.Specifically, one CNN model from scratch and two pretrained transfer learning models (ResNet50 and MobileNet V2) .
The table 4.11 presents the results in all performance metrics.

	CNN	**ResNet50**	**MobileNet V2**
Accuracy	96.58%	92.83%	94%
Sensitivity	96%	91%	92%
Precision	96%	91%	92%
F1-Score	95%	90.5%	92.5%

Table 4.11: Performance comparison between the three model.

The results show that the CNN model achieved the best result with a 96.58% accuracy , 96% sensitivity, a precision of 96% and an f1-score of 95%. then comes MobileNetV2 with 94% accuracy and a difference of 1.17% than ResNet50 at 92.83% accuracy .

4.6.1 Performance comparison between CNN and transfer learning architectures

As mentioned previously, the CNN model showed superiority over ResNet50 and MobileNetV2. The primary justification for that is The two Transfer learning methods having 50 and 53 layers against our relatively small data of 6000 CXR images , the performance might be improved by increasing the epochs and data augmentation and feature normalization. therefore giving the two networks more time and data to train . Another suggested approach would be using convex nonnegative matrix factorization (CNMF) as it automatically extracts sparse and meaningful features from a set of nonnegative data vectors. Singular value decomposition (SVD) is also another technique to decompose a matrix into several component matrices, exposing many of the useful and interesting properties of the original matrix. Also, to minimize information loss , Kullback-Leibler Divergence (KL divergence)

CNN continues to prove its competence in image classification with future suggested modification as data augmentation and more pre-processing methods .

4.6.2 Performance comparison between the transfer learning architectures

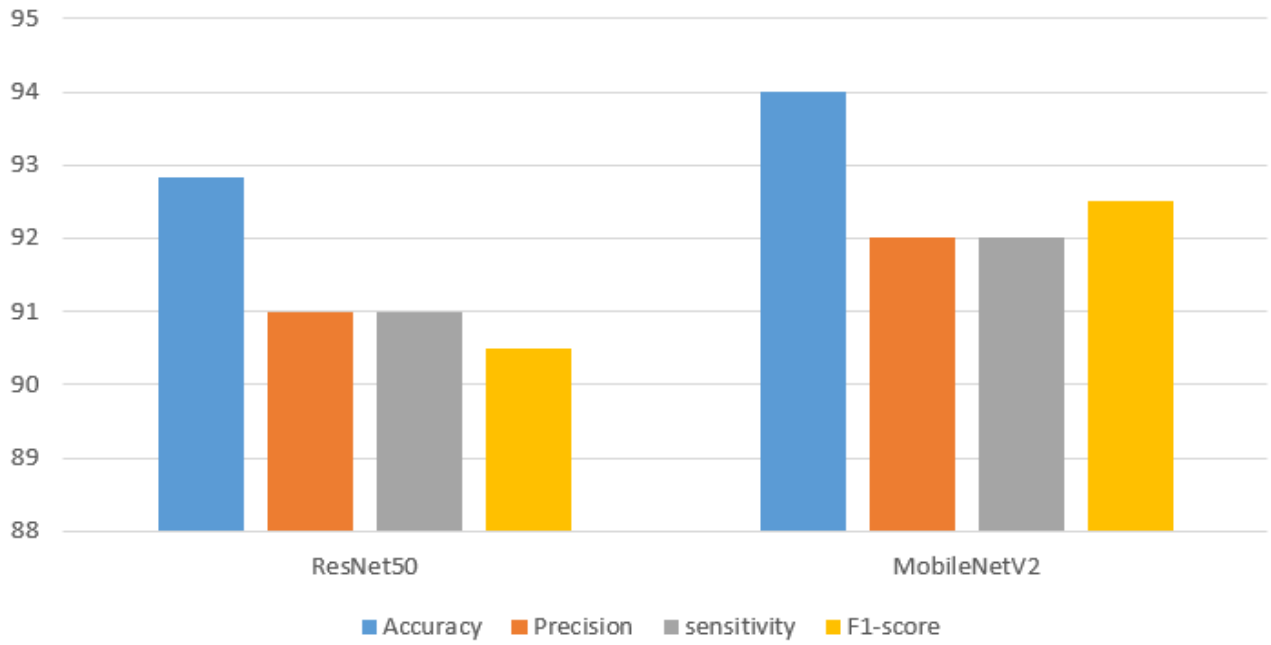


Figure 4.10: Preformance comparaison between ResNet50 and MobileNetV2.

With MobileNetv2 showing a better performance than ResNet50, it very noticable that the size of the parameters is one of the main reason . MobileNetV2 total number of parameters is 2,259,265 while ResNet50 is 36,441,217 which is a huge difference of 34181952 parameters , that's because MobileNetV2 is a lightweight network that uses depthwise separable convolutions so it significantly reduces the number of parameters when compared to ResNet50 that has a close depth (50 layers).
Figure 4.11 is the performance comparaison between ResNet50 and MobileNetV2.

4.6.3 Performance comparison with the previous studies

Preformance / Work	Accuracy	Precision	Sensitivity	F1-Score
ResNet50[25]	97.50	96.77	100	0.9836
MobileNetV2[25]	95.83	98.24	93.33	0.9573
ResNet50[28]	92.63	NM	NM	NM
CNN	96.58	96	96	95
ResNet50	92.83	91	91	90.5
MobileNetV2	94	92	92	92.5

Table 4.12: Performance comparison with the previous studies.

* NM : Not Mentioned.
It can be seen in table 4.12 that our ResNet50 model scored second in comparison to the previous work mentioned, as the ResNet50 model from the research [25] had an accuracy of 97.50% which is 4.67% higher . that can be related to the data size of [25] that significantly smaller (406 images) with a higher epoch of 50 epochs .

Our MobileNetV2 model was sightly behind by 1.83% difference of accuracy from the 95.83% accuracy of [25] , which can be fixed by a longer training time due to our database being bigger .
As our CNN model is built from scratch we can't compare it with any previous work mentioned.
Figure 4.12 shows the accuracy comparison between the previous works and our work.

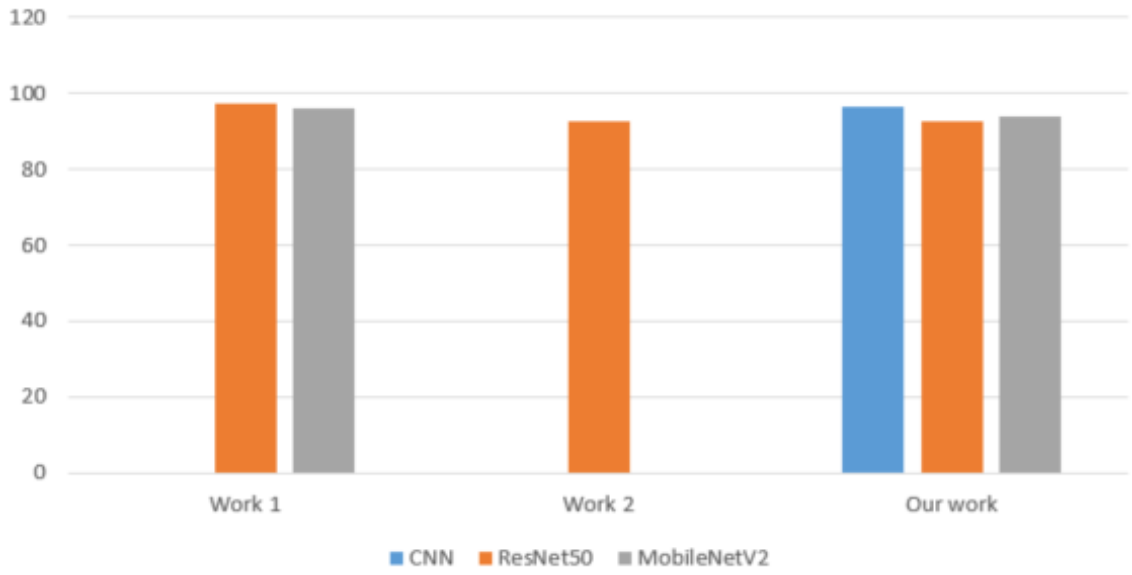


Figure 4.11: Accuracy comparison between previous work and our work.

4.7 Conclusion

This chapter went through the process of the implementation of the three proposed approaches mentioned previously (CNN, ResNet50 and MobileNetV2) and the results generated and their meaning. We touched on the software used and the dataset chosen and the modification done , as well as the training ,validation and testing process .Finally , we discussed the results in comparison with each other and with previous work and it was found that the CNN network from scratch showed better results followed by MobileNetV2 than ResNet50.

General Conclusion

In this work, We experimented with three CNNmodels: a deep learning model and two pre-trained model in an attempt to classify the Covid-19 CXR images from the normal CXR images. The motive behind our work was to avoid the mortality spikes that came from misdiagnosing Covid-19 and the latency of coming with a diagnosis in the early stages of the pandemic.
we have comprehensively evaluated the effectiveness of a simple CNN model built from scratch, a pre-trained ResNet50 model and a pre-trained MobileNetV2 model. The first model produced an accuracy of 96.58%, the second model gave an accuracy of 92.83% while the third model accuracy's 94%. Which showed that the CNN is the best fit compared to the other two models.
The finding of this work proves the effectiveness of CNN for image classification and how the concept of dimensionality reduction suits the huge number of parameters in an image.

This work also paves the way for further development of effective deep and transfer learning models for a more accurate diagnosis of COVID-19 infection in the future. We plan to improve our models by utilizing data augmentation and more pre-processing methods for the images for more reliable models.
This work is developed for binary classification and as there has not been many multi-class classifiers , in future work we intent to verify the effectiveness of the models for a multi-class classification problem.

Finally , we aim to create a simple platform for doctors and medical field personnel to diagnose patients. The automated version of their manual task will make their job easier and faster, allowing the disease to be controlled much more quickly.

Bibliography

[1] University of TARTU. *Sampling and quantization — Digital Image Processing*. URL: https://sisu.ut.ee/imageprocessing/book/2.

[2] DINFOS Pavilion. *Vector vs. Raster Images: Choosing the Right Format*. URL: https://pavilion.dinfos.edu/Article/Article/2223089/vector-vs-raster-images-choosing-the-right-format/.

[3] Research Guides at University of Michigan Library. *Image File Formats*. URL: https://guides.lib.umich.edu/c.php?g=282942&p=1885348.

[4] ShutterStock. *PPI vs. DPI: Demystifying the World of Online and Print Resolution*. URL: https://www.shutterstock.com/blog/ppi-vs-dpi-resolution-guide.

[5] Swarnendu Ghosh et al. "Understanding Deep Learning Techniques for Image Segmentation". In: (July 2019). URL: http://arxiv.org/abs/1907.06119.

[6] Luo Gang, Opas Chutatape, and Shankar M. Krishnan. "Detection and measurement of retinal vessels in fundus images using amplitude modified second-order Gaussian filter". In: *IEEE Transactions on Biomedical Engineering* 49 (2 2002), pp. 168–172. ISSN: 00189294. DOI: 10.1109/10.979356.

[7] Junbo Zhao et al. "Power System Dynamic State Estimation: Motivations, Definitions, Methodologies, and Future Work". In: *IEEE Transactions on Power Systems* 34 (4 July 2019), pp. 3188–3198. ISSN: 15580679. DOI: 10.1109/TPWRS.2019.2894769.

[8] World Health Organization. *Coronavirus*. URL: https://www.who.int/health-topics/coronavirus#tab=tab_1.

[9] Centers for Disease Control and Prevention. *Symptoms of COVID-19*. URL: https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html.

[10] Franks Teri J et al. "Lung pathology of severe acute respiratory syndrome (SARS): a study of 8 autopsy cases from Singapore". In: *Human pathology* 34.8 (2003), pp. 743–748.

[11] John D. Kelleher. *Deep Learning*. The MIT Press, 2019, p. 283. ISBN: 9780262537551.

[12] IBM education cloud. *What are Convolutional Neural Networks?* URL: https://www.ibm.com/cloud/learn/convolutional-neural-networks.

[13] Sepp Hochreiter and Jurgen Schmidhuber. "LONG SHORT-TERM MEMORY". In: *Neural Computation* (1997).

[14] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, 1097–1105.

[16] Daniel Yapi et al. "A Learning-Based Approach for Automatic Defect Detection in Textile Images". In: *IFAC-PapersOnLine* 48 (3 Jan. 2015), pp. 2423–2428. ISSN: 2405-8963. DOI: 10.1016/J.IFACOL.2015.06.451.

[17] Aqueel Anwar and Towards data science. *Difference between AlexNet, VGGNet, ResNet, and Inception*. 2019. URL: https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96.

[18] Marco Castelluccio et al. "Land Use Classification in Remote Sensing Images by Convolutional Neural Networks". In: (Aug. 2015). URL: http://arxiv.org/abs/1508.00092.

[19] Christian Szegedy et al. "Going deeper with convolutions". In: ().

[20] Andrew G. Howard et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". In: (Apr. 2017). URL: http://arxiv.org/abs/1704.04861.

[21] Mark Sandler et al. *MobileNetV2: Inverted Residuals and Linear Bottlenecks*.

[22] Wei Wang et al. "A novel image classification approach via dense-mobilenet models". In: *Mobile Information Systems* 2020 (2020). ISSN: 1875905X. DOI: 10.1155/2020/7602384.

[23] Rachna Jain et al. "Deep learning based detection and analysis of COVID-19 on chest X-ray images". In: *Applied Intelligence* 51.3 (2020), 1690–1700. DOI: 10.1007/s10489-020-01902-1.

[24] Prashant Patel. *Chest X-ray (Covid-19 Pneumonia)*. URL: https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia.

[25] Soumya Ranjan Nayak et al. "Application of deep learning techniques for detection of COVID-19 cases using chest X-ray images: A comprehensive study". In: *Biomedical Signal Processing and Control* 64 (2021), p. 102365. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2020.102365. URL: https://www.sciencedirect.com/science/article/pii/S1746809420304717.

[26] J.P. Cohen, P. Morrison, and L. Dao. *COVID-19 image data collection*. 2020. URL: https://github.com/ieee8023/covid-chestxray-dataset.

[27] X. Wang et al. *hestx-ray8: Hospitalscale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017. DOI: pp.20972106.

[28] Aras M. Ismael and Abdulkadir Şengür. "Deep learning approaches for COVID-19 detection based on chest X-ray images". In: *Expert Systems with Applications* 164 (2021), p. 114054. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2020.114054. URL: https://www.sciencedirect.com/science/article/pii/S0957417420308198.

[29] GitHub. *COVID-19 [dataset]*. Retrieved March 10, 2020. 2020. URL: https://github.com/ieee8023/covid-chestxray-dataset/tree/master/images.

[30] Kaggle. *Covid-19 X-ray chest and CT [dataset].* Retrieved April 20, 2020. 2020. URL: https://www.kaggle.com/bachrr/covid-chest-xray.

[31] Kaggle. *X-ray chest [dataset].* Retrieved March 10, 2020. 2020. URL: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia.

[32] *QaTa-COV19 Dataset — Kaggle.* URL: https://www.kaggle.com/datasets/aysendegerli/qatacov19-dataset.

[33] *About Python™ — Python.org.* URL: https://www.python.org/about/.

[34] *Google Colab.* URL: https://research.google.com/colaboratory/faq.html.

[35] Python.org. *What is Python? Executive Summary.* URL: https://www.python.org/doc/essays/blurb/.

[36] *TensorFlow.* URL: https://www.tensorflow.org/.

[37] *Keras: the Python deep learning API.* URL: https://keras.io/.

[38] Anas M. Tahir and Muhammad E. H. Chowdhury and Yazan Qiblawey and Amith Khandakar and Tawsifur Rahman and Serkan Kiranyaz and Uzair Khurshid and Nabil Ibtehaz and Sakib Mahmud and Maymouna Ezeddin. *COVID-QU-Ex Dataset.* 2022. DOI: {10.34740/KAGGLE/DSV/3122958}. URL: {https://www.kaggle.com/dsv/3122958}.