PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
University of Mohamed Khider – BISKRA
Faculty of Exact Sciences, Science of Nature and Life
Computer Science Department

Order N°: GLSD04/M2/2021

Thesis

Presented to obtain the academic master's degree in

Computer science

Option : **GLSD**

-

A Deep learning approach for children pneumonia early diagnosis

By :

REMAIGUI BRAHIM

Defended on 27/june/2020, in front of the jury composed of::

Okba Tibermacine	MCA	President
KAHLOUL Laid	Professor	Supervisor
Djaber Khaled	MAA	Examiner

Session 2021-2022

Acknowledgements

First and Foremost, I want to express gratitude, praise and thanks to allah Almighty, who helped me and enabled me to reach this day.

Secondly, I want to send a great thank you to my parents, who had great credit for my success, May God bless you and prolong your lives.

Thirdly, I'd want to express my gratitude to Professor KAHLOUL Laid, my instructor and supervisor, for guiding me in the proper manner during this study.

Then I would like to thank all computer science department teachers,

Finaly I want to thank everyone who helped me, especially Iskandar, Ayman and Ghouzlan ,and also my colleagues in the department and my colleagues at work (Informatic servicein in ELBARKA company) and all family members and friends

REMAIGUI BRAHIM

Abstract

Pneumonia is a bacterial or viral illness that causes inflammation in the lungs, it strongly affects children. Early diagnosis is an important factor in terms of the successful treatment process. Generally, a professional radiologist can diagnose the condition by checking out the chest X-ray pictures. Nevertheless, it might be ambiguous or mistaken with other conditions as well as it can make the diagnosis subjective. Therefore, computer-aided diagnosis systems are needed to guide the clinicians.

In this master project, we have utilized the convolutional neural network (CNN) technique to build a model that can detect the pneumonia for children. We used this model because CNN-based deep learning classification algorithms can automatically extract high-level representations. The model was trained, validated, and tested using a public data set, and the obtained results are considered satisfactory since they demonstrated an improvement compared to the existing works which have used the same data set.

.

Keywords: *—Deep learning, neural convolutional networks, early diagnosis, pneumonia, smart healthcare.*

Résumé

La pneumonie est une maladie bactérienne ou virale qui provoque une inflammation des poumons, elle affecte fortement les enfants. Un diagnostic précoce est un facteur important pour la réussite du processus de traitement. Généralement, un radiologue professionnel peut diagnostiquer la condition en vérifiant les images radiographiques du thorax. Néanmoins, il peut être ambigu ou confondu avec d'autres conditions et rendre le diagnostic subjectif. Par conséquent, des systèmes de diagnostic assistés par ordinateur sont nécessaires pour guider les cliniciens.

Dans notre projet de master, nous avons utilisé la technique des réseaux de neurones convolutifs (CNN) pour construire un modèle capable de détecter la pneumonie chez les enfants. Nous avons utilisé ce modèle car les algorithmes de classification à base d'apprentissage en profondeur, basés sur CNN, peuvent extraire automatiquement des représentations de haut niveau. Le modèle a été entraîné, validé et testé à l'aide d'un dataset public. Les résultats obtenus sont jugés satisfaisants car ils indiquent une amélioration par rapport aux travaux existants qui ont utilisé le même dataset. .

Keywords: —*Apprentissage profond, réseaux de neurones convolutifs, diagnostic précoce, pneumonie, soins de santé intelligents.*

Contents

List of Figures

List of source code

List of Tables

Chapter 1

General Introduction

Pneumonia is a swelling (inflammation) of the tissue in one or both lungs. The air sacs get filled with fluid or pus (purulent material). It is usually caused by a bacterial infection, but it can also be caused by a virus, such as coronavirus (COVID-19). Pneumonia is usually associated with the elderly, nevertheless, it is considered the biggest infectious cause of children's death worldwide. Every year, the [13] records more than 800,000 children less than five years old deaths, including over 153,000 newborns caused by pneumonia. Statistically, two children per minute die from pneumonia and almost all of these deaths are preventable.

Artificial intelligence has achieved great success recently in the field of health-care. Several researchers invest in machine learning (ML), deep learning (DL) and other AI techniques to provide innovative solutions for medicine applications. Indeed, using DL demonstrated remarkable successes recently in several diagnosis and prediction applications; the authors of [50, 37, 35, 51, 38] developed DL based approach for early breast cancer detection, the authors of [41, 49, 19, 46] used DL for the early detection of diabetes based on retinopathy images, the authors of [18, 28, 47] worked on brain tumours for adults and children, authors of [30, 26, 27]) developed DL based approaches to estimate the age of children based on their bones X-rays images, etc.

Concerning pneumonia disease, an early diagnosis is an important factor in terms of the successful treatment process. Generally, the disease can be diagnosed from chest X-ray images by an expert radiologist. The diagnoses can be subjective for multiple reasons such as the appearance of disease which can be unclear in chest X-ray images or can be confused with other diseases. Therefore, computer-aided diagnosis systems are needed to guide the clinicians. As a

matter of fact, Deep learning convolutional neural networks have been proven to be very effective in many areas such as image recognition. Using X-ray images in a DL approach for pneumonia diagnosis and classification attracted currently several researchers due to the COVID-19 pandemic [22, 31, 36, 21], etc. Thus, in this work we are interested in proposing a deep-learning architecture for children pneumonia detection based on X-ray images. We aim to improve previous obtained results in some existing researches which have been experimented on the same dataset. Results of this work were submitted, accepted and presented as an oral presentation in [32].

This manuscript is structured as follows:

- **Chapter 2: Background of the Work.** This chapter will first begin by providing an explanation about the lungs, it's components and the pneumonia. After that it will discuss health care, and how Artificial intelligence impacted healthcare field. it will also explain artificial intelligence, machine learning and deep learning, finally this chapter will mention some previous related works.

- **Chapter 3: Design of a deep learning architecture for pneumonia detection in pediatric.** This chapter will describe the dataset used in our work, and the system design phases

- **Chapter 4: Implementation and results.** This chapter will discuss how to create the deep learning system for pneumonia diagnosis, as well as our architecture and tools utilized in this project. the results obtained are also provided.

- **Chapter 5: Conclusion and Perspectives.** We will synthesize and assess our thoughts and findings, by offering suggestions for improvement and provide some viewpoints.

Chapter 2

Background of the work

2.1 Introduction

Artificial intelligence (AI) is a broad discipline of computer science that focuses on creating intelligent computers that can accomplish jobs which would normally need human intellect. In this topic, we find machine learning (ML) and deep learning (DL), which have demonstrated significant success in a variety of fields, most notably health care, where ML and DL can evaluate medical data to give better service to patients. This chapter will first talk about health care (its stages and usage of artificial intelligence in it), then it will talk about the lung and its components as well as the pneumonia, after that we will present artificial intelligence, machine learning and deep learning, and finally we will mention some previous related works.

2.2 Pneumonia

2.2.1 The lungs

The lungs (Figure 2.4) are the organs of respiration. They are located in the thorax, either side of the mediastinum. The function of the lungs is to oxygenate blood. They achieve this by bringing inspired air into close contact with oxygen-poor blood in the pulmonary capillaries [10].

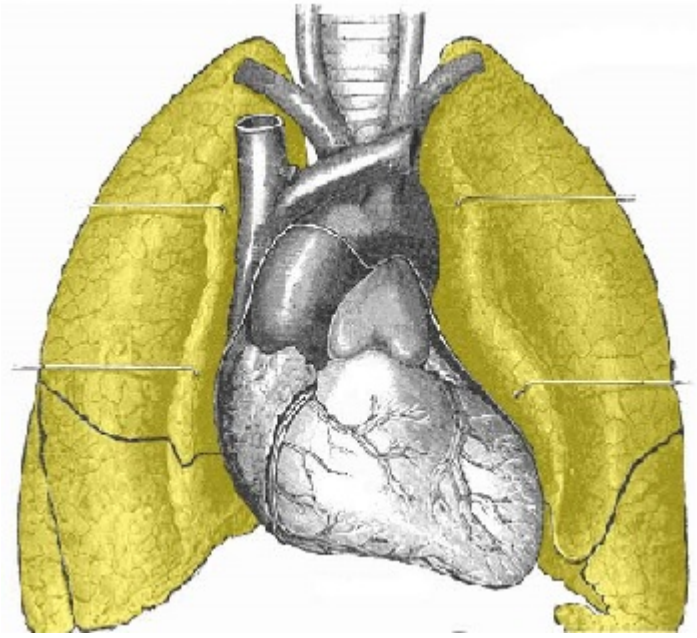


Figure 2.1: Anatomical position of the lungs.

2.2.1.1 Lung Structure

The lungs are roughly cone shaped, having three surfaces and three borders, an apex, a base, and three surfaces. Due to the presence of the heart, the left lung is somewhat smaller than the right (Figure 2.2). Each lung consists of [10]:

Apex:

The lung's blunt superior end. It protrudes over the level of the first rib and into the neck's floor.

Base:

The inferior surface of the lung, which sits on the diaphragm

Lobes:

(two or three) - These are separated within the lung by fissures.

Surfaces:

(three) - These are the areas of the thorax that they are facing. Costal, mediastinal, and diaphragmatic are their names.

Borders:

(three) – The edges of the lungs, named the anterior, inferior and posterior borders. ...

- **Lobes:** The lobular structure of the right and left lungs differs. There are three lobes in the right lung: superior, medium, and inferior. There are two fissures that separate the lobes:

  Oblique fissure:

  It runs from the inferior lung border to the posterior lung border in a superoposterior orientation.

  Horizontal fissure:

  It runs horizontally from the sternum to the oblique fissure at the level of the 4th rib.

  The left lung contains superior and inferior lobes, which are separated by a similar oblique fissure. There are three lung surfaces, each corresponding to an area of the thorax.

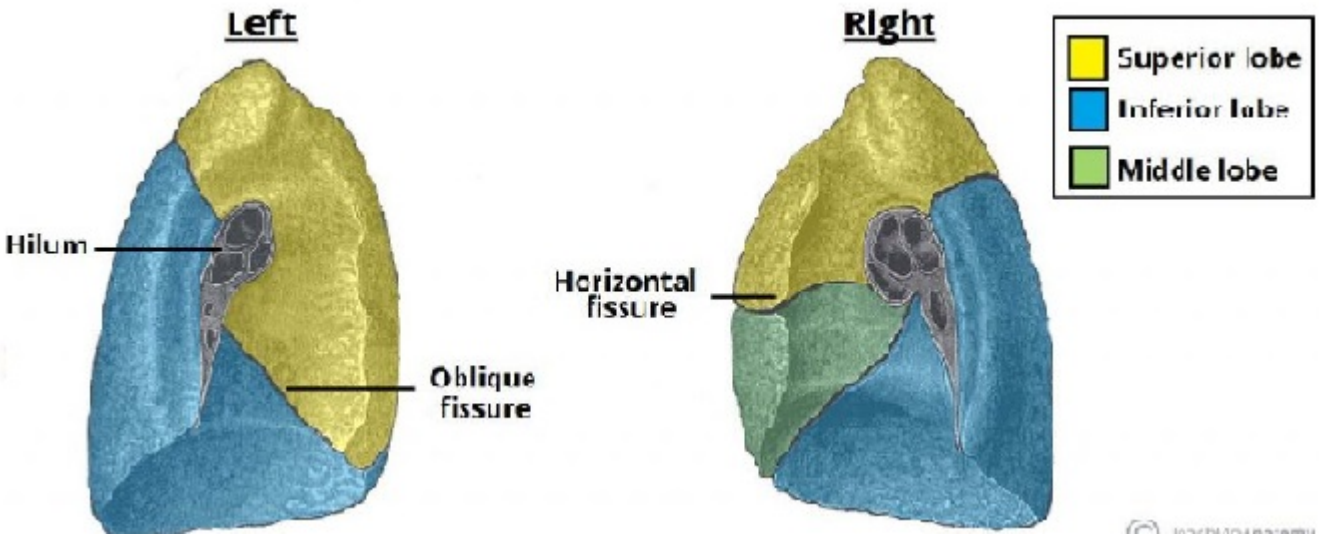


Figure 2.2: The lobes and fissures of the lungs. The oblique fissures are similar in both lungs.

- **surface:** The lateral aspect of the middle mediastinum is faced by the mediastinal surface of the lung. On this surface lies the lung hilum (the point at which structures enter and exit the lung). The diaphragmatic surface forms the foundation of the lung. It has a concave form and sits on the diaphragm's dome. Due to the elevated elevation of the right dome above the liver, this concavity is deeper in the right lung. The costal area is convex and

smooth. It is positioned against the chest wall's interior surface. The costal pleura, which separates it from the ribs and innermost intercostal muscles, is connected to it.

- **Borders:** The intersection of the mediastinal and costal surfaces forms the lung's anterior boundary. A deep notch, produced by the apex of the heart, marks the anterior boundary of the left lung. The cardiac notch is what it's called. The inferior border divides the costal and mediastinal surfaces from the base of the lung. The back border is rounded and smooth (in contrast to the anterior and inferior borders, which are sharp). It is produced by the posterior junction of the costal and mediastinal surfaces.

2.2.2 Pneumonia disease

Pneumonia (Figure 2.3) [17] is a frequent acute respiratory infection that affects the alveoli and distal air-ways; it is a serious public health issue that is linked to high morbidity and short- and long-term mortality in people of all ages all over the world. Community acquired pneumonia and hospital-acquired pneumonia are the two types of pneumonia. Pneumonia can be caused by a wide range of microorganisms, including bacteria, respiratory viruses, and fungus, and their prevalence varies greatly across the country. Pneumonia is more common in susceptible peo-ple, such as children under the age of five and elderly persons who have had previous chronic illnesses. The pathogen's features are less important than the host's immunological response in the development of the disease. Pneumonia patients frequently have respiratory and systemic symptoms, and the diagnosis is based on both clinical and radio logical findings. It is critical to identify the microorganisms that are causing the problem, as delayed or ineffective antibi-otic therapy might result in poor outcomes. The management of pneumonia will be improved by new antibiotic and non-antibiotic medications, as well as speedy and accurate diagnostic technologies that can detect infections and drug resistance.

Pneumonia kills more children than any other infectious disease [13], taking the lives of over 800,000 children under five every year, or around 2,200 every day. This includes over 153,000 newborns.

Figure 2.3: Pneumonia disease.

2.3 presentation of X-ray images

2.3.1 What is a chest X-ray?

A chest X-ray is a diagnostic imaging procedure that allows doctors to see the whole respiratory system, including the lungs, trachea, bronchi, and pleura. The heart, as well as the vertebrae and ribs, they are all examined by using a chest X-ray. X-rays are used to take photographs of the body component being studied in this [14].

2.3.2 Functioning of X-ray images

X-ray machines are essentially high-tech cameras. Instead of visible light, X-rays are used to expose film and make pictures. These rays are absorbed by bone tissues, which contain a lot of calcium, but they pass through soft tissues like skin and fat. As a result, the darker portions on an X-ray correspond to less compact tissues, whereas the brighter areas reflect denser or thicker tissues. As a result, metal items seem to be white.

2.3.3 Uses of Chest x-rays

A thoracoscopic radiological test is non-invasive, which means radiological examination that creates a picture of the intestine and internal organs. To perform a thoracoscopic test using faisceau X (Figure 2.4), the thorax is quickly exposed to the X-rays of an X-ray device, and a

picture is captured on film or on a computer. Physicians request thoracoscopic radiography tests for a variety of reasons. This radiological baseline test can be used to evaluate a variety of clinical problems. On a thoracoscopic radiograph, one of the conditions of base that can be seen is:

- **Pneumonia**
- **Hypertrophy of the heart**
- **Lung mass**
- **Rib cracks**
- **Fluid around the lung (pleural irradiation)**
- **Air around the lung (pneumothorax)**
- **Covid-19**
- **Pulmonary opacity**
- **Tuberculosis**

Overall, a chest X-ray exam is a simple, rapid, low-cost, and relatively safe procedure with low radiation risk. It's also really easy to find.

2.4 Machine learning

Machine learning is a branch of artificial intelligence (AI) (See Figure 2.5) that allows computers to learn and develop on their own without having to be explicitly programmed. Machine learning is concerned with the creation of computer programs that can access data and learn on their own. The learning process starts with observations or data, such as examples, direct experience, or instruction, so that we may seek for patterns in data and make better judgments in the future based on the examples we offer. The fundamental goal is for computers to learn on their own, without the need for human involvement, and to change their behavior accordingly [23].

Figure 2.4: Chest x-ray.

Machine learning models are fed training data and are tweaked to make accurate predictions for that data. The models' major objective is to be able to generalize their learnt knowledge and make accurate predictions for new, previously unknown data. The generalization capacity of a model is often measured during training using a different dataset called the validation set, which is then utilized as feedback for further model tweaking. The finished model is tested on a test set, which is intended to simulate how the model would perform when presented with new, unknown data, after numerous cycles of training and tweaking [39].

2.4.1 Machine learning methods

2.4.1.1 Supervised machine learning

The use of labeled datasets to train algorithms that reliably categorize data or predict outcomes is characterized as supervised learning, often known as supervised machine learning.As more data is introduced into the model, the weights are adjusted until the model is well fitted. This happens throughout the cross validation process to verify that the model does not overfit or underfit. Organizations may use supervised learning to tackle a range of real-world issues at scale, such as spam classification in a distinct folder from your email. Neural networks, nave bayes, linear regression, logistic regression, random forest, support vector machine (SVM), and

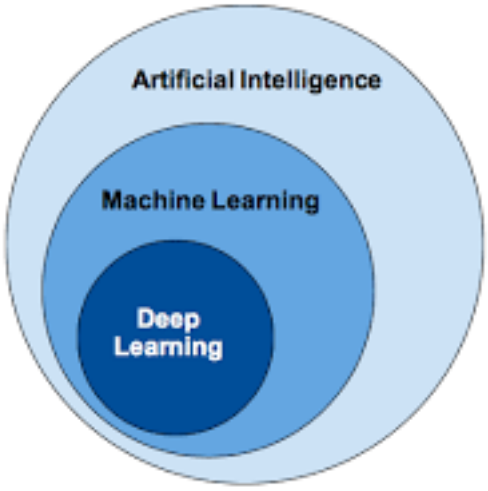


Figure 2.5: Relationship between AI, ML and DL.

other approaches are used in supervised learning (see Figure 2.6).

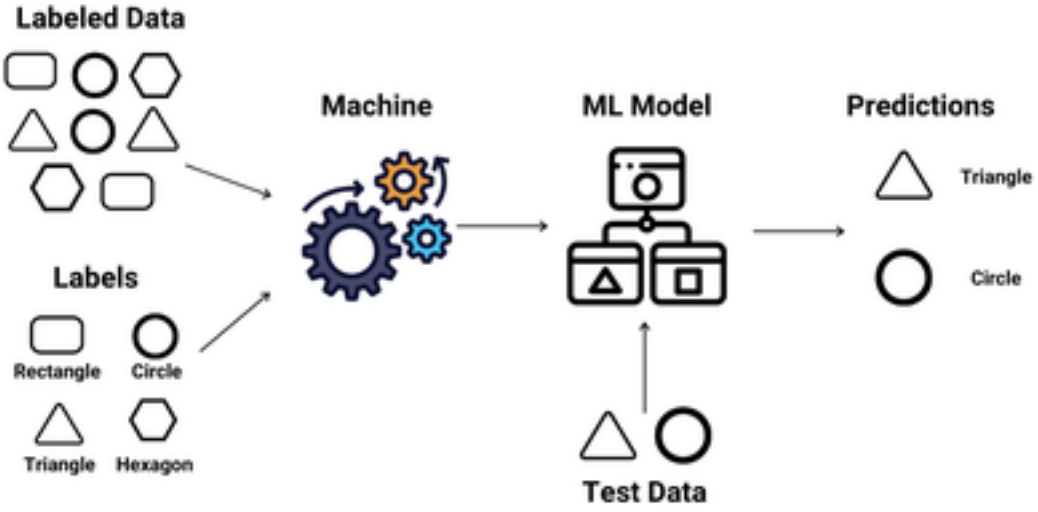


Figure 2.6: Supervised learning Example.

2.4.1.2 Unsupervised machine learning

Unsupervised learning Figure 2.7, also known as unsupervised machine learning, analyzes and clusters unlabeled information using machine learning techniques. Without the need for human interaction, these algorithms uncover hidden patterns or data groupings. Because of its capacity to find similarities and contrasts in data, it's perfect for exploratory data analysis, cross-

selling techniques, consumer segmentation, picture and pattern recognition. Principal component analysis (PCA) and singular value decomposition (SVD) are two typical methodologies for reducing the number of features in a model through the dimensionality reduction process. Neural networks, k-means clustering, probabilistic clustering approaches, and other algorithms are utilized in unsupervised learning [23].

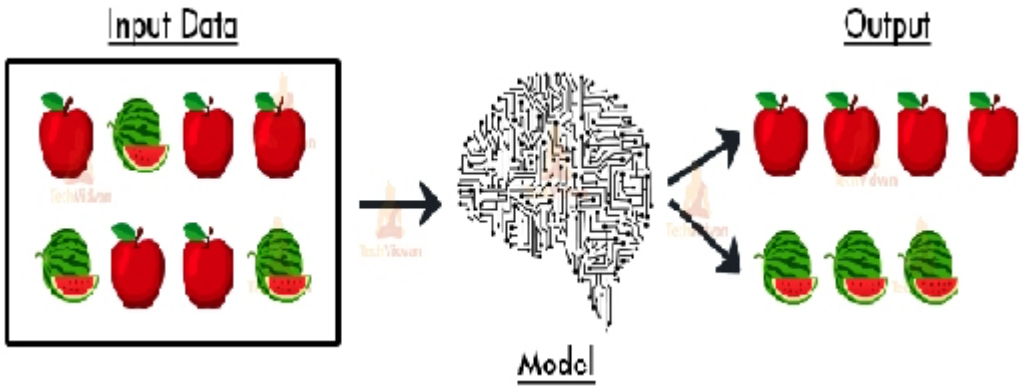


Figure 2.7: Unsupervised learning Example.

2.4.1.3 Semi-supervised learning

Between supervised and unsupervised learning, semi-supervised Figure 2.8 learning is a good compromise. It guides categorization and feature extraction from a larger, unlabeled data set using a smaller labeled data set during training. Semi-supervised learning can overcome the problem of not having enough labeled data to train a supervised learning algorithm (or not being able to afford to label enough data)([23]).

2.4.1.4 Reinforcement machine learning

is a type of learning that interacts with its surroundings by performing activities and detecting faults or rewards. The most important elements of reinforcement learning are trial and error search and delayed reward. This technology enables machines and software agents to automatically select the best behavior in a given situation in order to improve their efficiency. For the agent to learn which action is better, simple reward feedback is necessary; this is known as the reinforcement signal (see Figure 2.9) [23].

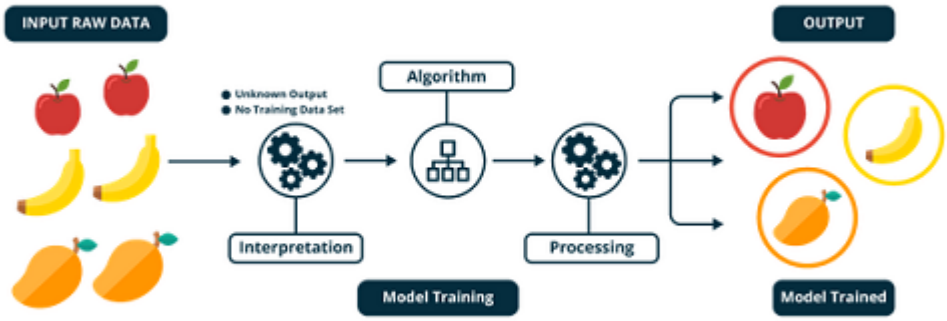


Figure 2.8: Semi-supervised learning Example.

Figure 2.9: Reinforcement Learning Example.

2.5 Deep learning

Deep learning [29] is considered as an advanced phase of machine learning [52], where it mainly uses neural networks to learn and then to predict,based on data. It is a set of different algorithms that are used to design a complex generalization system that can take any kind of problem and gives predictions. The main difference between deep learning and machine learning is that the first type of learning is based on learning features automatically (Automatically learning representations of data), while the second type requires extracting the feature manually (See

Figure 2.10). Deep learning is widely and unanimously continuous to be the best method to classify data, and it has proven its usefulness and exactitude in data classification, there for it is found in almost every field of study such as healthcare [29].

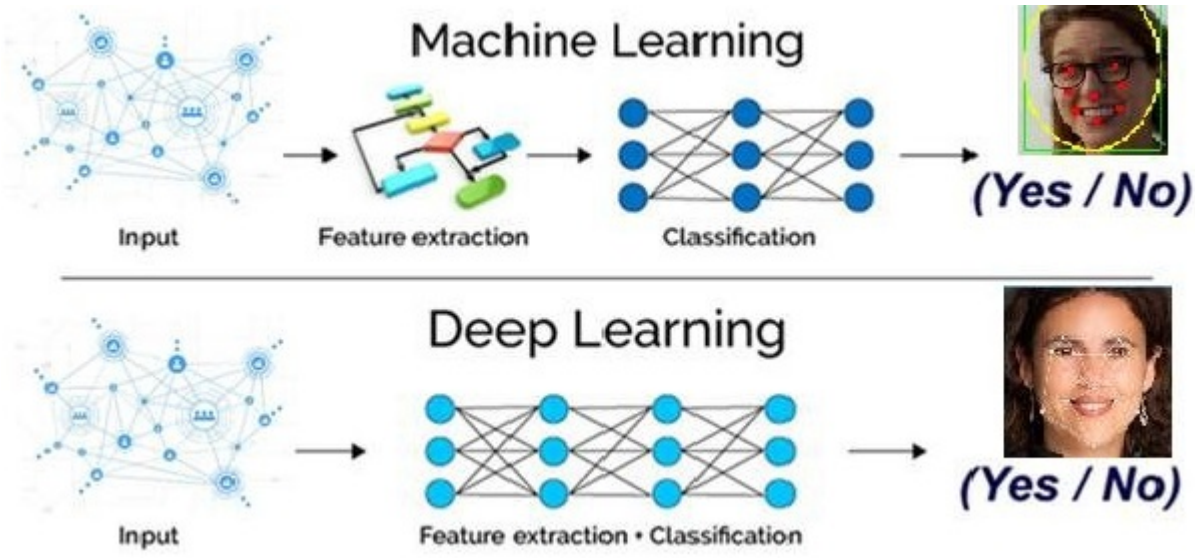


Figure 2.10: Difference between Machine Learning and Deep Learning.

2.5.1 Artificial neural networking

A neural network is a set of algorithms that uses a technique that mimics the human brain to de-tect underlying correlations in a batch of data. Artificial neural networks (See figure 2.11), or neural networks, are a subset of machine learning that are at the foundation of deep learning algorithms. The nodes in neural networks make up the layers. A node is a location where processing takes place that is loosely modeled after a neuron in the human brain. When it encounters adequate stimulus, it lights up, and it is known as a node. A node combines data input with weights that may either amplify or dampen the input, allowing it to give relevance to inputs in relation to the method it is trying to learn. Here's a diagram of what I'm talking about[Application of Artificial Intelligence in Healthcare: Chances and Challenges] [40].

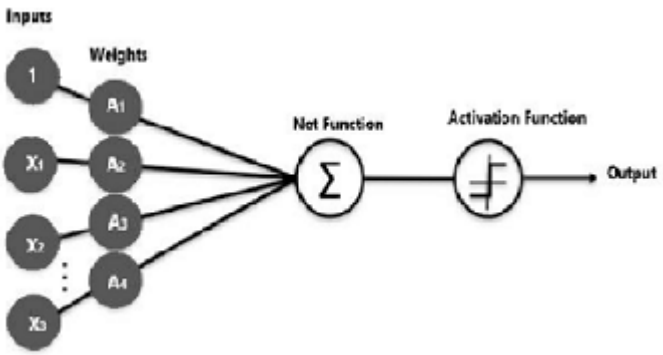


Figure 2.11: Artificial neuron.

An artificial neural network (multi-layer) is made up of many layers of neurons, including an input layer, a hidden layer, and an output layer.(see Figure 2.12) This neural network improves on the previous one, allowing it to handle more difficult issues.

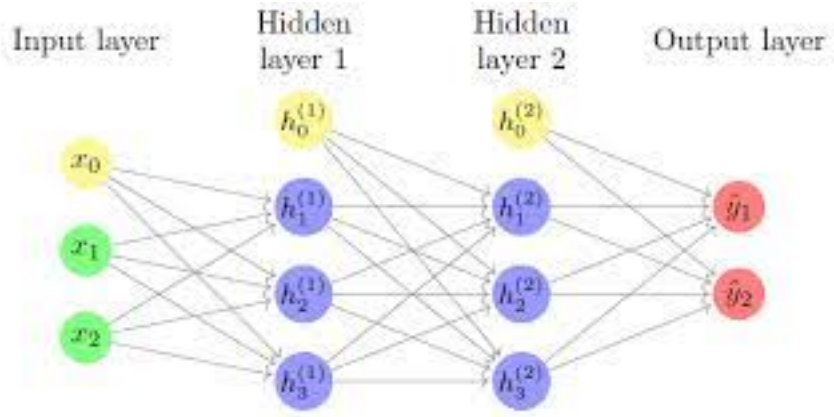


Figure 2.12: Artificial neural network.

2.5.1.1 Activation function

A signal's activation function is a mathematical function. It will mimic the activation potential discovered in human brain biology. If the stimulation threshold is met, it will enable information to flow or not allow information to pass. Its specific function will be to determine whether or not a neuron response should be activated. A neuron can only do the following tasks:

$$X = \sum(entrée \times poids) + biais \tag{2.1}$$

The activation function will be applied to this output. The following are the most common activation functions in neural networks [4]:

- **ReLU:** The Rectified Linear Unit (ReLU) function is the simplest and most widely used activation function. It returns x if x is greater than 0, 0 otherwise. In other words, it is the maximum between x and 0 see Figure 2.13 [6].

$$\text{fonction ReLU}(x) = \max(x, 0) \tag{2.2}$$

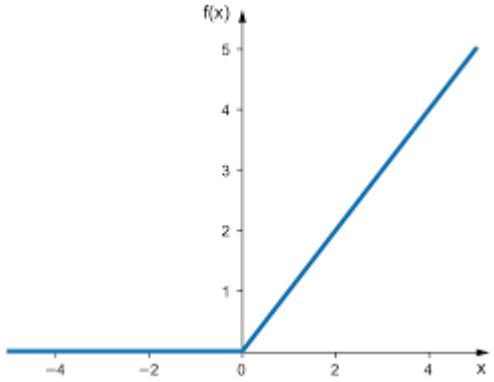


Figure 2.13: Function of RelU Rectified Linear Unit.

- **Sigmoid** The Sigmoid function returns a probability value between 0 and 1. As a result, it's extensively employed in binary classification, where a model only needs to determine two classes. As a result, for the categorization of film reviews, the closer the Sigmoid value is to 1, the more likely the model judges the review to be favorable. On the other hand, the closer it gets to zero, the more it's seen as negative (see Figure 2.14) [6].

$$\text{fonction Sigmoid}(x) = \frac{1}{1+\exp(-x)} \tag{2.3}$$

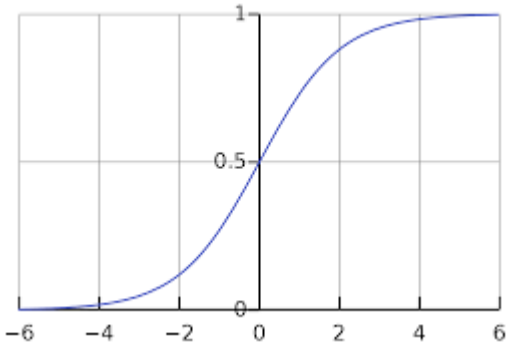


Figure 2.14: Function of Sigmoid .

- **Softmax** The Softmax function show in Figure 2.15 allows it to transform a real vector into a probability vector. It is often used in the final layer of a classification model, especially for multiclass problems. In the Softmax function, each vector is treated independently. The axis argument defines the input axis on which the function is applied [6].

$$\text{Softmax}(x) = \frac{exp(x)}{\sum exp(xi)} \tag{2.4}$$

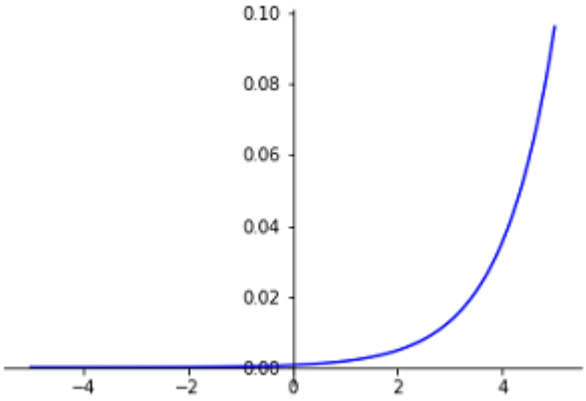


Figure 2.15: Function of Softmax .

2.5.2 Convolutional Neural Network

A Convolutional Neural Network is a type of artificial neural network that filters inputs for relevant information using convolutional layers. The convolution operation transforms a feature map by mixing input data (feature map) with a convolution kernel (filter). To extract the most usable information for a certain job, the filters in the convolutional layers (conv layers) are adjusted based on learning parameters. Convolutional networks automatically adapt to discover the optimum feature for the job at hand. When faced with a normal object identification job, the CNN would filter information about the geometry of the item, but when faced with a bird recognition test, it would extract the color of the bird. This is based on CNN's notion that distinct classes of items have varied forms, but different species of birds differ more in color than shape ([1]).

2.5.2.1 Components of a Convolutional Neural Network

An input layer, an output layer, and one or more hidden layers make up convolutional networks. The neurons in the layers of a convolutional network are organized in three dimensions, unlike those in a standard neural network (width, height, and depth dimensions, see Figure 2.16). This enables the CNN to convert a three-dimensional input volume into an output volume. Convolution layers, pool-ing layers, normalizing layers, and fully linked layers make up the hidden layers. Multiple convolutional layers are used in CNNs to filter input volumes to higher levels of abstraction. Using pooling layers for limited translation and rotation invariance, CNNs increase their detection capabilities for strangely situated objects. By lowering memory consumption, pooling also allows for the use of additional convolutional layers.By shifting all inputs in a layer towards a mean of zero and variance of one, normalization layers are used to normalize over local input areas.Other regularization strategies can be utilized, such as batch normalization, which normalizes across all activations for the whole batch, or dropout, which ignores randomly chosen neurons through-out the training phase. Fully-connected layers feature neurons that perform the same functions as convolutional layers (calculate dot products), but they are coupled to all activations in the preceding layer. Inception modules, which employ 1*1 convolutional kernels, are used in more modern CNNs to minimize memory use even more

while allowing for more efficient processing (and thus training). As a result, CNNs may be used in a variety of machine learning applications [1].

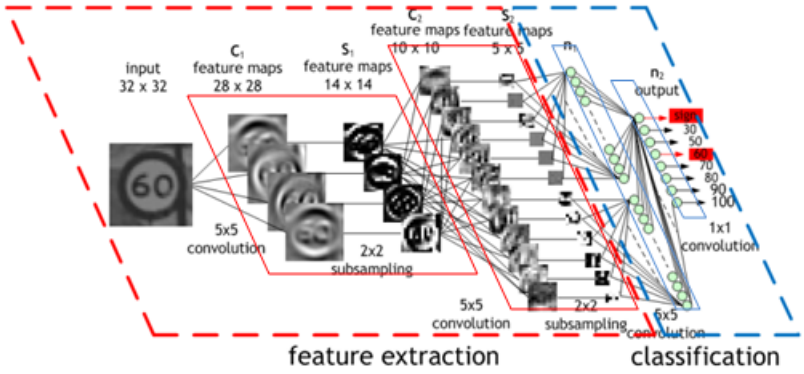


Figure 2.16: Components of a Convolutional Neural Network.

2.5.2.2 Convolutional layer

The convolution layer is the most important part of a convolutional neural network, and it is always the initial layer. Its goal is to detect the existence of a collection of characteristics in the photos that are given to it as input. The approach is to "slide" a window representing the feature over the picture and calculate the convolution product between the feature and each area of the scanned image. In this case, a feature is viewed as a filter, and the two names are interchangeable. The convolution layer therefore receives several images as input and calculates their convo-lution with each filter. The filters match the traits we're looking for in the photographs to a tee. We get an activation map, or feature map, for each pair (image, filter), which informs us where the features are in the picture: the greater the value, the more the associated region in the image resembles the feature. Unlike previous techniques, the characteristics are learnt by the network during the training phase rather than being specified according to a formality. This is the magic of convolutional neural networks: by adjusting to the situation at hand, they can recognize the distinguishing features of a picture on their own. If the topic is how to tell the difference between cats and dogs, the specific qualities can instantly define the form of the ears or feet. Basically, information provided in this paragraph about convolutional layer are raken from the site [3].

2.5.2.3 Pooling layer

The pooling layers, also known as downsampling, is a dimensionality reduction technique that reduces the number of factors in the input. The pooling process sweeps a filter across the whole input, similar to the convolutional layer, however this filter does not contain any weights. Instead, the kernel uses an aggregation function to populate the output array from the values in the receptive field.

There are two main types of pooling:

Max pooling:

The filter picks the pixel with the highest value to transmit to the output array as it advances across the input. In comparison to average pooling, this strategy is more frequently employed (see Figure 2.17).

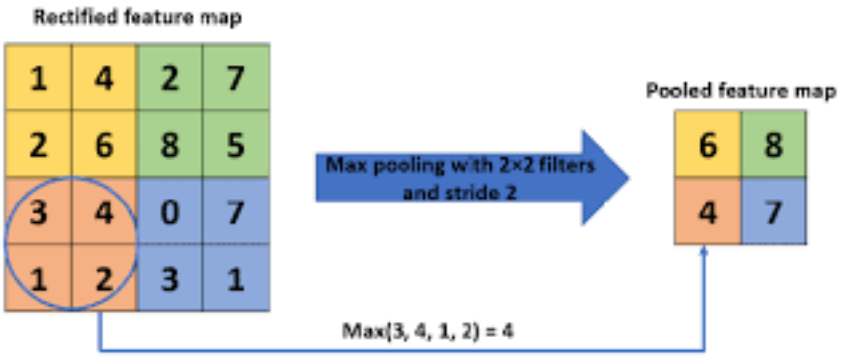


Figure 2.17: Example of Max Pooling operation .

Average pooling:

The average value inside the receptive field is calculated as the filter passes over the input and sent to the output array [11](see Figure 2.18).

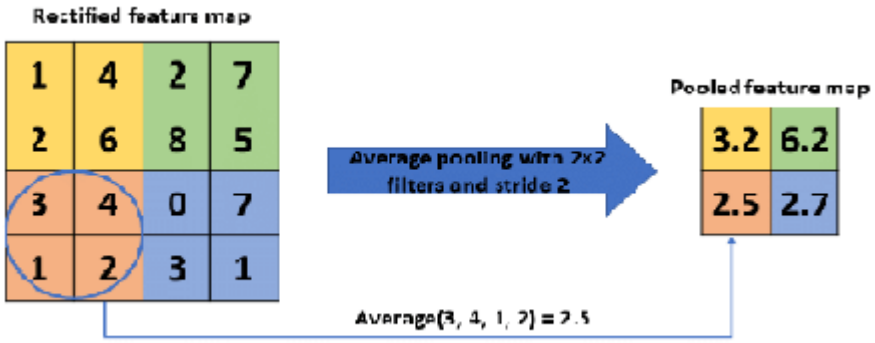


Figure 2.18: Example of Average Pooling operation .

2.5.2.4 Fully Connected layers

There may be one or more fully-connected layers after numerous convolutional and pooling layers that try to do high-level reasoning. This layer performs classification tasks based on the characteristics retrieved by the preceding layers and their various filters. While convolutional and pooling layers often utilize ReLu functions to categorize inputs, FC layers typically use a softmax activation function to provide a probability from 0 to 1 ([11]).

2.5.3 Popular CNN Architectures

There are an endless number of architectures; we'll go through a few of the more common ones, such as VGGNet.

2.5.3.1 LeNet

Yan LeCun introduced LeNet as a digit recognition system. LeNet-5 has the following basic architecture (see Figure 2.19): two convolutions (conv) layers, two subsampling levels, two fully connected layers, and a Gaussian output layer. There are 431k weights and 2.3M Multiply and Accumulates (MACs) in all.

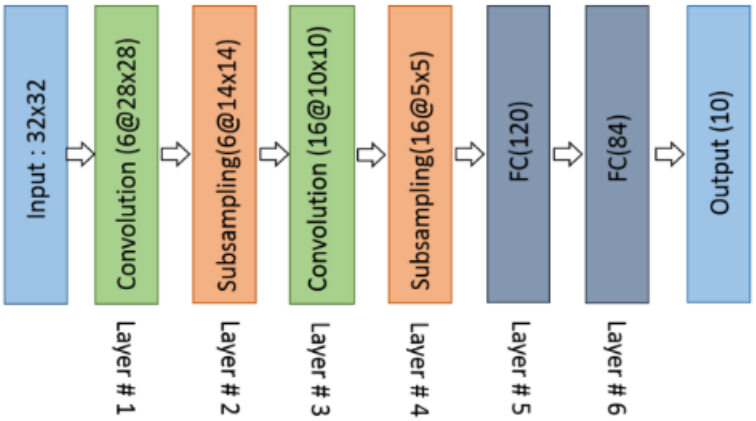


Figure 2.19: Architecture of LeNet .

2.5.3.2 AlexNet

Figure 2.20 depicts the architecture of AlexNet. The first convolutional layer uses 96 distinct 11x11 size receptive filters to accomplish convolution (conv) max-pooling (MXP) with Local Response Normalization (LRN). 3x3 filters with a stride size of 2 are used for max pooling procedures.In the second layer, the identical processes are carried out using 5x 5 filters. With 384, 384, and 296 feature maps, 3x3 filters are employed in the third, fourth, and fifth convolutional layers, respectively. With dropout, two fully connected (FC) layers are employed, followed by a Softmax layer. AlexNet was the first to demonstrate that deep learning might be used to solve computer vision problems.

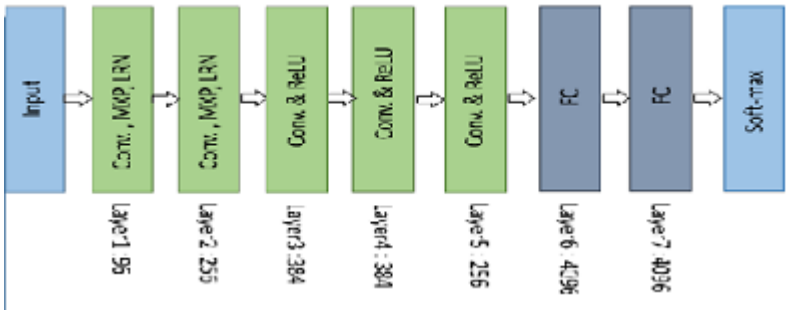


Figure 2.20: Architecture of AlexNet.

2.5.3.3 VGGNet

The VGG architecture is made up ReLU activation is used in both convolutional layers.A single max pooling layer and multiple fully linked layers follow the activation function, both of which use the ReLU activation function. A Softmax classification layer is the model's last layer(see Figure 2.21).

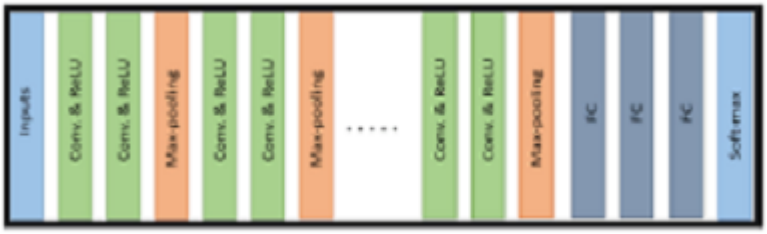


Figure 2.21: Basic building block of VGG network .

2.5.3.4 GoogLeNet

GoogLeNet used a stack of Inception layers (shown in Figure 2.22) with varied receptive fields formed by different kernel sizes to increase state-of-the-art recognition accuracy. Before the computationally costly layers, these kernels allowed for dimensionality reduction. GoogLeNet had a total of 22 layers, significantly more than any other network before it. GoogLeNet, on the other hand, used far fewer network parameters than its predecessors AlexNet and VGG.

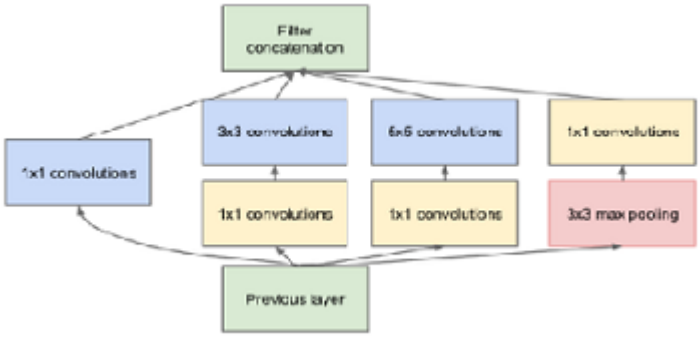


Figure 2.22: Architecture of GoogLeNet .

2.5.4 Transfer learning

Transfer learning is the reuse of a pre-trained model on a new problem. It's particularly popular in deep learning right now 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. After little tweaking, most popular CNN architectures may be utilized for transfer learning [15].

2.6 Healthcare

2.6.1 Definition

Medical practitioners' attempts to restore our bodily and emotional well-being, this action is referred to as health care. The term can also apply to services that assist people in maintaining their emotional health. The people and organizations that provide these services are called health care providers. The term also applies to the organized provision of medical care to humans or a community [5].

2.6.2 Stages

Healthcare is divided into four levels; primary, secondary, tertiary, and quaternary.

- **Primary care:** A primary care Figure 2.23 is a type of broad treatment in which a person is examined by a general physician or practitioner for a medical problem. Patients in need of medical attention are treated by general doctors since the degree of damage in a case requiring basic care is not too severe, dentists, pediatricians, nurses, and OB/GYNs are among the primary care providers. Although some of these medical professionals are specialized in treating certain groups of in-dividuals, their primary goal is to improve a person's general health and minimize the likelihood of hospitalization. Hence, the primary care is the most fundamental kind of medical treatment, and it has been demonstrated to save money in the healthcare system by lowering emergency room visits and long-term hospital stays [9].

Figure 2.23: Primary care.

- **Secondary care:** Secondary care Figure 2.24 is the second level of healthcare. The next level of patient care begins when a doctor recommends a patient to a specialist. A person is treated by an expert in the area that is making them unwell at this stage. Secondary care experts are specialized on certain bodily systems and parts. A cardiologist, for example, is a specialist in medical matters concerning the human heart. A gastroenterologist, on the other hand, is a doctor who specializes in diseases of the stomach and gastrointestinal system [9].

- **Tertiary care:** Tertiary care occurs after a person has been hospitalized and requires a higher degree of care that is not limited to a single bodily component or system. Once a patient is admitted to the hospital, their biological functions are continuously monitored by medical technology [9]. Tertiary care necessitates a high degree of medical competence as well as high-tech equipments.

- **Quaternary care:** Quaternary care is the last type of healthcare and is an extension of tertiary care. Because it necessitates complex technology and procedures, quaternary care is not available at every institution. It's not a common type of patient care because it's only used when other therapies have failed to relieve a patient's symptoms. Drug studies and experimental medical therapies may be included.

 *

Figure 2.24: Secondary care .

The majority of people only come into contact with primary and secondary medical providers once or twice throughout their lifetimes. In the event of a serious injury or accident, a person may be admitted in the tertiary care [9].

2.6.3 Artificial intelligence and healthcare

There are many ways where AI can positively impact Healthcare practice, either by accelerating the pace of research or by helping clinicians making better decisions. Here are some examples of how AI could be used ([7]):

- **Artificial intelligence in disease detection and diagnosis:** Unlike humans, the AI never requires sleep. Machine learning models could be used to observe the vital signs of patients receiving intensive care and alert clinicians if certain risk factors increase. While medical devices such as heart monitors can track vital signs, AI can collect data from these devices and look for more complex conditions, such as sepsis.

- **Personalized disease treatment:** With virtual assistance from AI, precision medicine could be more manageable. Because AI models can learn and retain preferences, AI has the potential to provide real-time, personalized recommendations to patients around the clock.

Rather than repeating information with a new person each time, a healthcare system could offer patients round-the-clock access to an AI-powered virtual assistant who could answer questions based on the patient's medical history, preferences, and personal needs.

- **Effectiveness of clinical trials:** During clinical trials, a lot of time is spent assigning medical codes to patient results and updating relevant data sets. AI can help speed up this process by enabling faster and smarter medical code searches. Medical companies have recently found that AI helps them reduce the number of medical code searches by more than 70%.

- **Accelerated drug development:** Drug discovery is often one of the longest and most expensive parts of drug development. AI could help reduce new drug development costs in two main ways: by creating better drug designs and by finding promising new drug combinations. Using AI would overcome many of the big data challenges faced by players in the life sciences industry.

2.7 Related works

In this part, we will present some previous works similar to ours.

- **Classification of Images of Childhood Pneumonia using Convolutional Neural Networks:** This is the work of [45] they provided a description a comparative classification of Pneumonia using Convolution Neural Network. The database used was the dataset Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification made available by [34] with a total of 5863 images, with 2 classes: normal and pneumonia. In order to evaluate the generalization capacity of the models, cross-validation of k-fold was used. The classification models proved to be efficient compared to the work of [34] which obtained 92,8 and the present work had an average accuracy of 95.30.

- **Classification of x-ray images for detection of childhood pneumonia using pre-trained neural networks:** This is the work of A [43] In this work they described a comparison between three pre-trained neural networks for the classification of chest X-ray images: Xception, Inception V3, and NasNetLarge. Networks were implemented using learning

transfer; The database used was the chest x-ray data set, which contains a total of 5856 chest x-ray images of pediatric patients aged one to five years, with three classes: Normal Viral Pneumonia and Bacterial Pneumonia. Data were divided into three groups: validation, testing and training. A comparison was made with the work of [34] who implemented the Inception V3 network in two ways: (Pneumonia X Normal) and (Bacterial Pneumonia X Viral Pneumonia). The nets used had good accuracy, being the NasNetLarge network the best precision, which was 95.35 (Pneumonia X Normal) and 91.79 (Viral Pneumonia X Bacterial Pneumonia) against 92.80 in (Pneumonia X Normal) and 90.70 (Viral Pneumonia X Bacterial Pneumonia) from kermany's work, the Xception network also achieved an improvement in accuracy compared to kermany's work, with 93.59 at (Normal X Pneumonia) and 91.03 in (Viral Pneumonia X Bacterial Pneumonia).

- **Pneumonia Detection from Chest X-ray Images Based on Convolutional Neural Network:** This is the work of A [20] In this work, they developed a straightforward VGG-based model architecture with fewer layers. In addition, they tackled the inadequate contrast of chest X-ray images, which brings about ambiguous diagnosis, the Dynamic Histogram Enhancement technique is used to pre-process the images. The parameters of our model are reduced by 97.51 compared to VGG-16, 85.86 compared to Res-50, 83.94 compared to Xception, 51.92 compared to DenseNet121, but increased MobileNet by 4. However, the proposed model's performance (accuracy: 96.068, AUC: 0.99107 with a 95 confidence interval of [0.984, 0.996], precision: 94.408, recall: 90.823, F1 score: 92.851) is superior to the models mentioned above (VGG-16: accuracy, 94.359, AUC: 0.98928; Res-50: accuracy, 92.821, AUC, 0.98780; Xception: accuracy, 96.068, AUC, 0.99623; DenseNet121: accuracy, 87.350, AUC, 0.99347; MobileNet: accuracy, 95.473, AUC, 0.99531). The original Pneumonia Classification Dataset in Kaggle is split into three sub-sets, training, validation and test sets randomly at ratios of 70, 10 and 20. The model's performance in pneumonia detection shows that the proposed VGG-based model could effectively classify normal and abnormal X-rays in practice, hence reducing the burden of radiologists.

- **Development of a CNN architecture for the classification of x-ray images of infections pulmonary:** MAMEN Abdelkarim and SAOULI Rachida, 2021 thesis from the university

of mohamed khieder-Biskra. They built a convolutional neural network to classify lung diseases, (viral and bacterial pneumonia, COVID-19, and tuberculosis). They collected the data set from kaggel. And reported an training accuracy of 98.36% and a accuracy validation of 91.774% ,testing accuracy 91.53%.

2.8 Conclusion

In this chapter, we have covered the main aspects which are the basics of our current work,We have discussed health care, and it's stages, how artificial intelligence affects it. We have put up a comprehensive description of the human lung and its anatomy, as well as information regarding pneumonia and its effects on children. We have also covered the fundamentals of machine learning and deep learning. We as well referenced some past relevant work. The following chapter will describe our system design for a new deep learning architecture for pneumonia detection in pediatric.

Chapter 3

Design of a deep learning architecture for pneumonia detection in pediatric

3.1 Introduction

For the early detection of pneumonia in children, we have developed a novel CNN architecture. In this chapter, we do try to discuss it. We will first present our general architecture and then we will detail it. We talk about preparing the data (pre-processing and dataset splitting phases), then training the model and testing it. In the next section, we will present the detailed operation of our model.

3.2 System design

In order to build a deep learning model which can categorize data (x-ray pictures for pneumonia detection in pediatric) in two different cases: a patient with pneumonia or a healthy person, we following steps: first we prepare the dataset, where the system obtains the data then split it and also performs pre-processing on it. Second we train the CNN model on the splitted dataset, and finally we have a model which is able to classify new pneumonia images. Figure 3.1 depicts the whole structure of the system.

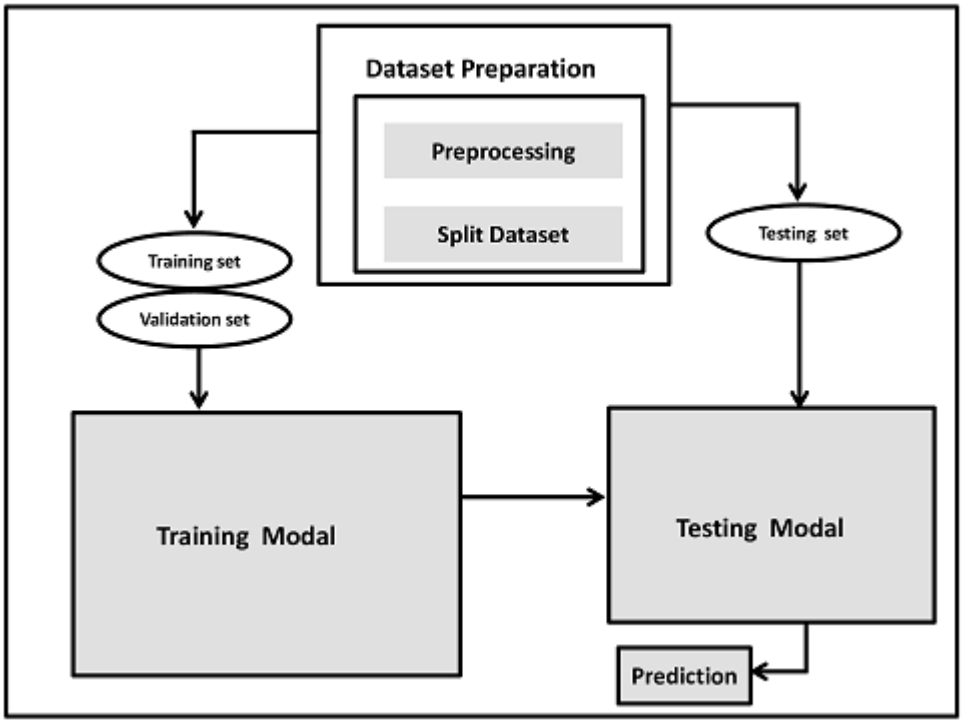


Figure 3.1: Diagram of our System.

3.2.1 Data preparation

In this work we have developed, from scratch, a CNN model that can detect pneumonia in early stages of childhood. Our System follows certain steps, it begins by obtaining the dataset, afterwards, preprocessing and splitting it, and then feeding the split dataset to the CNN model. As a result, we will have a model that can classify chest X-ray images into two classes (healthy or pneumonia).

3.2.1.1 Data set

- **First Dataset:** We have used the data set "Pediatric Pneumonia Chest X-ray" available in Kaggle website [2]. The set of images contains 5856 X-ray images (JPEG) divided intwo three categories (Viral Pneumonia, Bacterial Pneumonia and Normal) provided by the [34]. You can see in the Fugger 3.2. Chest x-ray images (anteroposterior) were selected from pediatric patients aged one to five years.

  The dataset still has quality control, where garbled, ow-quality images have been removed.

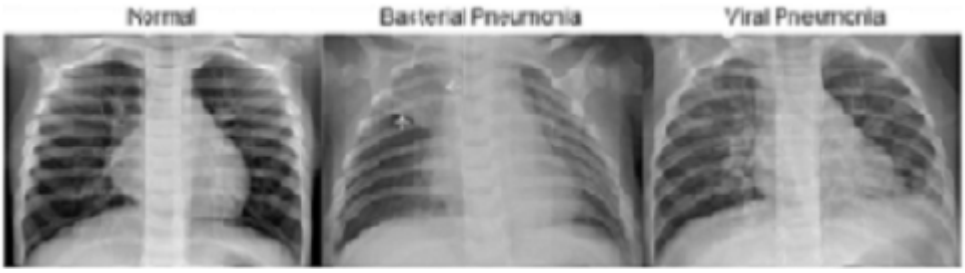


Figure 3.2: Illustrative examples of x-rays in patients with and without Pneumonia.

The diagnosis was classified by two physicians specialist and checked by a third expert in order to extinguish the errors [34]. The dataset consists of 5856 images divided as follows 1583 images of normal patients, 2780 of patients with bacterial pneumonia, 1493 of patients with viral pneumonia. We used it in training, verification and test.

Second Dataset: [42] Chest X-Ray Images (Pneumonia) available in Kaggle website. The dataset is organized into 3 folders (train, test, validation) and contains subfolders for each image category (Pneumonia/Normal). There are 5,863 X-Ray images (JPEG) and 2 categories (Pneumonia/Normal). Chest X-ray images (anterior-posterior) were selected from retrospective cohorts of pediatric patients of one to five years old from Guangzhou Women and Children's Medical Center, Guangzhou. All chest X-ray imaging was performed as part of patients' routine clinical care. We used this dataset in the test.

3.2.1.2 Pre-processing

We ought to pass through the pre-processing stage before sending the data to the CNN model for training. This phase takes a lot of time and effort because a large amount of information is needed from the experts of the needed field to adequately adapt the data into the best case results. In this work, we resized our images which were sized (1024 x 1024 x 3) to be all in shape (224 x 224 x 3), width in pixels, height in pixels and three red-green-blue colors respectively. In fact, the best results obtained in this work are achieved with the previous size of the input images. We have resized our images, because if we use large images directly, it will require the use of a larger size input shape in the CNN architecture, which will increase the number of parameters in the CNN model and lengthen the training process.

3.2.1.3 Splitting dataset

In deep learning, the most common dataset splitting approach is to divide the dataset into training, validation, and test sets. For each problem, the ideal sample distribution ratio in each set changes. In this work we divided our data as follows (See Figure 3.3):

- **70% for training.**
- **20% for validation .**
- **10% for the test.**

This split is the most common in deep learning projects.

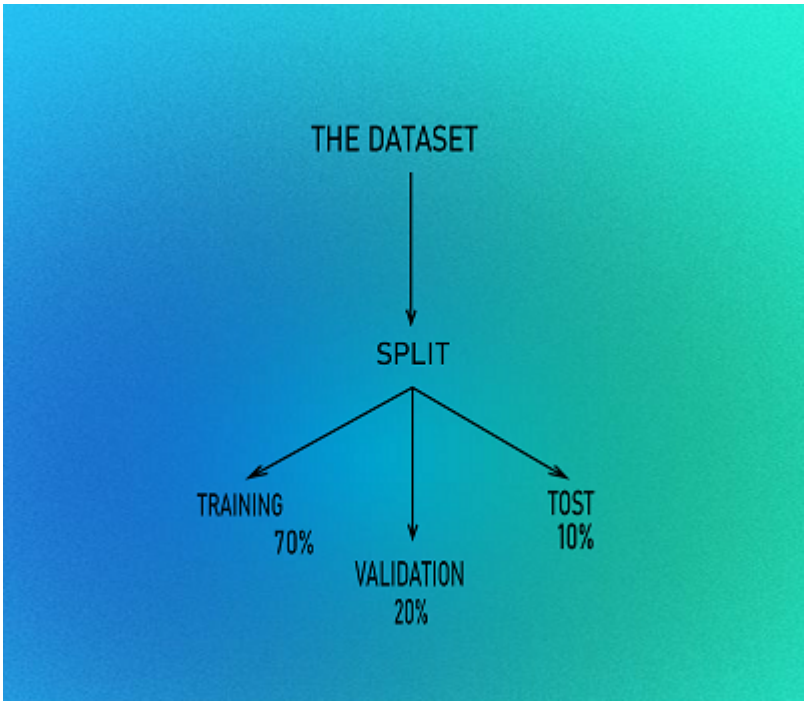


Figure 3.3: Spilt.

3.2.2 Training Model

In order to obtain a function capable of distinguishing between the previous images (patients with pneumonia and healthy people), we attempted to train our structure as shown in the Figure 3.4 of the training set.

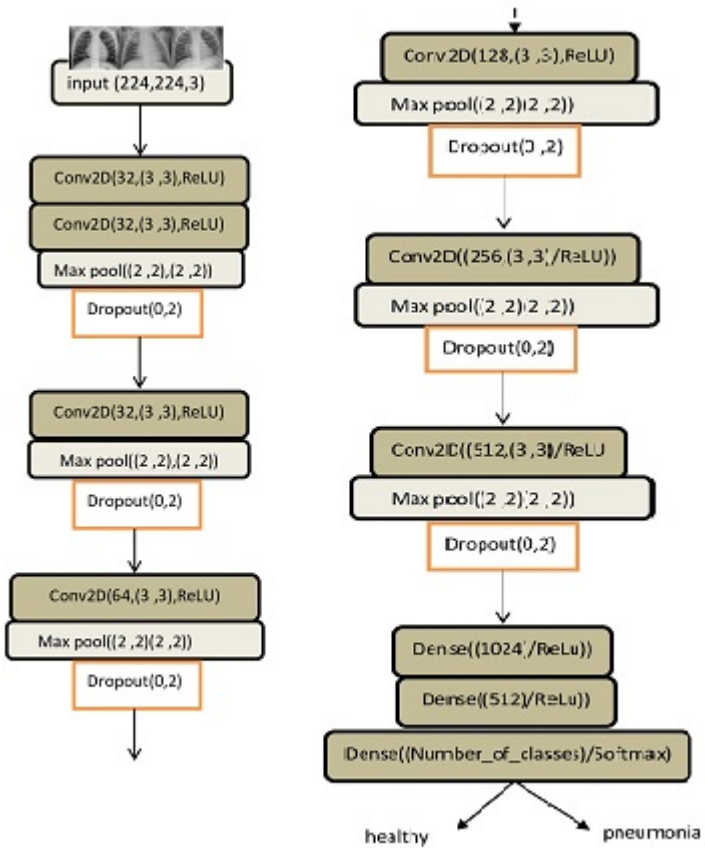


Figure 3.4: Our CNN Model.

Our structure contains convolutional, pooling, and fully connected layers, with the convolutional and pooling layers being critical aspects in realizing the CNN's extraction capabilities. To accomplish the extraction and mapping of local features from the training set, the structure is first arranged alternatively by convolution layer and grouping layer, then progressively by numerous fully connected layers.

The network's final layer uses a soft-max approach to classify those feature maps and output scores proportional to the "probability" that each patch belongs to each of the learnt classes. We used Dropout layers with an arbitrary rate(generally from 0.1 to 0.5)so that we can avoid over fitting

The accuracy of overall prediction is greatly influenced by the initialization of CNN parameters. The hyper-parameters were empirically initialized and fine-tuned during the training phase. Table 3.1 lists the hyper-parameters that were employed in this investigation.

To test our intermediately trained model, we employ a validation set. The number of feature maps, the kernel size of the convolutional layers, and the filter size of the grouping layers in the network,these elements are tuned using the accuracy or error of the validation set training output as a feedback parameter. We modified the hyper-parameters and updated the training dataset throughout the training-validation cycles until the training accuracy was adequate, and then we had our CNN architecture.

	parameter	hyper parameter
Convolutional layer	Kernels	Kernel size , number of kernels stride ,padding and activation function
Pooling layer	None	Pooling method ,filter size , padding and strides
Fully connected layer	weights	number of weights
dropout layers	None	dropout rate

Table 3.1: The hyperparameters used in the convolutional neural network

3.2.3 Testing Model

By submitting an image to the network, of the same size used from the training, we obtain a class prediction from the learned model. A test set has to be used just once, at the very end of the project, to evaluate the performance of the final model, which has been fine-tuned and selected using training and validation sets. Ensemble predictions on test set were analyzed for true positives (TP), true negatives (TN), false positives (FP), and false negatives (FP). These parameters were used to calculate a variety of performance metrics, including including precision

(PR), accuracy (ACC), F1-score (F1), according to the following equations:

$$\begin{aligned} Accuracy &= \frac{TP+TN}{TP+TN+FP+FN} \\ Precision &= \frac{TP}{TP+FP} \\ F1-score &= \frac{TP}{TP+\frac{1}{2}(FP+FN)} \end{aligned} \tag{3.1}$$

3.3 Conclusion

In this chapter, we have presented the dataset used and we have discussed how it was prepared (phases of pre-processing and spliting data), also we have discusses about training the model and then testing it. In the next chapter, we will present the implementation of a big part of our system and we will give details about CNN model.

Chapter 4

Implementation and results

4.1 Introduction

In the current chapter, we present the adopted methods in order to design a deep learning system for pneumonia detection using the CNN model. This chapter presents the development environments and tools as well as the implementation and current code of our system. We will also present the obtained results.

4.2 Development environments and tools

4.2.1 Google colab

The term Colaboratory, often shortened to "Colab" ([24]), is a product of Google Research. Colab allows anyone to write and run Python code of their choice through the browser. It is an environment particularly suitable for machine learning, data analysis and education. In more technical terms, Colab is a hosted Jupyter notebook service that requires no configuration and provides free access to computing resources, including GPUs.

4.2.2 Python

Python [44] is a dynamically semantic, interpreted, object-oriented high-level programming language. Its high-level built-in data structures, together with dynamic typing and dynamic binding, making it ideal for Rapid Application Development and as a scripting or glue lan-

guage for connecting existing components. Python's simple, easy-to-learn syntax emphasizes readability, which lowers program maintenance costs. Modules and packages are supported by Python, which fosters program modularity and code reuse. The Python interpreter and its substantial standard library are free to download and distribute in source or binary form for all major platforms.

4.2.3 TensorFlow

[16] is an end-to-end open source machine learning platform. It offers a comprehensive and flexible ecosystem of tools, libraries, and community resources that allow researchers to advance in the field of machine learning, and developers to easily create and deploy applications that use this technology.

4.2.4 Keras

Keras [25] is an open source library written in Python (under MIT license) based mainly on the work of Google developer François Chollet as part of the ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System) project. A first version of the cross-platform software was released on March 28, 2015. The purpose of this library is to enable rapid building of neural networks. In this framework, Keras does not function as its own framework but as an application programming interface (API) for accessing and programming different machine learning frameworks. Theano, Microsoft Cognitive Toolkit (formerly CNTK) and TensorFlow are among the frameworks supported by Keras.

4.2.5 Numpy

NumPy ([12]) is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

4.2.6 Matplotlib

Matplotlib [48] is a plotting library available for the Python programming language as a component of NumPy, a big data numerical handling resource. Matplotlib uses an object oriented API to embed plots in Python applications.

4.2.7 Kaggle

[8] is an online community of data scientists and machine learning practitioners. Kaggle allows users to find and publish datasets.

4.3 Developing the back-end

We utilized Python to create our deep learning CNN model, which included well-known deep learning APIs and tools. Keras covers the efficient TensorFlow numerical computing library, making it one of the most powerful and easy-to-use Python frameworks for constructing and assessing deep learning models. The steps below show how we used Python and Keras to create a deep learning CNN model.

4.3.1 Dataset preparation

4.3.1.1 Download Dataset to Google Colab

First, we download the data from Kaggle to our google Colab account

```
! kaggle datasets download -d andrewmvd/pediatric-pneumonia-chest-xray
}
```

4.3.1.2 Pre-processing

We have sized the images from (1024 x 1024 x 3) to (224 x 224 x 3) as previously shown.

```
import PIL
import os
import os.path
from PIL import Image
f ='/content/newDataset3/train/PNEUMONIA'
```

```
for file in os.listdir(f) :
    f_img = f+"/"+file
    img = Image.open(f_img)
    img = img.resize((224,224))
    img.save(f_img)
```

Listing 4.1: Resize images

4.3.1.3 Splitting Dataset

As mentioned earlier, we have split our dataset into 70 training, 20 validation and 10 test. by using split folder.ratio. splitfolders.ratio(input folder,output,seed,ratio). Details are in Table 4.1 and Listing 4.2

```

input_folder='/content/Pediatric Chest X-ray Pneumonia/train'
output="/content/newDataset3"
splitfolders.ratio(input_folder ,output ,
seed =1337 , ratio =(.7 ,.2 ,.1))
}
```

Listing 4.2: Splitting Dataset

Function	Argument
Split folder.ratio()	input folder; path to patches dataset seed; set seed value for shuffling the items; defaults to 1337. ratio: division percentage (.7,.2,.1) Output folder; path to final dataset

Table 4.1: Split with a ratios.

4.3.2 Building our CNN Model

4.3.2.1 Import libraries and modules

To use Keras to create deep neural networks, we must first import the different libraries and modules listed in Listing 4.3 and detailed in Table 4.2.

```
import keras,os
from keras.models import Sequential
from keras.layers import Dense, Conv2D, MaxPool2D , Flatten,Dropout
```

```
from keras.preprocessing.image import ImageDataGenerator
import numpy as np
from keras.callbacks import ModelCheckpoint, EarlyStopping
```

Listing 4.3: Import libraries and modules

4.3.2.2 Loading dataset

The ImageDataGenrator class was used to load into memory our training and validation datasets, In this step target size and normalization data and class mode are defined. Afterwards, an iterator is needed to progressively get images for a single dataset. It is done by calling the flow from directory() function and specifying the dataset directory, such as the train, test, or validation directory. More details are found in listing 4.4 .

```
trdata = ImageDataGenerator()
traindata = trdata.flow_from_directory(directory="/content/newDataset3/
    train",target_size=(224,224))
tsdata = ImageDataGenerator()
testdata = tsdata.flow_from_directory(directory="/content/newDataset3/val"
    , target_size=(224,224))
```

Listing 4.4: Loading dataset

4.3.2.3 Initialization of CNN model parameters

We must first initialize the network's parameters before we can begin training it. The parameters' values are calculated based on previously proposed values discovered in the literature and via experimentation. The hyper-parameters used in this study are described in detail in Table 4.3.

4.3.2.4 Creating the model

After loading the datasets and initializing the hyper-parameters, we designed our architecture, according to Listing, which was determined after trying many CNN configurations. The Table 4.4 describes the layers of the CNN model used.

```
model = Sequential ()
```

Listing 4.5: Sequential model

The sequential model is a method of creating deep learning models in which an instance of the Sequential class and model layers are created and added to it.

Libraries/modules	Descriptions
Numpy	Is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays.
Sequential	For creating deep learning models where an instance of the Sequential class and model layers are created and added to it. A Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor.
ImageDataGenrator	It's used to charge our train, validate memory data sets, and generate tensor lots. picture data with real-time data augmentation
target_size	If the image size does not match the goal size, all photos will be scaled to the target size (224,224)
Dense	It's utilized to create a dense layer.
Dropout	A dropout is applied to the input. Dropout is a technique for avoiding overtraining by setting an input unit fraction rate to 0 at random during each update throughout the training period.
Activation	Adds an activation function to the layer sequence.
Flatten	The map of grouped features is converted into a single column that is sent to the fully connected layer.
Conv2D	Convolutes the layer input with a convolution kernel to generate an output tensor.
Maxpoling2D	The input representation was downsampled by taking the largest value for each dimension along the feature axis across the window provided by the pool size.
Adam	Adam's approach is implemented by an optimizer. The Adam optimization method is a stochastic gradient descent method based on adaptive estimate of first and second order moments.

Table 4.2: Description of libraries and models .

```
#inpot 1
model.add(Conv2D(input_shape=(224,224,3),filters=32,kernel_size=(3,3),
    padding="same", activation="relu"))
model.add(Conv2D(filters=32,kernel_size=(3,3),padding="same", activation="
    relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
model.add(Dropout (0.2))
#layer 2
model.add(Conv2D(filters=32,kernel_size=(3,3),padding="same", activation="
    relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
model.add(Dropout (0.2))
#layer 3
model.add(Conv2D(filters=64,kernel_size=(3,3),padding="same", activation="
```

Hyper-parameters	Descriptions	Values
Input shape	Size of input patch (image)	(224,224,3)
Epochs	The number of epochs is a hyperparameter that defines the number times for learning	100 .
Batch size	is a hyper parameter that specifies how many samples must be processed before the internal model parameters are updated	32 Default
Patience	The number of epochs that will stop the training if there are no improvement.	30.
. Class num	The number of classes	2.

Table 4.3: Hyper-parameter description.

```
    relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
model.add(Dropout (0.2))
#layer 4
model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same",
    activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
model.add(Dropout (0.2))
#layer 5
model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same",
    activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
model.add(Dropout (0.2))
#layer 6
model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same",
    activation="relu"))
model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
model.add(Dropout (0.2))
#layer 7
model.add(Flatten())
model.add(Dense(units=1024,activation="relu"))
#layer 8
model.add(Dense(units=512,activation="relu"))
model.add(Dropout (0.2))
#layer 9
model.add(Dense(units=2, activation="softmax"))
```

Listing 4.6: Creating CNN model

Layers	Composed of
Layer number 1	Conv2D(input_shape =(224 ,224 ,3)(32,(3,3)), MaxPooling(2,2) Conv2D(32,(3,3)),MaxPooling(2,2) , padding("same") ,activation("relu"),Dropout(0.2)
Layer number 2	Conv2D(32,(3,3)),MaxPooling(2,2), padding("same") ,activation("relu"),Dropout(0.2)
Layer number 3	Conv2D(64,(3,3)),MaxPooling(2,2), padding("same") ,activation("relu"),Dropout(0.2)
Layer number 4	Conv2D(128,(3,3)),MaxPooling(2,2), padding("same") ,activation("relu"),Dropout(0.2)
Layer number 5	Conv2D(256,(3,3)),MaxPooling(2,2), padding("same") ,activation("relu"),Dropout(0.2)
Layer number 6	Conv2D(512,(3,3)),MaxPooling(2,2), padding("same") ,activation("relu"),Dropout(0.2)
Layer number 7	Flatten(),Dense(1024),activation("relu")
Layer number 8	Dense(512),activation("relu"), Dropout(0.2)
Layer number 9	Dense(number of classes), activation("softmax")

Table 4.4: CNN Layers composition

To add layers to our model, we used the 'add()' method (see Listing 4.6). In our output layer, we have num nodes (classes num=2), one for each conceivable outcome (0-1).

4.3.2.5 Compiling CNN Model

Once the model is created, we can config it with model.compile() method , which takes three arguments: optimizer, loss and metrics (see Table 4.5).

The optimizer controls the learning rate. We have used: - 'adam' as our optimizer, it adjusts the learning rate throughout training. The learning rate 'lr' determines how fast the optimal weights for the model are calculated.

- 'categorical crossentropy' for our loss function measures the performance of a classification model ,it's output is a probability value between 0 and 1.

- We used 'accuracy' and 'prcesion' metrics to see the accuracy score on the validation set while training the model. It is a function that is used to judge the performance of your model.

```
from tensorflow.keras.optimizers import Adam
opt = Adam(0.0001)
model.compile(optimizer=opt,
loss=keras.losses.categorical_crossentropy, metrics=['accuracy','Precision
    '])
```

Listing 4.7: Compiling Model

Function	Arguments
model.Compile()	optimizer = adam : The adam optimizer adjusts the learning rate throughout training learning rate = 0.0001 Loss function = categorical crossentropy Metrics = ['Precision','accuracy',]

Table 4.5: Compiling Model arguments

4.3.2.6 Model summary

Summary method can be called to display the content of our model once it's ready (see Fugger 4.1).

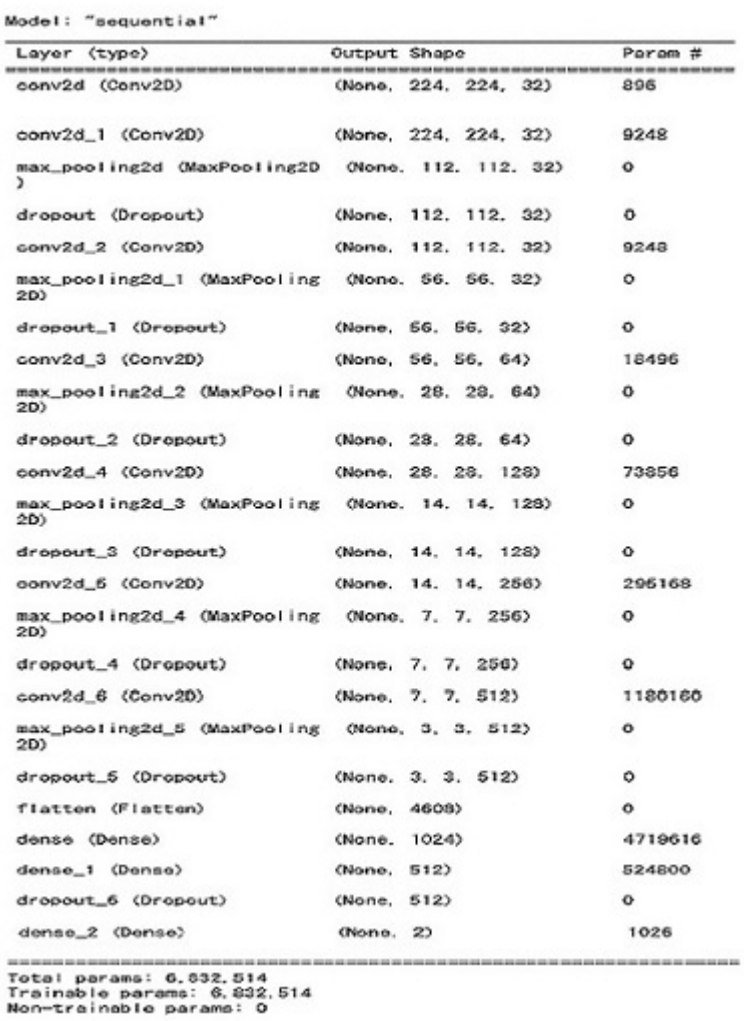
```
model.summary ()

```

Listing 4.8: Model summary

4.3.2.7 Training CNN Model

Fitgenerator, ModelCheckpoint, and Earlystopping were used to train the model with different arguments based on the structure, we'll describe the role of each argument and other functions in this section.

```
checkpoint = ModelCheckpoint("my_model.h5", monitor='val_accuracy',
    verbose=1, save_best_only=True, save_weights_only=False, mode='auto',
    Shuffle = True, period=1)
early = EarlyStopping(monitor='monitor ', min_delta=0, patience=30,
    verbose=0, mode='auto')
```



Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 224, 224, 32)	896
conv2d_1 (Conv2D)	(None, 224, 224, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 112, 112, 32)	0
dropout (Dropout)	(None, 112, 112, 32)	0
conv2d_2 (Conv2D)	(None, 112, 112, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 32)	0
dropout_1 (Dropout)	(None, 56, 56, 32)	0
conv2d_3 (Conv2D)	(None, 56, 56, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 64)	0
dropout_2 (Dropout)	(None, 28, 28, 64)	0
conv2d_4 (Conv2D)	(None, 28, 28, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 128)	0
dropout_3 (Dropout)	(None, 14, 14, 128)	0
conv2d_5 (Conv2D)	(None, 14, 14, 256)	295168
max_pooling2d_4 (MaxPooling2D)	(None, 7, 7, 256)	0
dropout_4 (Dropout)	(None, 7, 7, 256)	0
conv2d_6 (Conv2D)	(None, 7, 7, 512)	1180160
max_pooling2d_5 (MaxPooling2D)	(None, 3, 3, 512)	0
dropout_5 (Dropout)	(None, 3, 3, 512)	0
flatten (Flatten)	(None, 4608)	0
dense (Dense)	(None, 1024)	4719616
dense_1 (Dense)	(None, 512)	524800
dropout_6 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 2)	1026

Total params: 6,832,514
Trainable params: 6,832,514
Non-trainable params: 0

Figure 4.1: Model summary.

```
hist = model.fit_generator(steps_per_epoch=len(traindata),generator=
    traindata, validation_data= testdata, validation_steps=len(testdata),
    shuffle= True ,epochs=100,callbacks=[checkpoint,early])
```

Listing 4.9: Training Model

- **ModelCheckpoint :** callback is used in conjunction with training using model.fit() to save a model or weights (in a checkpoint file) at some interval.
- **EarlyStopping :** This callback allows you to specify the performance measure to monitor, the trigger, and once triggered, it will stop the training process, the main objective of this method is to avoid over fitting.

Function	Arguments
ModelCheckpoint	-Model name -Monitor of performance -Verbose mode 0,1 or 2 , we used 1 for an animated progress save best only , if = true it only saves when the model is considered the "best" -save weights only , if = false , the full model is saved -mode ,one of max,min or auto ,in our case the save best only will overwrite the mode mode period is just an additional argument for backwards compatibility

Table 4.6: Function description for the Checkpoint

Function	Arguments
EarlyStopping	- Monitor: Quantity to be monitored. -Monitor of performance -Patience: Number of epochs with no improvement after which training will be stopped -Verbose: Verbosity mode, 0 or 1. Mode 0 is silent, and mode 1 displays messages when the callback takes an action. -mode is one of max,min or auto ,here it will follow the monitor if val accuracy then mode will be max,if val-loss then the mode will be min

Table 4.7: Function description for the earlyStopping

- **model.fitgenerator :** This method is used to train the model.

4.3.3 Testing our CNN Model

We tested the trained model that is ready for prediction by using the third subset. We have used this code shown in "listing" befor starting the testing process. Data such as the training process using function must first be entered into memory. We also used load-model to charge the models and we used the predict-generator to generate predictions from the data generator input samples.

```
import matplotlib.pyplot as plt
from tensorflow import keras
from keras.models import load_model
from tensorflow import keras
from keras.preprocessing import image
```

Function	Arguments
fit_generator	-step per epoch is calculated using len(traindata). -validation-steps is calculated using len(testdata)) -epochs = 100 - Callbacks; a list of callback functions applied during the training of our model callbacks=[checkpoint,early].

Table 4.8: Function description for the model fitting

```
import PIL
import os
import os.path
from PIL import Image
from os import listdir
from PIL import Image as PImage
import numpy as np
from keras.preprocessing.image import ImageDataGenerator
from keras.preprocessing.image import img_to_array
from keras.preprocessing.image import load_img
from keras.preprocessing.sequence import TimeseriesGenerator

import numpy as np
import argparse
import glob
import PIL
import os
import os.path
from PIL import Image
MODEL_PATH ='/content/the_model_sep.h5'
model = keras.models.load_model(MODEL_PATH)
indiceNo =0
indicePon =0
total=0
f ='/content/chest_xray/test/NORMAL'
for file in os.listdir(f):
  f_img = f+"/"+file
  img = image.load_img(f_img, target_size=(224, 224))
  img = np.asarray(img)
  img = np.expand_dims(img , axis =0)
  output = model.predict(img)
  if output [0][0] > output [0][1] :
    print("NO ", output [0])
    indiceNo += 1
    total += 1
```

```
  if output [0][1] > output [0][0]:
    print("YES ", output [0])
    indicePon += 1
    total += 1
print("====================")
print("NO pneumonia predict")
print("====================")
print("NO",indiceNo)
print("YES",indicePon)
```

Listing 4.10: Testning Model

4.4 Empirical evaluation and results

4.4.1 Results of the training phase

We created the following visualization to show the performance of our deep learning CNN during training:

- a graph of "accuracy" on the train "acc" dataset over the training epochs (Figure 4.2).

- a graph of "loss" on the train dataset "loss" over the training epochs (Figure 4.3).

- a graph of "val-accuracy" on the validation dataset 'Val acc' on the training epochs (Figure 4.2)

-a graph of "val_loss" on the "val_loss" validation dataset on training epochs (Figure 4.3).

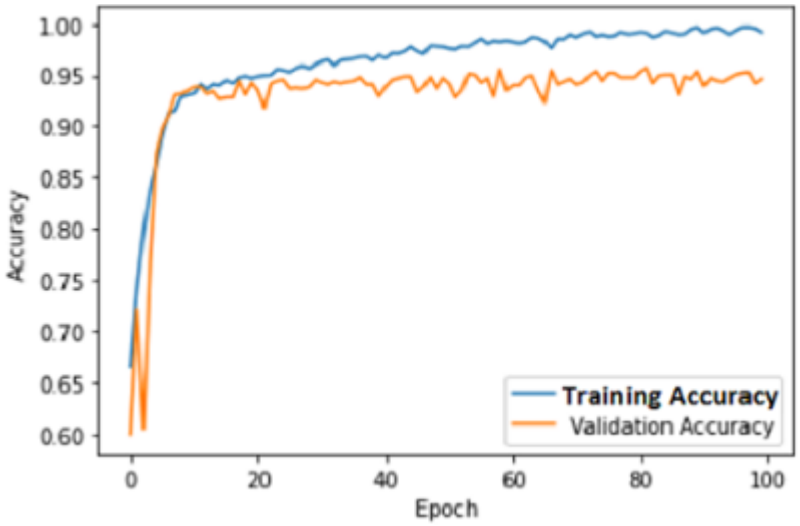


Figure 4.2: The model Accuracy.

When we trained the proposed model presented previously for 100 epochs using the data set presented in the second section. The training accuracy and validation accuracy reached values of 99.3% and 95.89%, respectively (see Figure 4.2).

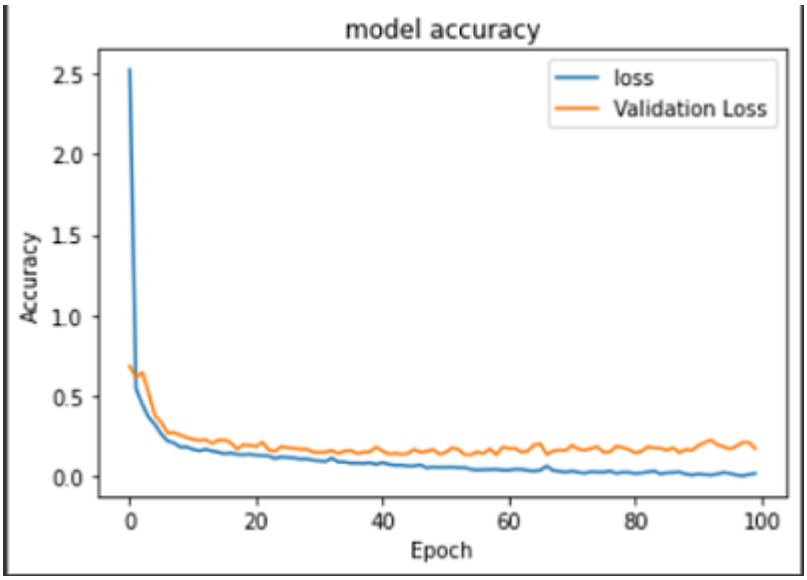


Figure 4.3: The model Loss.

The loss function during the training phase is depicted in Figure 4.3. Hence, the loss is decreased and reaches values of 0.016 and 0.12 in training phase and validation phase, respectively.

4.4.2 Model evaluation on test data

4.4.3 Testing with the first dataset

We utilize the third subset, which includes 587 images, in this phase to see how well our CNN model performs on new images from the Dataset. We have obtained: Accuracy test, Precision test and F1score equal to 97.10%, 97.12%, and 97.11% respectively (see Figure 4.4).

We also used The confusion matrix which is about a table that is often used to describe the performance of a classification model ,(see Figure 4.5) Row 0 of column 0 represents correct answers for Class 1 (Not Pneumonia): 152 out of 159 pictures. Row 1 of column 0 represents wrong answers for Class 1: 7 out of 159 pictures. Row 0 of column 1 represents wrong answers for Class 2 (yes Pneumonia): 10 out of 428 pictures. Row 1 of column 1 represents correct answers for Class 2: 418 out of 428 pictures.

```
==============TEST RESULTS============
Found 587 images belonging to 2 classes.
19/19 [==============================] - 1s 55ms/step
Accuracy  : 0.9710391822827938
Precision : 0.9712704828287552
f1Score : 0.9711235856257262
```

Figure 4.4: Model testing First Dataset.

The results of the matrix show that the model performs efficiently, as it was incorrect in just 7 out of 159 images in the first category and 10 out of 420 in the second.

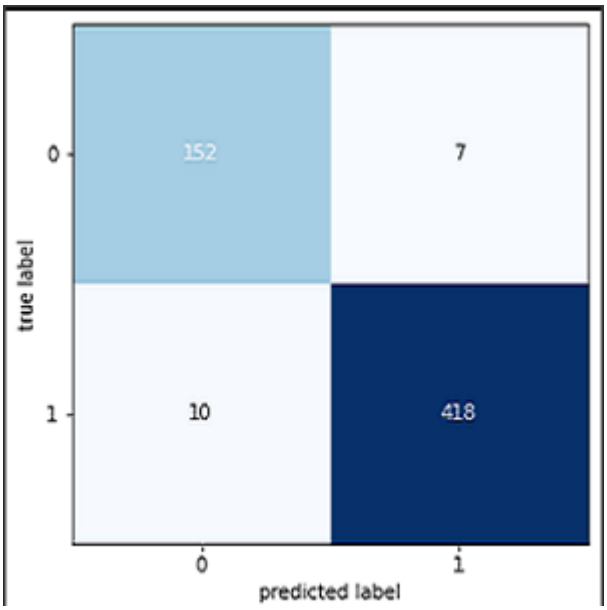


Figure 4.5: Confusion matrix First Dataset .

4.4.4 Test for second dataset

We utilize the third subset, which includes 624 images, in this phase to see how well our CNN model performs on new images from the Dataset. We have obtained : Accuracy test, Precision test and F1score equal to 93.26%, 93.84%, and 93.11% respectively (see Figure 4.6). We also used The confusion matrix, see Figure 4.7 Row 0 of column 0 represents correct answers for Class 1 (Not Pneumonia): 191 out of 234 pictures. Row 1 of column 0 represents wrong answers for Class 1: 41 out of 234 pictures. Row 0 of column 1 represents wrong answers for Class 2 (yes

```
==============TEST RESULTS============
Found 624 images belonging to 2 classes.
20/20 [==============================] - 8s 290ms/step
Accuracy  : 0.9326923076923077
Precision : 0.9384739870534643
f1Score : 0.9311887394574879
```

Figure 4.6: Model testing Second Dataset.

Pneumonia): 1 out of 390 pictures. Row 1 of column 1 represents correct answers for Class 2: 389 out of 390 pictures. The results of the matrix show that the model performs efficiently, as it was incorrect in just 41 out of 234 images in the first category and 1 out of 390 in the second.

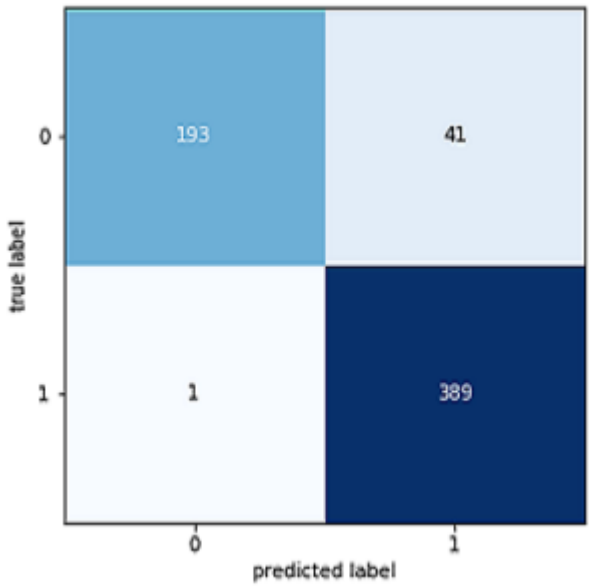


Figure 4.7: Confusion matrix Second Dataset.

4.4.5 Table of Comparison

Our work can be objectively compared with four works which have developed deep learning model and used the same mentioned data set which are ([20]) , ([33]),([43]) ([45]) see Table 4.9.

	Accuracy	Validation accuracy	Precision	Test Accuracy
Our architecture	99.3%	95.89%	99%	97.1%
([20]) their model	95.38%	95%	87.64%	Nope
([20]) Vgg-16	94.79%	Nope	85%	Nope
([20]) ResNet50	94.27%	Nope	83%	Nope
([20]) MobileNet	94.53%	Nope	87%	Nope
([33]) architecture 1	85.26%	Nope	75.00 %	Nope
([33]) architecture 2	92.31%	Nope	87.00 %	Nope
([43]) NasNetLarge	Nope	96.50%	Nope	95.35%
([43]) Xception	Nope	99.00%	Nope	93.59%
([45]) they model	Nope	Nope	Nope	95.30%

Table 4.9: COMPARATIVE TABLE

4.5 Conclusion

In this chapter we have mentioned the tools, libraries and the frameworks we have utilized in our study. We also discussed our CNN model and provided the execution of a large portion of our system and presented the results we obtained.

Chapter 5

Conclusion and Perspectives

5.1 Conclusion

Deep learning applications has achieved great success in recent years, and this success included many uses, most notably in the field of health care, where we find many works such as following up patients and detecting diseases that are difficult for doctors to diagnose, and it has also achieved other successes in this field.

Pneumonia is a serious disease for humans as it leads to the lost of lives of many people around the world, especially the elderly and young children between two and five years old (two children die per minute due to this disease, according to [13] statistics).

Early diagnosis of this disease is an important factor in the success of the treatment process. Therefor, we developed a CNN deep learning model for early diagnosis of children's inflammatory vision disease. We used one data set [2] for training and two data sets for experiment.

At the end, we obtained a CNN model that is able to detect pneumonia in children, and the results were satisfactory

5.2 Perspectives

However, several aspects of this study might be improved, which we see as promising for our work.

-Make our model able to differentiate between bacterial and viral pneumonia.

-Collect a local dataset to train, validate, and test our suggested architecture.

-Creating other CNN models for diagnosing other lung diseases.

References

[1] Convolutional neural network (cnn). https://developer.nvidia.com/discover/convolutional-neural-network.

[2] dataset of pediatric pneumonia chest x-ray. https://www.kaggle.com/andrewmvd/pediatric-pneumonia-chest-xray. Accessed: 2022-01-09.

[3] Découvrez les différentes couches d'un cnn. https://openclassrooms.com. Accessed: 2022-02-09.

[4] Fonction d'activation. https://deeplylearning.fr/cours-theoriques-deep-learning/fonction-dactivation/. Accessed: 2022-02-20.

[5] health-care. https://marketbusinessnews.com/financial-glossary/health-care/. Accessed: 2022-02-20.

[6] inside-machinelearning. httpshttps://inside-machinelearning.com/fonction-dactivation-comment-ca-marche-une-explication-simple/. Accessed: 2022-01-19.

[7] intelligence-artificielle-dans-la-sante. https://mbamci.com/intelligence-artificielle-dans-la-sante.

[8] Kaggle. https://www.kaggle.com/. Accessed: 2022-05-20.

[9] levels-of-healthcare. https://manzilhealth.com/levels-of-healthcare/, note = Accessed: 2022-03-10.

[10] The lungs. https://teachmeanatomy.info/thorax/organs/lungs/#Lung_Structure., note = Accessed: 2022-02-09.

[11] Neural networks. https://www.ibm.com/cloud/learn/neural-networks. Accessed: 2022-04-15.

[12] Numpy. https://numpy.org/doc/stable/user/whatisnumpy.html.

[13] pneumonia disease. https://https://data.unicef.org/topic/child-health/pneumonia/. Accessed: 2022-04-20.

[14] Radiographie pulmonaire. https://testhttps://www.acrim.fr/nos-examens/radiologie/pulmonaire.

[15] transfer-learning. https://builtin.com/data-science/transfer-learning. Accessed: 2022-01-19.

[16] Why tensorflow. https://www.tensorflow.org/?hl=fr.

[17] Michael S. Niederman Rosario Menéndez James D. Chalmers Richard G. Wunderink Antoni Torres, Catia Cilloniz and Tom van der Poll. Pneumonia. *PRIMER*, 2021.

[18] Amalie Hindsholm Ian Law Liselotte Højgaard Claes Nøhr Ladefoged, Lisbeth Marner and Flemming Littrup Andersen. Deep learning based attenuation correction of pet/mri in pediatric brain tumor patients: Evaluation in a clinical setting. *ORIGINAL RESEARCH*, 2019.

[19] Ling Dai, Liang Wu, Huating Li, Chun Cai, Qiang Wu, Hongyu Kong, Ruhan Liu, Xiangning Wang, Xuhong Hou, Yuexing Liu, et al. A deep learning system for detecting diabetic retinopathy across the disease spectrum. *Nature communications*, 12(1):1–11, 2021.

[20] Yushuang Li Dejun Zhang, Fuquan Ren. Pneumonia detection from chest x-ray images based on convolutional neural network. *electronics*, 2021.

[21] Khalid El Asnaoui. Design ensemble deep learning model for pneumonia disease classification. *International Journal of Multimedia Information Retrieval*, 10(1):55–68, 2021.

[22] Khalid El Asnaoui, Youness Chawki, and Ali Idri. Automated methods for detection and classification pneumonia based on x-ray images using deep learning. In *Artificial Intelligence and Blockchain for Future Cybersecurity Applications*, pages 257–284. Springer, 2021.

[23] expert.a. machine-learning-definition, 2022.

[24] Google. Colaboratory, 2022.

[25] Digital Guide. Keras, 2022.

[26] Guoyao Hao and Yifei Li. Bone age estimation with x-ray images based on efficientnet pretraining model. In *Journal of Physics: Conference Series*, volume 1827, page 012082. IOP Publishing, 2021.

[27] Jin He and Dan Jiang. Fully automatic model based on se-resnet for bone age assessment. *IEEE Access*, 9:62460–62466, 2021.

[28] An Hu and Navid Razmjooy. Brain tumor diagnosis based on metaheuristics and deep learning. *International Journal of Imaging Systems and Technology*, 31(2):657–669, 2021.

[29] Yoshua Bengio Ian Goodfellow and Aaron Courville. Deep learning. *MIT press*, 2016.

[30] Simukayi Mutasa Derek Merck David W.Swenson Rama S.Ayyala Ian Pan, Grayson L.Baird. Rethinking greulich and pyle: A deep learnin approach to pediatric bone age assessment using pediatric trauma hand radiographs. *ORIGINAL RESEARCH*, 2020.

[31] Abdullahi Umar Ibrahim, Mehmet Ozsoz, Sertan Serte, Fadi Al-Turjman, and Polycarp Shizawaliyi Yakoi. Pneumonia classification using deep learning from chest x-ray images during covid-19. *Cognitive Computation*, pages 1–13, 2021.

[32] Remaigui Ibrahim, Kahloul Laid, and Benhrazallah Saber. A new deep learning architecture for pneumonia detection in pediatrics. In *International Multi-Conference on Systems, Signals and Devices, May 06-10, Setif, Algeria. (accepted and presented)*. IEEE, 2022.

[33] Kataria. G Kaushik V.S Hemanth. D.J Jain .R, Nagrath. P. Pneumonia detection in chest x-ray images using convolutional neural networks and transfer learning. *Measurment*, 2020.

[34] Daniel Kermany, Kang Zhang, Michael Goldbaum, et al. Labeled optical coherence tomography (oct) and chest x-ray images for classification. *Mendeley data*, 2(2), 2018.

[35] SanaUllah Khan, Naveed Islam, Zahoor Jan, Ikram Ud Din, and Joel JP C Rodrigues. A novel deep learning based framework for the detection and classification of breast cancer using transfer learning. *Pattern Recognition Letters*, 125:1–6, 2019.

[36] Marco La Salvia, Gianmarco Secco, Emanuele Torti, Giordana Florimbi, Luca Guido, Paolo Lago, Francesco Salinaro, Stefano Perlini, and Francesco Leporati. Deep learning and lung ultrasound for covid-19 pneumonia detection and severity classification. *Computers in Biology and Medicine*, 136:104742, 2021.

[37] Joseph H. Rothstein Eugene Fluder Russell McBride Li Shen, Laurie R. Margolies. Deep learning to improve breast cancer detection on screening mammography. *Weiva Sieh*, 2019.

[38] William Lotter, Abdul Rahman Diab, Bryan Haslam, Jiye G Kim, Giorgia Grisot, Eric Wu, Kevin Wu, Jorge Onieva Onieva, Yun Boyer, Jerrold L Boxerman, et al. Robust breast cancer detection in mammography and digital breast tomosynthesis using an annotation-efficient deep learning approach. *Nature Medicine*, 27(2):244–249, 2021.

[39] A. Lundervold and Ar. *Zeitschrift fur Medizinische Physik*, 2018.

[40] Ravi Manne. *Current Journal of Applied Science and Technology*, 2021.

[41] Mahsa GHADERI Raheleh KAFIYEH Mehdi Torabian, ESFAHANI. Classification of diabetic and normal fundus images using new deep learning method. *Leonardo Electronic Journal of Practices and Technologies*, 2018.

[42] PAUL MOONEY. Chest x-ray images (pneumonia), 2018.

[43] Domingos Bruno Sousa Santos Nator Junior Carvalho Costa, Jose Vigno Moura Sousa. Classification of x-ray images for detection of childhood pneumonia using pre-trained neural networks. *Revista Brasileira de Computação Aplicada*, 2020.

[44] Python. What is python? executive summary, 2022. Accessed: 2022-02-09.

[45] Arata Andrade Saraiva, Nuno M Fonseca Ferreira, Luciano Lopes de Sousa, Nator Junior C Costa, José Vigno M Sousa, DBS Santos, Antonio Valente, and Salviano Soares. Classification of images of childhood pneumonia using convolutional neural networks. In *BIOIMAGING*, pages 112–119, 2019.

[46] Riad Sebti, Siham Zroug, Laid Kahloul, and Saber Benharzallah. A deep learning approach for the diabetic retinopathy detection. In *The Sixth Smart City Applications Iinternational Conference*, Accepted. Springer, 2021.

[47] Muhammad Imran Sharif, Muhammad Attique Khan, Musaed Alhussein, Khursheed Aurangzeb, and Mudassar Raza. A decision support system for multimodal brain tumor classification using deep learning. *Complex & Intelligent Systems*, pages 1–14, 2021.

[48] Techopedia. Matplotlib, 2022.

[49] Nikos Tsiknakis, Dimitris Theodoropoulos, Georgios Manikis, Emmanouil Ktistakis, Ourania Boutsora, Alexa Berto, Fabio Scarpa, Alberto Scarpa, Dimitrios I Fotiadis, and Kostas Marias. Deep learning for diabetic retinopathy detection and classification based on fundus images: A review. *Computers in Biology and Medicine*, page 104599, 2021.

[50] Dayong Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, and Andrew H Beck. Deep learning for identifying metastatic breast cancer. *arXiv preprint arXiv:1606.05718*, 2016.

[51] Keping Yu, Liang Tan, Long Lin, Xiaofan Cheng, Zhang Yi, and Takuro Sato. Deep-learning-empowered breast cancer auxiliary diagnosis for 5gb remote e-health. *IEEE Wireless Communications*, 28(3):54–61, 2021.

[52] Xian-Da Zhang. Machine learning. In *A Matrix Algebra Approach to Artificial Intelligence*, pages 223–440. Springer, 2020.