REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

A NOVEL DEEP LEARNING FRAMEWORK ENHANCED BY HYBRID OPTIMIZATION USING DUNG BEETLE AND FICK'S LAW FOR SUPERIOR PNEUMONIA DETECTION

Abdulazeez Mohammed Ibrahim SABAAWI

Master's Thesis

Supervisor

Asst. Prof. Dr. Hakan KOYUNCU

İstanbul, 2024

A NOVEL DEEP LEARNING FRAMEWORK ENHANCED BY HYBRID OPTIMIZATION USING DUNG BEETLE AND FICK'S LAW FOR SUPERIOR PNEUMONIA DETECTION

Abdulazeez Mohammed Ibrahim SABAAWİ

Information Technologies

Master's Thesis

ALTINBAŞ UNIVERSITY

2024

The thesis titled A NOVEL DEEP LEARNING FRAMEWORK ENHANCED BY HYBRID OPTIMIZATION USING DUNG BEETLE AND FICK'S LAW FOR SUPERIOR PNEUMONIA DETECTION prepared by ABDULAZEEZ SABAAWI and submitted on 09/12/2024 has been **accepted unanimously** for the degree of Master of Science in Information Technologies.

_____

Asst. Prof. Dr. Hakan KOYUNCU

Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Hakan KOYUNCU	Department of Computer Engineering, Altınbaş University	_____
Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	Department of Computer Engineering, Altınbaş University	_____
Asst. Prof. Dr. Warish PATEL	Department of Computer Engineering, Parul University	_____

I hereby declare that this thesis meets all format and submission requirements for a Master's thesis.

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Abdulazeez SABAAWI

Signature

DEDICATION

I wish to extend my profound appreciation to my supervisor, Assist. Prof. Dr. Hakan Koyuncu. His unwavering support and invaluable guidance have been the bedrock of this research journey, offering insights, instructions, and precise information. Gratitude also flows to the other educators who executed their academic duties impeccably. Moreover, heartfelt thanks embrace all those who bolstered me with encouragement, wise counsel, and invaluable information, nurturing this endeavour with their unwavering support. I dedicate the fruit of my humble efforts to my beloved mother, the inexhaustible pillar of my spirit and the warm embrace that cradled my dreams. Without her, I wouldn't be who I am today. To my dear father, my support after God, who dedicated his life to pave the way for my success. He has been the best supporter and encourager. To all my dear family members whose mention would exceed these lines, I extend my heartfelt gratitude for your unwavering love and support. To everyone who offered their assistance and encouragement throughout my journey, I express my sincere appreciation and profound thanks.

ABSTRACT

A NOVEL DEEP LEARNING FRAMEWORK ENHANCED BY HYBRID OPTIMIZATION USING DUNG BEETLE AND FICK'S LAW FOR SUPERIOR PNEUMONIA DETECTION

SABAAWI, Abdulazeez Mohammed Ibrahim

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Asst. Prof. Dr. Hakan KOYUNCU

Date: 12/2024

Pages: 93

Pneumonia is Pneumonia, a leading cause of global mortality, particularly among children and the elderly, is characterized by inflammation of the air sacs in the lungs. Early and accurate detection is crucial to mitigating its spread and improving patient outcomes. Traditional diagnostic methods, including physical exams, blood tests, and chest X-ray analysis by radiologists, are effective but often time-intensive, resource-heavy, and prone to human error. These limitations underscore the need for advanced, efficient diagnostic solutions. This study leverages deep learning techniques, specifically Convolutional Neural Networks (CNN) and MobileNet architectures, to enhance pneumonia detection using chest X-ray images. Deep learning excels in image recognition tasks through automatic feature extraction, making it a powerful tool for medical diagnostics. The research applies these models to a large dataset of chest X-rays labeled as NORMAL or PNEUMONIA, utilizing rigorous preprocessing techniques, including normalization and augmentation, to improve data quality and diversity. Hybrid optimization techniques inspired by the dung beetle optimizer and Fick's law of diffusion are employed to optimise model performance, effectively navigating complex parameter spaces. Evaluation metrics such as accuracy, precision, recall, and F1-score validate the models. Results demonstrate the enhanced MobileNet model's superior performance, achieving an accuracy of 98.19%, significantly outperforming the CNN baseline. This highlights the potential of deep learning and

innovative optimization methods in advancing pneumonia diagnostics. The findings emphasize the broader implications of artificial intelligence in healthcare, offering a pathway for developing robust diagnostic tools that expedite accurate and timely disease detection. This research contributes to alleviating the global pneumonia burden and underscores the need for continued innovation in machine learning to enhance medical outcomes.

Keywords: X-ray Image Classification, Deep Learning, Convolutional Neural Networks, Machine Learning in Healthcare, Optimization Algorithms.

ÖZET

ÜSTÜN PNÖMONİ TESPİTİ İÇİN BOK BÖCEĞİ VE FICK YASASI KULLANILARAK HİBRİT OPTİMİZASYONLA GELİŞTİRİLMİŞ YENİ BİR DERİN ÖĞRENME ÇERÇEVESİ

SABAAWI, Abdulazeez Mohammed Ibrahim

Yüksek Lisans, Bilişim Teknolojileri, Altınbaş Üniversitesi

Danışman: Doç. Dr. Hakan KOYUNCU

Tarih: 12/2024

Sayfa: 93

Pnömoni, özellikle çocuklar ve yaşlılar arasında, küresel mortalitenin önde gelen nedenlerinden biridir ve akciğerlerdeki hava keseciklerinin iltihaplanması ile karakterize edilir. Hastalığın yayılmasını önlemek ve hasta sonuçlarını iyileştirmek için erken ve doğru tespit kritik öneme sahiptir. Fiziksel muayeneler, kan testleri ve radyologlar tarafından gerçekleştirilen göğüs röntgeni analizlerini içeren geleneksel tanı yöntemleri etkili olmakla birlikte, genellikle zaman alıcı, kaynak yoğun ve insan hatalarına açık yöntemlerdir. Bu sınırlamalar, gelişmiş ve verimli tanı çözümlerine olan ihtiyacı vurgulamaktadır. Bu çalışma, göğüs röntgeni görüntülerini kullanarak pnömoni tespitini geliştirmek için Derin Öğrenme tekniklerini, özellikle Evrişimsel Sinir Ağları (CNN) ve MobileNet mimarilerini kullanmaktadır. Derin öğrenme, veri üzerinden otomatik özellik çıkarımı yoluyla görüntü tanıma görevlerinde üstün performans gösterdiği için tıbbi tanılarda güçlü bir araç olarak kabul edilmektedir. Araştırma, NORMAL veya PNEUMONIA olarak etiketlenmiş büyük bir göğüs röntgeni veri setine bu modelleri uygulamakta ve veri kalitesini ve çeşitliliğini artırmak için normalizasyon ve veri artırmayı içeren titiz ön işleme tekniklerini kullanmaktadır.

Model performansını optimize etmek için, dung beetle optimizasyonu ve Fick'in difüzyon yasasından esinlenen hibrit optimizasyon teknikleri uygulanmıştır ve bu teknikler karmaşık parametre alanlarında etkili bir şekilde gezinmektedir. Doğruluk, kesinlik, geri çağırma ve

F1 skoru gibi değerlendirme metrikleri modellerin doğruluğunu kanıtlamaktadır. Sonuçlar, MobileNet modelinin üstün performans göstererek %98,19 doğruluk oranına ulaştığını ve CNN taban çizgisini önemli ölçüde geride bıraktığını göstermektedir. Bu, pnömoni tanısında derin öğrenme ve yenilikçi optimizasyon yöntemlerinin potansiyelini vurgulamaktadır. Bulgular, yapay zekânın sağlık alanındaki daha geniş etkilerini vurgulamakta ve doğru ve zamanında hastalık tespiti sağlayan güçlü tanı araçlarının geliştirilmesi için bir yol sunmaktadır. Bu araştırma, küresel pnömoni yükünün hafifletilmesine katkıda bulunmakta ve tıbbi sonuçları iyileştirmek için makine öğreniminde yeniliğin devam etmesi gerektiğinin altını çizmektedir.

Anahtar Kelimeler: Evrişimli Sinir Ağları, Derin Öğrenme, Röntgen Görüntüsü Sınıflandırması, Sağlıkta Makine Öğrenmesi, Optimizasyon Algoritmaları.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

AI	:	Artificial Intelligence
ANN	:	Artificial Neural Networks
CAD	:	Computer-aided diagnostics
CNN	:	Convolutional Neural Networks
CT	:	Computer Tomography
CXR	:	Chest X-Ray
DBO	:	Dung Beetle Optimizer
DL	:	Deep Learning
DNNs	:	Deep Neural Networks
EDA	:	Exploratory Data Analysis
FLA	:	Fick's Law Algorithm
HGS	:	Hunger Games Search
HHO	:	Harris Hawks Optimization
ICU	:	Intensive Care Unit
IU-X-Ray	:	Indiana University Chest X-Ray
KNN	:	K-Nearest Neighbors
MAs	:	Metaheuristic Algorithms
MeSH	:	Medical Subject Headings
ML	:	Machine Learning
MRIs	:	Magnetic Resonance Imaging

NLM	:	National Library of Medicine
PET	:	Positron Emission Tomography
ReLU	:	Rectified Linear Unit
RF	:	Random Forest
RT-PCR	:	Reverse Transcription-polymerase Chain Reaction
SIAs	:	Swarm Intelligence Algorithms
SMA	:	Slime Mould Algorithm
SVM	:	Support Vector Machines
TL	:	Transfer Learning
US	:	Ultrasound

1. INTRODUCTION

1.1 INTRODUCTION

Pathogenic organisms, chemicals, physical conditions, and pharmaceuticals produce pneumonia, which infects the lung parenchyma [1]. Non-infectious and infectious pneumonia exist. Viruses, bacteria, mycoplasmas, and immunological reactions can cause infectious pneumonia, while radiation, chemical, and physical factors can induce non-infectious pneumonia [2]. Computed tomography (CT), chest X-ray (CXR) imaging, and magnetic resonance imaging (MRI) provide lung disease diagnosis. CXR imaging is portable, cost-effective, ubiquitous in clinics, and radiation-efficient [3]. CXR pictures often show comparable locations for pneumonia and lung tumours, making pneumonia identification difficult even for experienced doctors; even experienced clinicians have trouble recognizing pneumonia in CXR pictures because they typically show identical locations for lung malignancies [4]. Traditional pneumonia diagnosis methods are time-consuming and inefficient, making it hard to diagnose patients.

X-ray imaging is often preferred over CT imaging, requiring less time, and high-quality CT scanners are not readily available in many underdeveloped regions [5]. In contrast, X-rays are widely accessible and crucial in epidemiological studies and clinical care. Many parts of the world suffer from a shortage of radiologists and skilled healthcare personnel, making accurate disease estimation challenging. Computer-aided diagnosis (CAD) using artificial intelligence (AI) has become increasingly popular as a solution to this issue [6]. A further complication in diagnosing this disease is that its defining features often overlap with those of other diseases, making accurate diagnosis difficult for medical experts [7]. The literature has examined several machine learning (ML) methods for pneumonia detection in chest X-rays (CXRs); however, they have drawbacks. They use handcrafted feature engineering, which is time-consuming and requires expertise [8], [9]. Complex and subtle patterns might be difficult to discern, reducing performance, especially with difficult medical imaging. Traditional ML models like support vector machines (SVM) and random forest (RF) may not adapt to CXRs' unique and dynamic properties, so reengineering is needed constantly to accept new datasets or imaging methods. Due to this inflexibility, they may struggle to keep up with medical imaging advances in real-time clinical settings.

Deep learning (DL) approaches address these problems, often matching or surpassing the accuracy of average radiologists in predicting diseases. DL is now a widely used algorithm for developing medical image classification tasks [10]. Furthermore, DL methods have demonstrated superior performance compared to traditional methods when using CXRs from pneumonia patients [11]. DL models have shown strong predictive capabilities, often outperforming doctors. DL techniques are applied to various tasks using CXR, including COVID-19 detection, tuberculosis detection, large-scale recognition, radiograph classification, and tuberculosis segmentation [12]. In response to the current global crisis, various models have been developed for COVID-19, with metaheuristic algorithms (MAs) standing out as particularly significant [13],[14],[15],[16]. One of the key elements in applying these models for predicting COVID-19 outcomes is selecting the most impactful features for diagnosis. Feature selection techniques are generally classified into three main types: wrapper methods [17], embedded methods [18], and filter methods [19],[20]. Among these, the wrapper method is frequently employed with MAs for feature selection, as it evaluates feature subsets based on classifier error rates. The method then provides results based on the classifier and the algorithm used. In classification tasks, training samples map conditions and decision labels.

Ghosh et al. [21] explored a feature selection classification model based on fundamental KNN to reduce the dimensionality of hyperspectral image data. To address this issue, Zhang [22] proposed a feature selection method that integrates KNN with self-adaptive differential evolution. Similarly, Mohamed et al. [23] developed a feature selector to handle large data dimensionality and work with KNN to identify the most relevant feature combinations. This selector draws inspiration from the evolutionary interactions among predators, parasites, and hosts in nature. Numerous deterministic and evolutionary optimization techniques have been developed to address a range of optimization problems, whether single, multiple, or numerous objectives [24],[22]. Metaheuristic algorithms (MAs) have proven effective in solving various optimization challenges [25],[26],[27]. Unlike traditional gradient-based methods, often viewed as black boxes, MAs offer greater versatility and can sometimes be more efficient [28],[29]. Among the notable MAs are swarm intelligence algorithms (SIAs), which are particularly effective for real-world and large-scale problems [29]. This has led to significant scholarly interest and development. Notable SIAs include the Slime Mould Algorithm (SMA) [30], Hunger Games Search (HGS) [31], Harris Hawks Optimization

(HHO) [32], and Runge-Kutta Optimization Algorithm (RUN) [33]. These methods have demonstrated exceptional performance across various fields [34][35].

1.2 PROBLEM STATEMENT

Pneumonia diagnosis from X-ray images is considered a challenging task in medical imaging due to the subtleness and often overlapping visual appearance of pneumonia with other lung pathologies such as tuberculosis, lung cancer, and pleural effusion. Traditional diagnosis is usually performed by the expert eyes of radiologists interpreting chest radiographs manually. However, this practice is very time-consuming and prone to human error, especially if the abnormalities are vague or different from the typical ones. AI would better solve this problem by using advanced ML algorithms, especially DL techniques, that automatically and precisely detect pneumonia on X-ray images. AI-driven models hold much promise in learning complex patterns in medical images, which are invisible to the human naked eye. These CNN models have proved to be of higher precision in diagnosing and, simultaneously, reducing the workload of doctors. Therefore, deep learning algorithms combined with the metaheuristic approach might be most suitable for enhancing health imaging in terms of accuracy and speed regarding pneumonia detection and its distinction from other endocrine diseases. However, several challenges still limit the deployment of AI systems in real-world clinical settings. These challenges include the need for large, well-annotated datasets, difficulty ensuring that models generalize well across different populations and varied imaging conditions, and potential biases in AI algorithms due to unbalanced or nonrepresentative data. There are also ethical considerations about the responsibility of AI systems for decisions made in clinical applications, particularly in high-stakes environments such as emergency care. Ongoing research is directed toward addressing these challenges by enhancing the interpretability and transparency of AI models, developing standardized protocols for model validation and deployment, and ensuring robust performance across diverse healthcare environments. Thus, the research problem lies not only in enhancing the technical capabilities of AI for pneumonia diagnosis but also in addressing the broader implications of integrating AI into clinical workflows, ensuring that these systems are reliable, fair, and clinically useful.

1.3 RESEARCH QUESTIONS

a. How can the optimization of CNNs help their performance in correctly classifying healthy and pneumonia through chest X-ray images?

This question investigates how optimal CNN architectures and training methodologies can be leveraged to minimise false positives in the diagnosis of anomalies of pneumonia from medical imaging.

b. What is the function of the metaheuristic Algorithm in feature selection for improving CXR images-based machine learning classifier performance on pneumonia diagnosis?

This question researches the efficacy of metaheuristic algorithms for extracting important features from radiological images to improve the predicting accuracy of machine learning (ML) models.

1.4 RESEARCH OBJECTIVE

The primary goal of this thesis is to design and investigate a current benchmark diagnostic model employing deep learning and machine learning approaches in the early discovery of pneumonia using chest X-ray images. Further details state that the research has endeavoured to make pneumonia differentiation from other respiratory diseases more efficient by optimizing convolutional neural networks (CNNs) and feature selection metaheuristic algorithms. In addition, the thesis aims to apply and examine data augmentation methods for better model resilience and generalization. The paper presented in this study will try to provide the design and modelling of a reliable automated tool that may support healthcare professionals in preparing these appropriate solutions against the spontaneous treatment process and enhance clinical decision-making for patient outcomes concerning the pneumonia pandemic.

1.5 THESIS STRUCTURE

Key chapters in this thesis are organized to investigate and respond to the research purposes systematically.

a. A literature review on previous and current diagnostic techniques, DL methods, and ML in medical image analysis will be conducted. This will analyze what has been accomplished to date while revealing gaps within this research that need to be addressed.

b. The methodology chapter explains the research design of the intended model, which includes data gathering and preprocessing steps; it also illustrates various CNN structures with elements for feature selection through metaheuristic algorithms.

c. The results and discussion chapter illustrates the testing results, includes an analysis involving model performance metrics, and compares to existing works.

d. This final chapter on conclusion and future work will detail the research contributions made by this thesis, provide a detailed analysis of how these findings can be used in real-world clinical practice applications to solve challenges both currently present and novel arising problems and discuss further avenues to explore that could improve the diagnostic capabilities of our model.

2. LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter, we will deal with the important areas of radiology and artificial intelligence that serve as an introduction to advanced computation techniques employed in medical images. The chapter starts with an introduction to radiology. It gives important information on how a radiological text should be read up and the interpretation of Chest X-ray (CXR) findings. After this, the chapter moves into artificial intelligence (AI), defining AI and how it is increasingly used in healthcare. At this core is an analysis of CNN that are crucial for processing images in a compute-efficient framework called MobileNet architecture. The chapter also briefed DBO and FLO Optimization, revealing relevance with the contrastive orthodox in optimization and promising aspects to enrich these computational models. The chapter finishes with a grand explicit literature review, framing the introduction of this research by giving it a wider background.

2.2 RADIOLOGY

Radiology, a major specialized field in medicine, contains two primary categories: diagnostic and interventional [36]. They analyze imaging tests, like MRIs and X-rays, made by other doctors to assist and guide a patient's care. Interventional radiologists, meanwhile, execute procedural interventions while imaging. Now we have a problem in that radiological images are evaluated on the fly according to how fast, tired and diverse in experience one of us is. The heavy training cost has created a demand-supply gap for certified radiologists, and as such, several healthcare institutions have outsourced medical image analysis tasks to third-party providers, including teleradiology companies, especially from countries like India [37]. Moreover, diagnostic delays or errors may result in patient discomfort and complications. To overcome this, there is extensive demand for automated deep learning (DL) architectures for the precise and time-efficient interpretation of these images. An illustration of a radiology report from the Indiana University X-ray dataset with related pictures [38]. It is represented in Figure 2.1.

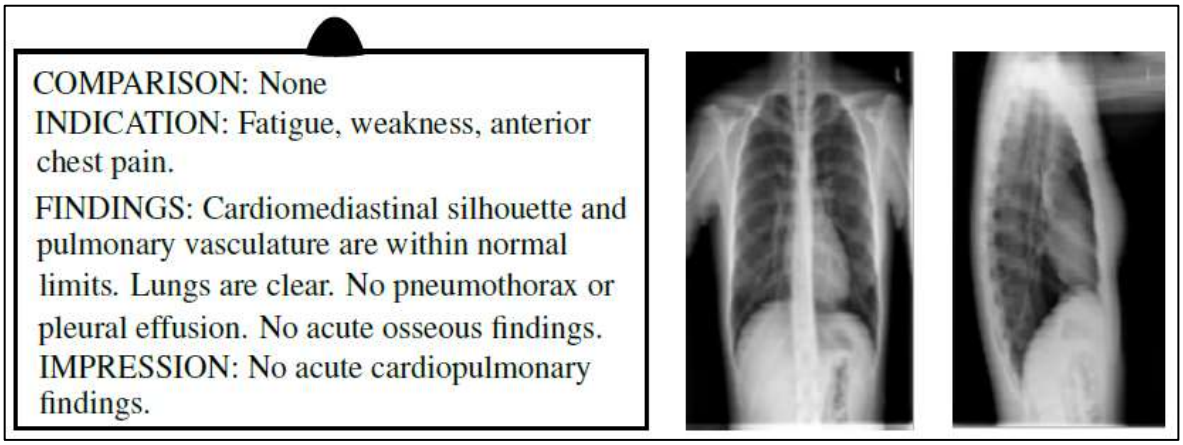


Figure 2.1: An Illustration of a Radiology Report Accompanied by the Corresponding Images [38].

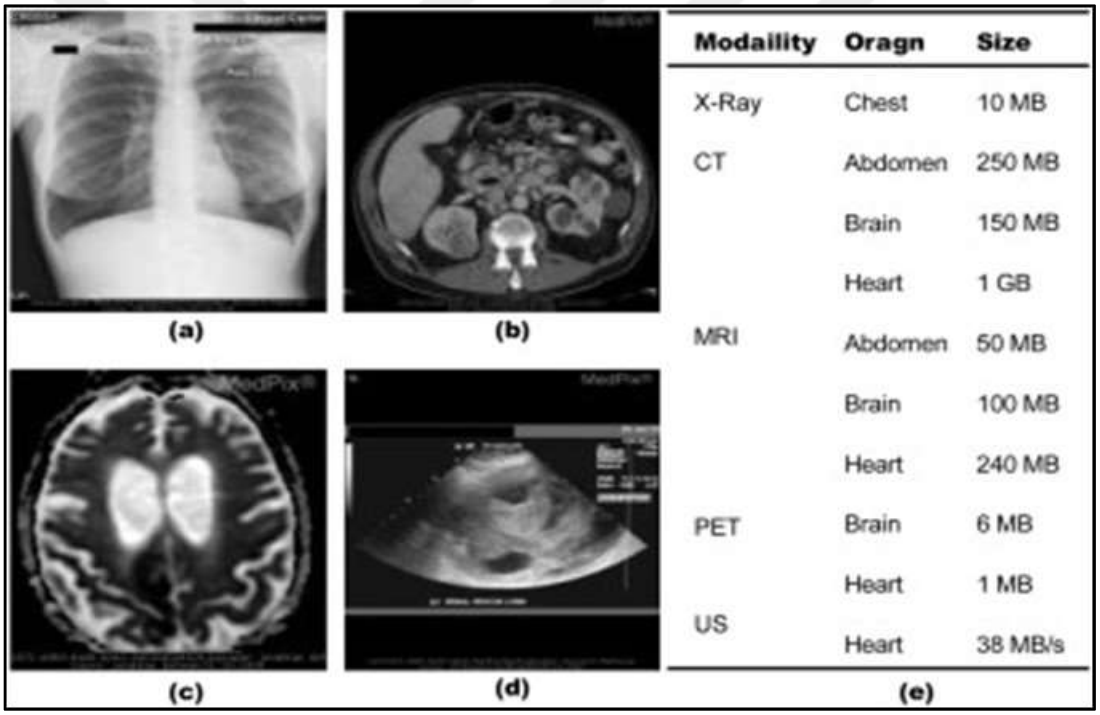
2.2.1 Radiology Text

Radiology reports are text documents that are written by radiologists to record the patient's prior medical history and symptoms as well as his interpretation of relevant Radiological images [39]. These reports are usually in a specified format with sections including comparison, indication reason for the exam, findings and impressions. Of particular importance is the impression section, which lists normal and abnormal findings by level of perceived significance [40]. In Figure 2.1, we show an instance from the dataset of Indiana University Chest X-Rays (IU X-Rays) [39], where for each report, there are two chest x-ray image archives that correspond to it. Radiologists use Metathesaurus [41], RadLex [42], and Medical Subject Headings [MeSH]. Over 100 controlled vocabulary systems provide One million biomedical terms and over five million concept names in the Metathesaurus. On the other side, RadLex covers radiology terminology, including imaging methods and equipment. The National Library of Medicine (NLM) developed MeSH, a complete controlled vocabulary for indexing scientific publications. Shin et al. [39] analysed IU X-ray results using MeSH terms [43]. Brain tumours and lung diseases use a semi-standardized descriptive system instead of a lexicon. Deep learning (DL) has demonstrated potential in generating radiology reports from images [40],[44],[45],[46]. Initially, researchers produced brief descriptive sentences based solely on image features, but later attempts aimed to create more detailed reports with multiple sentences, which introduced challenges related to content selection and sequencing. This approach can lead to reports that include information not directly observable from the images, such as the patient's nationality [39]. Despite these

advancements, current text-based DL algorithms still fall short, often lacking particular image labels.

2.2.2 Interpreting Radiology Images

Radiology includes X-ray, CT, MRI, Positron Emission Tomography (PET), and Ultrasound (US) [47]. Figure 2.2 shows these imaging methods and their properties. Chest radiography is the most common and requires precise and early interpretation to avert life-threatening diseases [48]. Managing over 100 chest X-rays (CXRs) daily is difficult for radiologists [49]. Italian and UK hospitals have used CXR to diagnose COVID-19 during the epidemic [50]. CXR is less sensitive than chest CT; however, portable radiology devices make documentation easier and reduce cross-infection.



Modaility	Oragn	Size
X-Ray	Chest	10 MB
CT	Abdomen	250 MB
	Brain	150 MB
	Heart	1 GB
MRI	Abdomen	50 MB
	Brain	100 MB
	Heart	240 MB
PET	Brain	6 MB
	Heart	1 MB
US	Heart	38 MB/s

Figure 2.2: Radiological Imaging Modalities and Their Characteristics: X-ray (a), CT (b), MRI (c), Ultrasound (US) (d), and Image Attributes (e) [38].

Recently, multiple large CXR datasets have been made accessible to academics to improve DL image analysis models. DL researchers are now more interested in CXR. Image Archiving and Communication Systems (PACSs) have been implemented in modern hospitals to store, manage, transmit, and process radiological pictures since the 1990s, making them crucial for image capture.

Image processing was standardised and simplified by DICOM in 1993. It is the most prevalent format file for medical imaging data like CXRs, CT scans, and MRI scans and offers extensive reporting and result features [51]. Many deep learning (DL) medical image classification models use the Joint Photographic Group (JPG) due to Compute Engine constraints. DL preprocesses radiology images differently due to CPU and memory constraints. CT and MRI scans are 3D, while X-rays are 2D. Complexity grows with 3D images, although 2D DL models perform better [52]. DL is also difficult to apply to X-ray, which are 2D representations of 3D bodies. DL algorithms may need to be changed to evaluate overlapping X-ray physiological characteristics [53]. DL algorithms, notably CNNs, can handle 2D and 3D images with little alterations despite these challenges. Radiology DL research continues. Medical image analysis using DL has proven successful in image classification, lesion diagnosis, segmentation, content-based picture retrieval, report preparation, and picture enhancement [54],[55]. NiftyNet5 [56] provides an Apache License-licensed framework with several medical imaging CNN algorithms for fast DL implementation. Polls have highlighted CNN's role in medical image analysis [37]. Biswas et al. [57] say cardiology, neurology, mammography, microscopic, dermatological, gastrointestinal, and pulmonary use DL models.

2.2.3 Interpreting Chest X-Ray (CXR) Findings

The thesis's chest X-ray (CXR) reports and images are categorised by heart, lung, and bone problems. The findings included cardiomegaly, consolidation, atelectasis, COVID-19-related changes, fractures, edema, an enlarged cardio-mediastinum, lung lesions, lung opacities, no significant abnormalities, pleural effusion, other pleural abnormalities, pneumonia, pneumothorax, and indications of medical support [58], [59]. Lung atelectasis occurs when part or all of a lung fails to expand [60]. Lung closure can result from respiratory, pleural fluid, or lymph node pressure on the bronchus. Intensive care CXR images show radio-opacity due to atelectasis [61]. Cardiomegaly occurs when the heart enlarges, This problem is often congenital and can be caused by HIV, diabetes, pregnancy, kidney ailments, heart valve difficulties, or thyroid issues [62]. The cardiothoracic ratio on a CXR is used to diagnose an enlarged heart that is more than 50% larger than the rib cage [58]. Consolidation occurs when fluid replaces air in a lung [63]. It causes lung tissue enlargement and stiffening and is important in numerous diseases, especially pneumonia.

Oedema causes breathing problems due to lung fluid collection [64]. Acute lung damage or congestive heart failure can cause lung opacities, CXRs show thicker bronchial walls and hazy blood vessel outlines. Oedema detection is crucial for congestive heart failure management [65]. An enlarged cardio mediastinum, often resulting from cardiomegaly, refers to an enlargement of the heart [66]. Early identification of this condition can aid in timely treatment, which may include medications, medical procedures, or surgery. Fractures are identified as irregularities in CXR images, such as rib fractures [67]. Rib fractures, common from various injuries, can lead to serious complications [68]. According to Livingston [69], In CXRs, almost half of the rib fractures are overlooked, contributing to higher morbidity and mortality rates. Lung lesions can signal severe conditions, including lung cancer, heart diseases, and respiratory disorders [70]. These lesions can vary from focal to diffuse or multifocal. Early detection of lung cancer, one of the main causes of cancer-related deaths among men, is crucial for improving patient outcomes.

2.3 ARTIFICIAL INTELLIGENCE DEFINITIONS AND APPLICATIONS IN HEALTHCARE

Artificial Intelligence (AI) refers to one of the fields of computer science focused on mimicking human cognitive processes to achieve accurate prediction or classification based on input data [71],[72]. As illustrated in Figure 2-3, AI encompasses various subsets, with ML and DL being particularly prominent in healthcare applications.

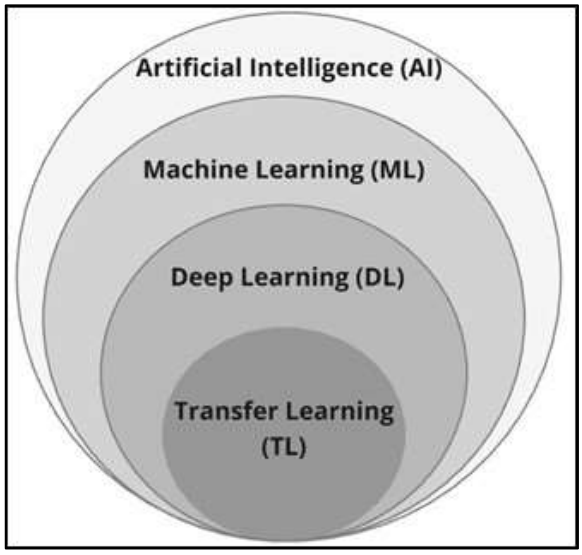


Figure 2.3: Subsets of Artificial Intelligence.

Artificial intelligence (AI) has made significant advancements in the past decade, demonstrating its superiority above traditional computing systems in various domains [73]. AI systems have been employed in the analysis of medical pictures and can automate jobs that were previously done by professionals, with a significantly reduced rate of failure. As a result, most research endeavours have been directed at public health issues since 2020. Consequently, numerous AI researchers began utilising non-invasive techniques such as X-ray scans to identify pneumonia cases. Several recent research have already employed their AI implementations to categorise pneumonia solely using medical imaging, demonstrating the soundness and precision of these techniques. Medical imaging techniques, such as X-rays, CT scans, and MRIs, can accurately identify the presence of pneumonia. X-rays are the prevailing and highly economical choice for automatic image analysis due to their widespread availability in several countries. However, it is important not to dismiss the possibility of exploring alternative screening methods. CT scans are crucial in the process of detecting pneumonia. In their study [74], examined the efficacy of CT scans in determining the severity of COVID-19 pneumonia in patients. The authors' findings indicate that patients with a high prevalence of acute pulmonary embolism are more prone to being brought to the ICU. Therefore, it is imperative to explore the use of CT scans. The healthcare sector is replete with state-of-the-art, visionary technology, which, when integrated with other methodologies, strives to transform the approach to patient care.

Machine Learning (ML), a branch of AI, involves the extraction of patterns and insights from data without explicit programming instructions [75]. Machine learning (ML) includes applying probabilistic and statistical models, machine learning, and deep learning techniques. It can be utilised to carry out various activities, such as categorisation, prediction, and decision-making. As shown in Figure 2.4, The machine learning method consists of two crucial phases: training and prediction. During the training phase, the chosen machine learning algorithm provides a constant flow of labelled data. In the prediction phase, the trained machine learning algorithm uses the learnt weights to estimate a value for unknown data. ML can be categorized into three main types: supervised learning, in which the machine learning algorithm is trained with a set of labelled data, which includes data points and their corresponding predicted outputs. This learning problem can be categorized into classification and regression [76]. In the classification process, the data is partitioned, sometimes in a balanced manner, into two or more distinct groups. The goal is to derive

meaningful insights from the labelled input data. However, regression refers to acquiring knowledge about continuous data, such as age. For instance, when considering a person's age, the input data could consist of their height and facial features, wherast, the output would be the precise age of that individual, expressed as a specific numerical value rather than a range or category. Supervised learning encompasses both multi-class and multi-label problems.

Unsupervised learning: in this form of learning, the chosen machine learning algorithm is trained on unlabeled data, which consists solely of data without any expected output or label. The goal is to discover patterns or correlations within the data that may be utilized to recognize learnt patterns in new, unseen data [77]; and reinforcement learning, The chosen machine learning algorithm is rewarded either positively or adversely depending on its interaction with a predetermined environment, as the name suggests. An illustration can be provided to demonstrate the proficiency of an AI driving agent, which is a machine learning algorithm, within a certain setting [78].

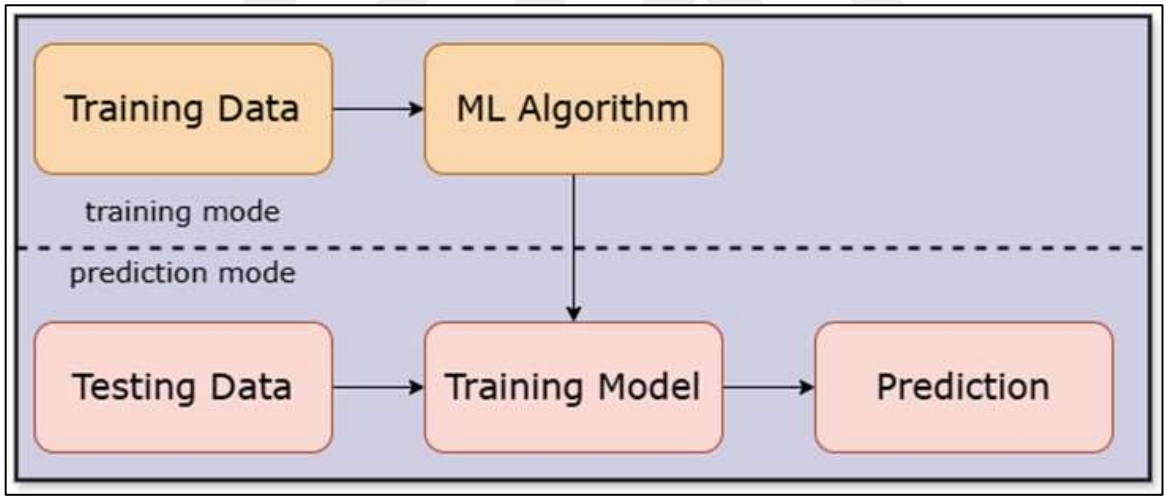


Figure 2.4: An Overview of the General Workflow of a Machine Learning Model.

DL, a subset of ML, employs structures inspired by human neurons known as Deep Neural Networks (DNNs) to learn from data [79]. These DNNs typically require large amounts of training data, which can be computationally intensive. The origins of deep learning may be traced back to the 1940s and 1950s when researchers initially began investigating the concept of artificial neural networks. Nevertheless, it was not until the late 2000s and early 2010s that deep learning began to receive broad acknowledgement, primarily because of the accessibility of high-performance computer resources and vast quantities of data [80]. Deep

learning is crucial as it enables machines to acquire proficiency in intricate jobs with exceptional precision, often surpassing conventional machine learning techniques. Deep learning has revolutionised picture categorization by enabling computers to autonomously identify objects and situations in photographs with exceptional precision [81]. Deep learning algorithms surpass traditional machine learning techniques in multiple aspects. One approach to automatically acquiring abstract features from unprocessed data inputs is constructing hierarchical representations of the data. Moreover, they become proficient with large datasets; hence, they can apply to tasks where large amounts of data are available. However, deep learning methods can be computationally time-consuming and may require a large amount of training data, which, in some sense, limits their applicability [82]. Deep learning algorithms have been applied in the classification of pneumonia diseases with high accuracy for the identification and diagnosis of different pneumonia disorders from photographs [83]. Deep learning methods make use of deep neural networks, which consist of a large number of layers of interconnected nodes, to extract intrinsic representations of image input. Deep learning approaches in the classification of pneumonia diseases have been one of the promising areas able to largely influence the domain of research related to the lungs and give improvement in patient care.

However, this computational load can be mitigated through Transfer Learning (TL). TL leverages DNNs pre-trained on extensive image datasets to adapt to new, smaller datasets for classification tasks, thereby reusing previously acquired knowledge to solve new problems. Although TL demands less data and training time compared to developing new DL architectures, it remains more computationally demanding than ML and often lacks the medical interpretability associated with traditional ML methods. This emphasizes the potential benefits of ML in healthcare, which is a central focus of this research. Recent technological advancements have led to the digitization and storage of extensive medical records [84]. This has spurred efforts to explore AI's potential in leveraging big data to support healthcare professionals in making informed diagnostic decisions, thereby enhancing patient care quality [85],[86]. AI shows great promise in healthcare by aiding in the prevention, diagnosis, and treatment of diseases with high precision while also reducing human error [71]. AI tools have proven effective in identifying various conditions, including skin cancer [87], lung cancer [88], breast cancer [89], heart diseases [90], [91], eye diseases [92], tuberculosis [93], COVID-19 [94], and pneumonia [95],[96]. Specifically, the use of

AI in pneumonia classification has been explored across various medical modalities, such as chest X-rays, CT, ultrasounds, and clinical data [95].

2.4 CONVOLUTIONAL NEURAL NETWORKS (CNNs)

CNNs have become more well-known for their enhanced capabilities in image prediction. The Layers of convolution in these networks, combined with filters, are instrumental in obtaining temporal and spatial features from images. These layers utilize a weight-sharing method, which significantly lowers computational demands [97]. From an architectural perspective, CNNs are basically feedforward artificial neural networks (ANNs) with two key characteristics: neurons that are part of the same filter are connected only to localized regions of the image to maintain spatial relationships, and weights are shared to minimize the overall parameter count of the model. A CNN comprises three main components: i) the Convolution layer, which is responsible for feature learning; ii) the Max-Pooling (or subsampling) layer, which reduces the image size and dimensionality, thereby decreasing computational effort; and iii) the Fully Connected layer, which provides the network with its classification capabilities [98]. Figure 2.5 illustrates the architectural layout of a CNN.

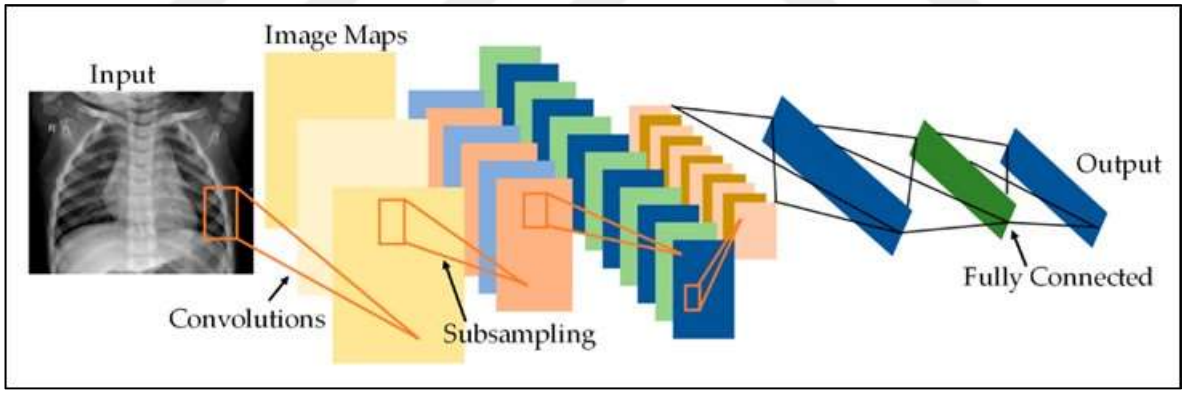


Figure 2.5: CNN Architecture.

The convolutional layer underpins CNN. Kernels or filters are used to convolutionally process incoming data to find local patterns and features. This initial step detects edges and textures for the network. It finds more complicated traits as the network deepens. Data spatial dimensions are reduced by pooling layers after convolutional layers. Optimising computational efficiency and reducing overfitting. Maximise and average pooling select the maximum and average values from each zone, respectively. Fully connected layers create predictions at the network's end using high-level features retrieved by preceding layers.

These layers enable complicated decision-making by connecting all neurones in adjacent layers.

Kernels weigh filters, which limit input data windows during convolution. Learning these kernel values lets the network automatically extract input features. To learn complex patterns, ReLU (Rectified Linear Unit) activation functions give the network non-linearity. To improve training, ReLU replaces negative values with zero. The filter's input convolution step size. Stride affects output spatial dimensions, receptive field, and retention. CNNs have seen various architectural developments to tackle different challenges in computer vision. Examples include AlexNet, LeNet-5 [99], VGGNet [100], Inception [101], Xception [102], ResNet [103], MobileNet [104], and DenseNet [105]. This thesis has focused on MobileNet architecture.

2.4.1 The Convolution Layer

The convolutional layer is the primary constituent of Convolutional Neural Networks. The layer being described becomes entitled to the core task of feature extraction. This layer largely identifies and detects different patterns and features present in the input image. The convolutional layer utilizes a set of filters scanning the input image and computing dot products between the entries of the filter and image pixels. These dot products will be further utilized in generating feature maps as outputs of the convolution layer [106]. Every filter in a convolutional layer recognizes one specific characteristic or pattern in the image. For instance, one filter could be engineered to look for edges and another for circles. Many features can be extracted from the input image by applying multiple filters in a convolutional layer. These features can then be used for image classification. Furthermore, the convolutional layer should maintain the spatial structure of an input image. This is attained through a local connectivity pattern, where every element in the feature map only shares a connection with a few elements in the input image. This enables the network to learn the spatial relationships between different pixels in the image, thus helping to retain the overall structure of the image [107].

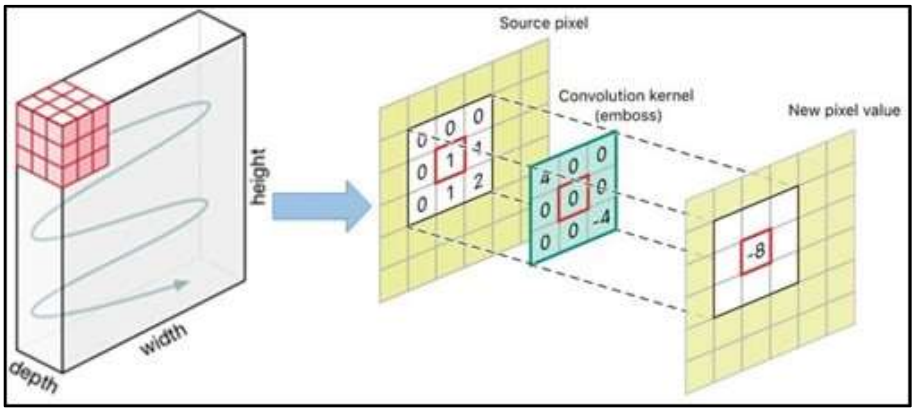


Figure 2.6: Convolution Operation.

Perhaps the most important component of any CNN is the convolution layer, as it is responsible for feature extraction from the input image, with the additional constraint of preserving the spatial structure of the image [108]. Subsequently, this information is transmitted to other levels in the network for additional analysis and eventual forecasting, as depicted in Figure 2.6.

2.4.2 The Pooling Layer

Pooling in a CNN is a down-sampling process that reduces the spatial dimensions of the feature maps. The main idea behind this process is to decrease computational load for the subsequent layers while retaining the most significant information contained within the feature maps. There are mainly two kinds of pooling used in CNNs: Max Pooling and Average Pooling, as illustrated in Figure 2.7. Max pooling is a process whereby, in every region of the feature map, the highest value is picked and taken as the new feature value. Average pooling is where the mean is taken for all the values within a given region and is taken to replace that region as the new feature value [109].

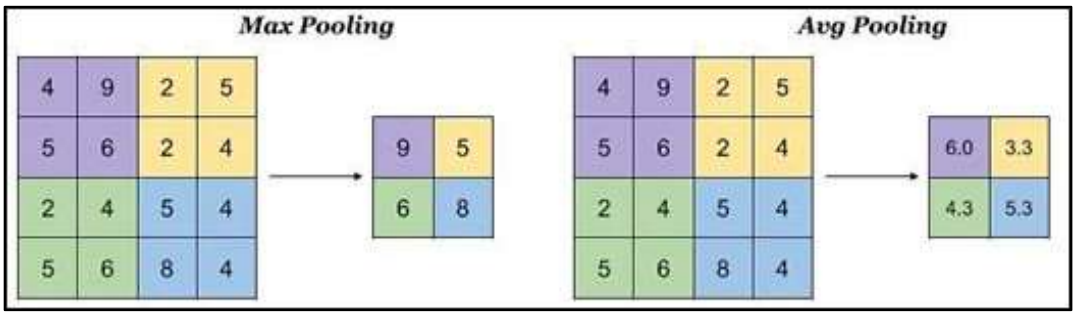


Figure 2.7: The Pooling Operations.

The pooling layer would be fed with a feature map; the extend of pooling operation controlled by a square window, the kernel size. Another adjustable parameter would be the stride, which controls the size of the step that the kernel takes at a time. The pooling layer outputs a new feature map with reduced spatial dimensions [110]. Pooling layers have shown to potentially improve the performance and stability of a CNN, turning them into one of the most prevalent elements in most of the recent cutting-edge topologies.

2.4.3 The Fully Connected Layer

CNNs are equipped with a layer known as the fully connected layer, often mentioned as the dense layer. In this layer, an input weight matrix is multiplied by a matrix of units called the weights. This is optionally added with a bias term and an activation function. The name "fully-connected layer" comes from the presence of a connection between every neurone in this layer and all neurones in the layer above [111]. In a CNN, the fully-connected layer is essentially an output layer, usually used to predict a class based on the features learned from the convolutional and pooling layers. This fully-connected layer gives an output vector in which every element of the vector is a projected score for a certain class. In the classification problem, the predicted class is decided by choosing the class that has a maximum score [112]. It can also be applied inside the CNN architecture, mostly in the configuration of an MLP. Here, fully-connected layers represent higher-level abstract information from the input, providing a compact representation that can then be fed into the next layer. During training, this fully-connected layer's weights and biases are learned by a supervised learning technique called backpropagation. Now, the goal will be to find what weight and bias settings minimize the prediction error on the training data [113].

2.5 MOBILENET ARCHITECTURE

MobileNet is an advanced architecture that merges a deep, lightweight convolutional neural network with an innovative technique known as depthwise separable convolutions. This design enhances its effectiveness for various computer vision tasks. The core components of MobileNet’s architecture are depthwise separable filters [114]. On a deeper dive into the architecture of MobileNet Figure 2.8, it is known for using depthwise separable convolutions which are very core to what makes this model more efficient and faster. This method divides the standard convolution operation into two except layers, depthwise convolutions and pointwise squares. Specifically, depthwise convolution employs one single convolutional

filter per input channel, which improves spatial feature learning in a model. Then, the pointwise convolution does 1x1 convolutions to mix those depth-wise features, giving a good feature map with relations together. They both drastically decrease the number of parameters and computational costs with respect to standard convolutions, which is especially important since MobileNet will be used in mobile devices or even embedded hardware.

MobileNet network is structured in layers containing depthwise separable convolutions, batch normalization and ReLU activation functions. Batch normalization aids the stability and speed of training by normalizing each layer's input, while ReLU acts as an efficient yet effective non-linearity. On the other hand, MobileNet contains a global average pooling layer before its final fully connected layer, which takes spatial dimensions of feature maps to a fixed size, and it helps in further reducing computing burden as well as parameterizing. The architecture is parameterized with width and resolution multipliers, allowing for fine-grained control over the trade-offs between model size and performance. MobileNet provides a set of pre-trained models optimized for fast processing and small size, which can be further improved by tuning the multiplier value on compression factors. MobileNet is an efficient model for real-time computer vision tasks, which combines depthwise separable convolutions with batch normalization and strategic pooling to gain performance without any meaningful increase in computational cost.

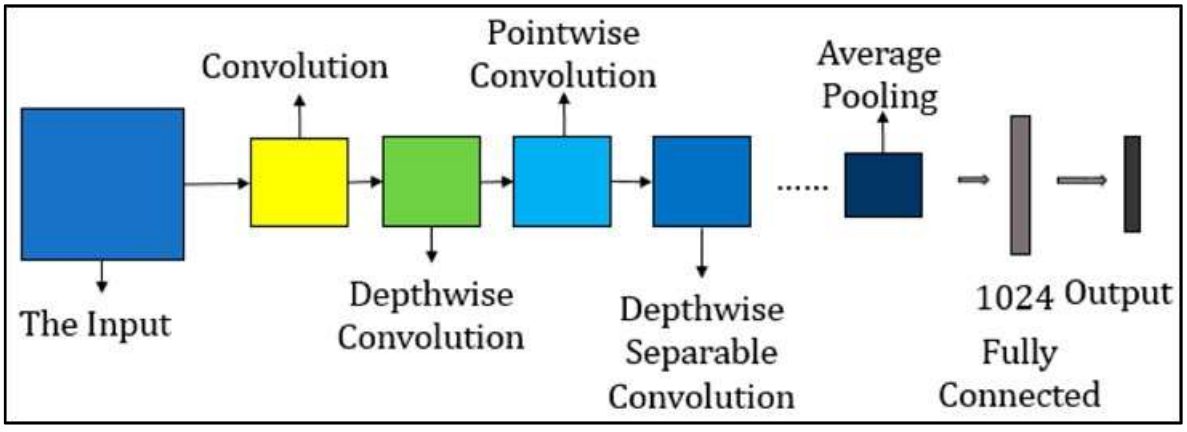


Figure 2.8: MobileNet Architecture [115].

2.6 OPTIMIZATION ALGORITHMS

Optimization problems are everywhere in different fields like industry, transportation, and finance; also, those that have constraints, high dimensionality, or non-linear behavior become stiff. The computational burden of these problems grows exponentially in the number of variables [116],[117]. Large-scale problems may regard issues with the typical optimization methods and thus convergence to solutions of lesser quality. Hill-climbing and simplex algorithms are classical methods that can adequately solve simpler tasks, but they require a strict computational logic which makes them only applicable to linear constraint problems with linearity, differentiability or continuity of the objective function [118],[119]. Therefore, there are a lot of real-world engineering problems in which these traditional approaches may not be good enough. Metaheuristic algorithms are powerful instruments for tackling difficult optimisation issues in response to these challenges [120]. Metaheuristics balance exploration and exploitation strategies to solve high-dimensional, non-linear problems, unlike standard methods. Exploration searches generally over the solution space, while exploitation refines the best solutions. This dual technique speeds up difficult optimisation [121].

The increasing adoption of metaheuristic optimization algorithms is driven by several key advantages. Their concepts are intuitive and easy to implement, they do not require gradient information, which is often a restrictive factor, and they are adept at navigating around local minima, making them well-suited for optimization tasks. Additionally, their adaptability to a wide range of problems across different domains has been accelerated by advancements in high-performance computing [122], [123]. Metaheuristic algorithms are diverse and can be categorized based on various criteria, including the nature of their search strategies and their sources of inspiration [124],[125]. These categories encompass single-point or population-based search methods, nature-inspired or abstract approaches, static or dynamic environments, single-neighborhood or multi-neighborhood exploration, and memory-based or memoryless algorithms. A notable trend in the literature is the classification based on inspiration sources [126],[127], which streamlines the vast array of algorithms into five primary categories. The first category includes algorithms inspired by biological evolutionary processes. Genetic Algorithms (GA) [128] are among the earliest and most prominent, using mutation, recombination, and natural selection to evolve solutions. Other algorithms in this category include Differential Evolution (DE) [129], Biogeography-Based

Optimization (BBO) [130], and several others, each drawing from different aspects of natural evolution to address optimization problems, .

Physics-based algorithms, which optimise using physical processes, are also important. Simulated Annealing (SA) [131] simulates metallurgy's annealing process to avoid local optima by probabilistically accepting poorer solutions. The Gravitational Search Algorithm (GSA) [132] mimics mass interactions, whereas the Black Hole Algorithm (BHA) [133] and Equilibrium Optimiser (EO) [134] employ different physical similarities. This category includes Fick's Law Algorithm (FLA) [135], Optical Microscope Algorithm (OMA) [136], and Kepler Optimisation Algorithm (KOA) [137], which use distinct physical principles to improve search tactics. Various human-inspired algorithms mimic human social behaviours, learning, and emotions. To find optimal solutions, Teaching-Learning-Based Optimisation (TLBO) [138] uses two phases of teaching and learning. League Championship Algorithm (LCA) [139] and Brain Storm Optimisation (BSO) [140] use human-like interactions and methods to solve optimisation problems.

Emerging as a more mathematical metaheuristic, the mathematics-based algorithms did not make use of any particular phenomenon or behavior. Each of these methods have their own way to perform searches, like, for instance, the Sine Cosne Algorithm (SCA) [141] utilizes trigonometric functions during its spatial search while other use mathematical formulas in order to make your best decision when it comes to optimizations such could be Gradient-Based Optimizer (GBO) [142], or Exponential Distribution Optimzier (EDO)[143]. As shown in Figure 2.9, metaheuristic algorithms can be classified into different categories based on their sources of inspiration. In this part of the review two metaheuristic algorithms are Dung Beetle Optimization (DBO), and Fick's Law Optimizer is discussed. To facilitate search, DBO is inspired by the natural behaviors of dung beetles-rolling and breeding activities-which provide each beetle with multiple (behaviors) to follow. Fick's Law Optimizer uses Fick's laws of diffusion to create a model that describes particles moving about in solution. This study compares these two algorithms and how they solve complex optimization problems, describing their methods and contributions to that field.

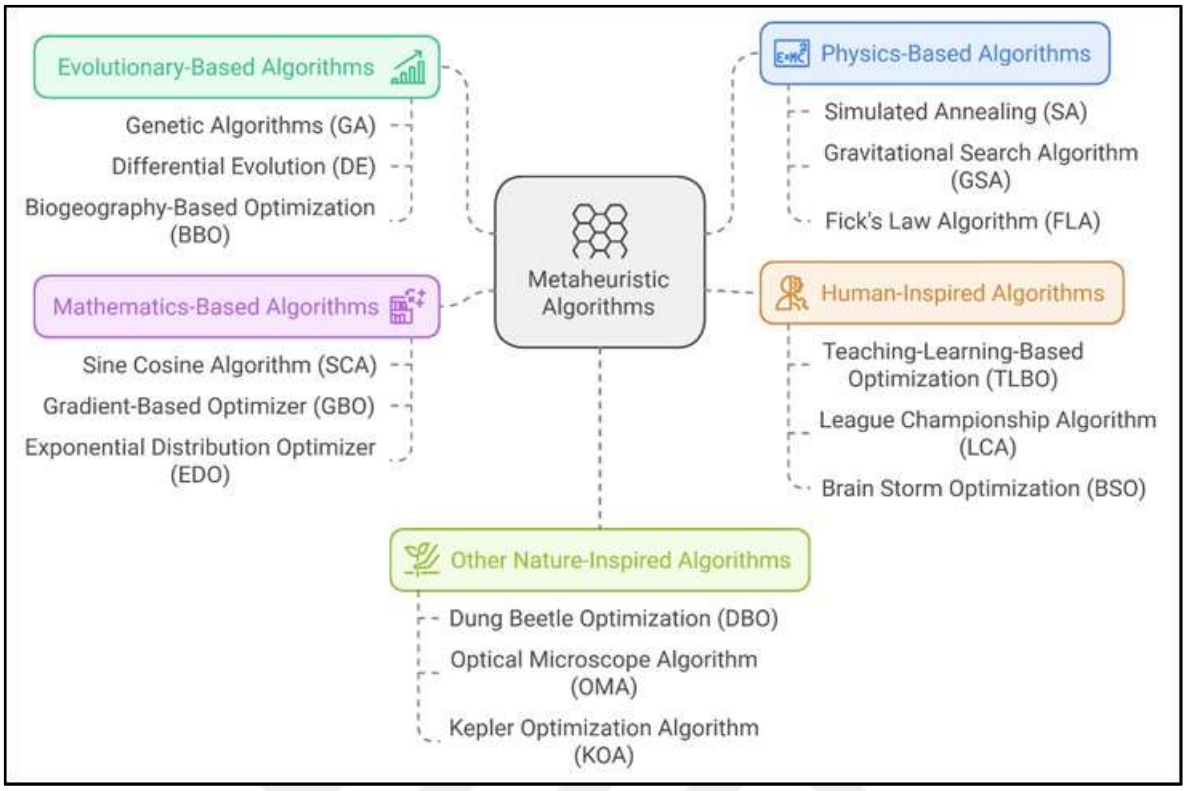


Figure 2.9: Classification of Metaheuristic Algorithms: Key Categories and Examples.

2.6.1 Dung Beetle Optimization (DBO)

DBO algorithm [144] is a complex and adaptive optimization approach inspired from the natural behaviors of dung beetles, especially their rolling (building ball) and breeding activities [145]. Specifically, the rolling dung beetles consist of equations to simulate position updates in DBO:

$$X_i\,(t+1) = X_i\,(t) + \alpha \times k \times X_i\,(t-1) + b \times \Delta x \qquad (2.1)$$

Where (X_i (t)) is the position of ith dung beetle at *t-th* iteration and k ∈(0, 0.2] a deflection coefficient that controls the degree to which beetle initiates search process where the parameter (b) is with (0, 1), and where Δx denotes boost in light intensity that calculated as ($\Delta x = |X_i\,(t) - X^{\omega}|$, here (X^{ω}) signifies the global worst position. This model makes it so that beetles can wander in their environment by changing positions based on past information about the world they inhabit and current conditions. When a dung beetle encounters an obstacle during its rolling behavior, it performs a reorientation dance to find a new direction. The update for the beetle's position during this dance is given by:

$$X_i\,(t\,+\,1)\,=\,X_i\,(t)\,+\,tan(\theta)|X_i\,(t)\,-\,|X_i\,(t\,-\,1)| \tag{2.2}$$

Where (θ) is an angle within [0, π]. If (θ) equals 0 or π, the position does not update, effectively halting the dance and maintaining the current position. Figure 2.10 illustrates the conceptual model of the boundary selection strategy used in the DBO algorithm. The model demonstrates how the algorithm simulates the behavior of dung beetles in their natural environment to optimize solutions. The different colored points represent various elements within the optimization process, including upper and lower boundaries (Ub and Lb), feasible solutions, precise solutions, and the current best solution (X*). The visualization aids in understanding how the DBO algorithm dynamically adjusts the boundaries and positions of the breeding balls to effectively explore and exploit the search space.

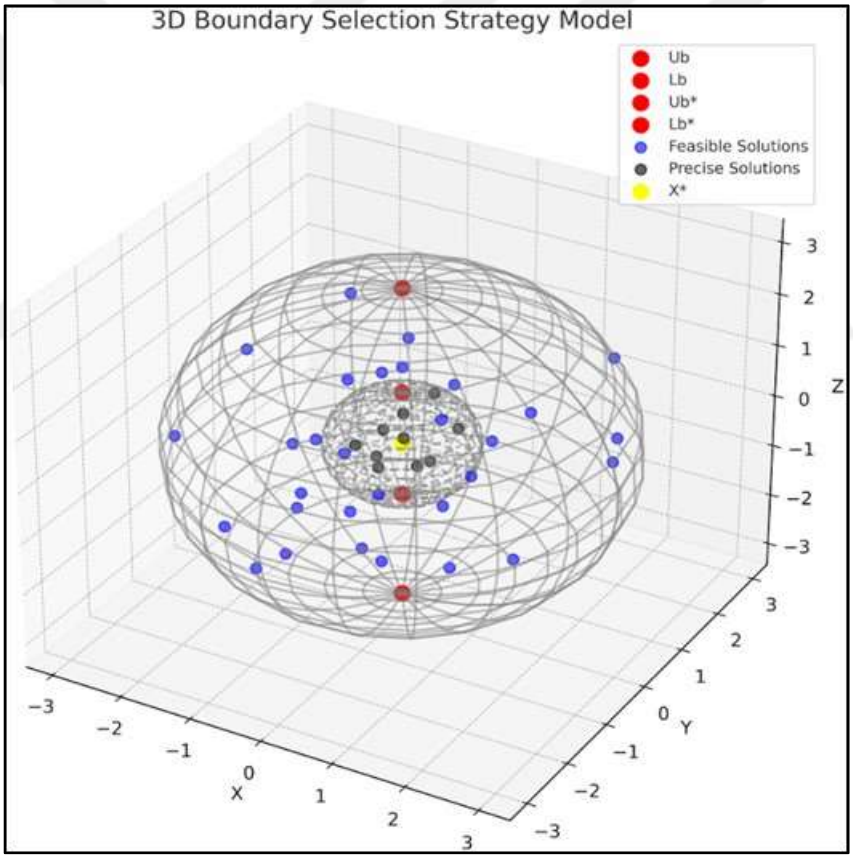


Figure 2.10: The Conceptual Model of Boundary Selection Strategy.

A crucial aspect of the DBO algorithm is modelling the behavior of female dung beetles when selecting spawning sites. The algorithm defines a boundary selection strategy to ensure effective breeding. This is represented by:

$$Lb^* = max(X^* \times (1 - R), Lb) \quad (2.3)$$

$$Ub^* = min(X^* \times (1 + R), Ub) \quad (2.4)$$

where (X^*) is the current local optimal position, (Lb^*) and (Ub^*) denote the lower and upper boundaries of the spawning area, respectively, and $(R = 1 - t/T_{max})$, $with\ T_{max}$ being the maximum number of iterations. Female beetles lay eggs within this defined area, adjusting the position of their breeding balls dynamically as described by:

$$B_i(t + 1) = X^* + b_1 \times (B_i(t) - Lb^*) + b_2 \times (B_i(t) - Ub^*) \quad (2.5)$$

Here, (B_i (t)) is the position of the *i-th* breeding ball, and (b_1) and (b_2) are random vectors of size 1×D. This ensures that the breeding balls stay within the designated spawning area. The foraging behavior of dung beetles is modelled by defining an optimal foraging area. The boundaries of this area are specified as:

$$Lb^b = max(X_b \times (1 - R), Lb) \quad (2.6)$$

$$Ub^b = min(X_b \times (1 + R), Ub) \quad (2.7)$$

where (X_b) represents the global optimal position. The update rule for the dung beetle's position in the foraging area is:

$$X_i(t + 1) = X_i(t) + C_1 \times (X_i(t) - Lb^b) + C_2 \times (X_i(t) - Ub^b) \quad (2.8)$$

with (C_1) and (C_2) being random vectors that control the beetle's exploratory behavior. Additionally, a category of dung beetles known as 'thieves' are incorporated into the DBO algorithm. These beetles seek to steal dung balls from others, which is modelled as:

$$X_i(t + 1) = X^b + S \times g \times (|X_i(t) - X^*| + |X_i(t) - X^b|) \quad (2.9)$$

Where (S) is a constant, and g is of size 1×D. The intuition behind this behavior has to do with the fact that it makes the optimization competitive across different coordinates. The DBO algorithm starts with a population of dung beetles and categorizes them into four types: Rolling Beetles, Breeding Balls, Minor includes Small beetles, and Stealing, which are named normal SDGS residents, respectively. An example would be in a population of 30 beetles: 20% roll, 20% breed balls, and approximately equal small and stealing beetles (

~25%); the race is already evenly distributed. This ensures there are different types of operations each category works through based on the rules and equations within the algorithm to optimize everything overall. The distribution of all candidate solutions from the DBO algorithm after several iterations finds positions for these agents and further notifies that using this method to analyze each step and subprocess in a single point where we identify maximal fitness.

2.6.2 Fick's Law Optimizer

A Fick's Law Optimizer optimization technique inspired by the physical laws of diffusion [135] can be used to solve complex multi-disciplinary design spaces. The algorithm uses these principles to direct search agents to better explore and exploit the solution space. Fick's Law Optimizer is based on modelling the diffusion process, where search agents act as diffusing particles that want to find the optimum with a gradual movement towards areas of lower cost, better quality solutions. Fick's Law Optimizer The basic idea of FLO is rooted in Adolf Fick's first law of diffusion, which states the flux due to concentration gradient. Fick's Law Algorithm (FLA) relies on a population matrix X to simulate molecular diffusion within an optimization context. The matrix, initialized based on molecule positions in a bounded search space, is defined as:

$$\begin{matrix} x_{1,1} & x_{1,j} & \dots & \dots & x_{1,D} \\ x_{2,1} & x_{2,j} & \dots & \dots & x_{2,D} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \lfloor x_{N,1} & x_{N,j} & \dots & x_{N,D-1} & x_{N,D} \rfloor \end{matrix} \tag{2.10}$$

Here, *N* is the size of the population, and *D* is the dimensionality of each individual within the population. Each element $X_{i,j}$ of this matrix specifies the *j*-th dimension of the *i*-th molecule. The following equation first initializes this matrix:

$$x_{i,j} = lb + rand(1, D) \times (ub - lb) \tag{2.11}$$

lb and ub are the lower and upper bounds of the search space, respectively. For individual fitness evaluations, the population is divided into two subfamilies, N1 and N2. Under the control of iteration parameter TF^t, molecular migration occurs from a high to low concentration area, and it is given by:

$$TF^t = \sin h\,(^t/_T)^{c1} \tag{2.12}$$

It can be expressed as an algorithm where (t) is the current iteration number, and (T) is the maximum number of iterations. The three discrete parts in the molecular position update method are the diffusion operator, namely (*DO*), equilibrium operator (*EO*), and steady-state operator (*SSO*), which were used to find the best value for the system. For that, the new equation is:

$$X_i^t = \begin{cases} DO & if\ TF^t < 0.9 \\ EO & if\ 0.9 \leq TF^t \leq 1 \\ SSO & if\ TF^t > 1 \end{cases} \tag{2.13}$$

In Equation (3), (TF^t) denotes the transition factor at the current iteration (t), computed using the hyperbolic sine function ($\sin h\left(^t/_T\right)^{c1}$). In this context, (t) represents the current iteration number, (T) denotes maximum number of iterations, and c1 is a control parameter influencing the transition function's behaviour. The transition factor (TF^t) directs the algorithm's capacity to equilibrate exploration and exploitation within the search space as time progresses.

In Equation (2.13), (X_i^t) indicates the position of the *i*-th molecule at iteration t, which is updated using different operators depending on (TF^t) value. When (TF^t) < 0.9, the DO is used to favor exploration in the search space. Then, when 0.9 ≤ (TF^t) ≤ 1, the EO is used for balancing exploration and exploitation. Finally, for the case (TF^t) > 1, an SSO is adopted with a steady-state equilibrium to favor exploitation and concentrate on refining the final optimization stages of the solution. During the pure diffusion phase, for (TF^t)< 0.9, molecules move between the compartments toward the concentration gradient. The parameter controlling this movement, T_{DO}^t, is given by:

$$T_{DO}^t = C_5 \times TF_t - r \tag{2.14}$$

The constant C_5 is typically fixed at 2, and (r) is a random variable drawn from a uniform distribution with expected values between 0 and 1. The course of molecular mobility is defined by:

$$X_{p,i}^t \begin{cases} from\ i\ to\ j\ ;\ T_{DO}^t < rand \\ from\ j\ to\ i\ ;\ others \end{cases} \tag{2.15}$$

The following formula is used to determine the number of molecules that shift from region *i* to region *j*:

$$NT_{ij} \approx X_i \times r1 \times (C_4 - C_3) + N_i \times C_3 \quad (2.16)$$

with additional constants C_3 and C_4 and a random integer r1 from 0 to 1. By including the diffusion process into this optimization framework, we may dynamically adjust the search space to conform to the natural diffusion rules mentioned in Fick's second law. Based on Equation (2.14), (T_{DO}^t) denotes the transfer parameter during the diffusion operator phase, which governs the migration of molecules between regions. The parameter (T_{DO}^t) is computed utilizing a constant C_5, generally established at 2, alongside the current transition factor (TF_t). The variable (r) is a random number produced from distributed uniformly ranging from 0 to 1, so randomness is included to emulate natural diffusion behaviour.

Based on Equation (2.15), the direction of molecular movement among regions *i* and *j* is determined by comparing the value of a parameter with a random number (rand) drawn from distributed uniformly. If a value is less than (rand), molecules transition from region *i* to region *j*; otherwise, they transition from region *j* to region *i*. Based on Equation (2.16), the number of molecules (NT_{ij}) moving from compartment *i* to compartment *j* is calculated as the number of molecules in compartment *i*, denoted as (X_i), times a random number (r1) among 0 and 1. The parameters (C_3) and (C_4) are diffusion coefficients that affect motion; (C_3) applies to local shifts and (C_4) to the gradient. The variable (N_i) refers to the initial number of molecules in region *i*, and the product $(N_i \times C_3)$ ensures that the diffusion process complies with the distribution of molecules in the region in question.

In the transition phase of the system from exploration to exploitation, the binding modes become excellently essential in deciding and adjusting molecular positions such that they reach stability at an equilibrium point. It will arrive at a point of equilibrium, where all parts of the region around a level surface are evenly flat and allow thorough searches in the neighbourhoods of current optimal solutions. This methodical transition is important to correctly show alternative solutions and not overlook local details in the search space by scoring out the observations based on probability. Being able to switch between wide exploratory moves and narrow exploitative fine-tunes, FLA could navigate through complex optimization landscapes to find a global optimum without getting trapped prematurely into

a local optimum. This balance is a critical methodology for the success of this algorithm in all the diverse optimization problems. FLA defines the second stage as marking the proper shift from the initial exploration phase, where individuals search for new knowledge, to focused exploitation, absorbing the last bits of information. It occurs through the equilibrium operator (EO), that makes molecules move towards an equilibrium condition where their concentration is the same everywhere in the system. This is a critical point, focusing the search on the most promising prospects discovered during exploration.

While the (EO) phase, the molecules adjust their positions to balance out concentration gradients that have become practically negligible and thus manage to successfully imitate the natural process of diffusion to reach an equilibrium. The update of position during this phase follows the equation:

$$X_{p,g}^{t+1} = X_{EO,p}^{t} + Q_{EO,g}^{t} \times X_{p,g}^{t} + Q_{EO,g}^{t} \times (MS_{p,Eo}^{t} \times X_{EO,g}^{t} - X_{p,g}^{t}) \qquad (2.17)$$

Here, $X_{p,g}^{t}$ and $X_{EO,g}^{t}$ are respective positions within groups p and g. The parameter $MS_{p,Eo}^{t}$ indicates the relative amount that belongs to group g and is thus a good measure of the spatial distribution of the group at iteration t. $Q_{EO,g}^{t}$ represents a diffusion rate factor for the area that corresponds to group g, where $Q_{EO,g}^{t}$ is computed by a function that updates the dynamics of diffusion between the areas. The above formulation will allow for the systematic investigation of the changes in position and diffusion rates in the groups, enhancing the capability of the model to account for spatial variations across different iterations. Diffusion rate factor of g, computed by:

$$Q_{EO,g}^{t} = R_{1}^{t} \times DF_{g}^{t} \times DRF_{EO,g}^{t} \qquad (2.18)$$

In this framework, DF_{g}^{t} functions as the directional factor, taking a value of either +1 or -1 to indicate movement direction. R_{1}^{t} is a random variable uniformly distributed between 0 and 1, introducing stochastic variation into the process. The diffusion rate, $DRF_{EO,g}^{t}$, is thereby defined about these parameters, shaping the diffusion behavior and facilitating adaptive movement across different iterations within the model. This configuration provides a nuanced approach to managing directional shifts and random fluctuations, essential for effectively simulating dynamic diffusion processes. Where $DRF_{EO,g}^{t}$, defined by:

$$DRF_{EO,g}^{t} = exp(-\frac{J_{p,EO}^{t}}{TF^{t}}) \quad (2.19)$$

where $dc_{g,EO}^{t}$ and $dx_{p,EO}^{t}$ represent the concentration and positional disparities among regions g and p, respectively. Reducing concentration gradients guides the movement of molecules, successfully emulating the diffusion process as outlined by Fick's law. During the algorithm's stable part, the molecules' positions are fine-tuned. The movement distance is inversely proportional to its fitness to get each molecule to the best position. Molecules with better fitness have shorter movement distances. The formula for where molecules are in this stable phase:

$$X_e = D_{\text{alphs}} \times e^{\Pi} \times cos(II \times 2\pi) \quad (2.20)$$

In this formulation, X_e denotes the updated position, representing the outcome of the equation and indicating the new position after applying transformations based on other variables. The parameter D_{alphs} functions as a scaling coefficient for the exponential and cosine components, potentially embodying a distance factor, scaling parameter, or other characteristic relevant to the system being modeled. The constant *e*, the base of the natural logarithm (approximately 2.71828), is utilized within the exponential function to represent growth or decay phenomena. Additionally, *II* is a random variable that adjusts the balance between global and local search strategies, supporting the adaptability of the model's search dynamics; it is calculated by:

$$II = (Fc - 1) \times rand + 1 \quad (2.21)$$

In this case, (rand) stands for a random variable that is usually restricted to the range from 0 to 1, thus bringing stochastic changes to the model. The factor (Fc) is a tuning knob that controls the trade-off between exploration (global search) and exploitation (local search) in the algorithm, thus shaping the search behavior. A bigger (Fc) value steers the algorithm toward wider than ever before exploration. Thus, the global search is expanded further, while a smaller (Fc) propels a more concentrated local search around attractive regions. Besides, the expression cos (II × 2π) adds a “wave” movement to the function because the cosine curve gives a periodic rotation as the argument is set at (II × 2π). This wavelike pattern is produced because of the cosine that allows periodic repetition.

This equation, X_e, employs a synthesis of exponential growth/decay and oscillatory (cosine) behaviour, modulated by a stochastic factor (*II*) governed by (Fc), to ascertain a new location in an optimization or search process. The objective is to equilibrate the exploration of novel regions with the exploitation of established advantageous locations within the search space. Calculation of total time complexity, the cumulative temporal complexity of FLA throughout all phases, accounting for the complexity contributions from all phases can be summarized by:

$$O(FLA) = O(N \times D) + O(T \times (O(NT_{12} \times D) + O\left(\left({}^{N}/_{2} - NT_{12}\right) \times D\right) + O\left(\left({}^{N}/_{2} - N_{transfer}\right) \times D\right) + 6 \times O\left({}^{N}/_{2} \times D\right)) \quad (2.22)$$

The largest terms are the primary focus in Big *O* notation, especially as *T* increases.

$$O(FLA) = O(N \times D) + O\left(T \times 8 \times {}^{N}/_{2} \times D\right)$$
$$= O(N \times D + 4 \times T \times N \times D) \quad (2.23)$$
$$= O(N \times D \times (1 + 4 \times T))$$

This calculation demonstrates that FLA's temporal complexity is linear concerning the population's size and its dimensionality, scaling linearly with the number of iterations increased by a factor of three. That illustrates that although the technique is economical for each iteration, its total computational cost escalates considerably with an increase in iterations and population size, a characteristic typical of population-based optimization algorithms.

2.7 RELATED WORKS

Pneumonia categorisation has been a captivating subject of study for numerous scholars. Multiple research investigations have been undertaken to assess the capability of AI classification systems in detecting and analysing pneumonia scans. The utilisation of AI systems in the medical field is unquestionably advantageous for the identification and diagnosis of pneumonia. Researchers in [146] proposed a method for classifying X-ray images into COVID-19 and normal categories using a multi-model classification process. This approach integrates a Support Vector Machine (SVM) with the VGG16 Convolutional network by adding an extra layer of convolution, pooling, and dense connections. The Radial

Basis function was used for transformations to achieve optimal results. The proposed CovXmlc model, compared with five existing models, demonstrated up to 95% accuracy with a minimal dataset, significantly surpassing existing models in metrics such as recall, precision, and f-score. On the other hand, study [147] proposed using a neutrosophic method to distinguish between viral and bacterial pneumonia, which are often confused with COVID-19. This method categorizes images into True (T), False (F), and Indeterminate (I) sets, enhancing the clarity and detail of lung opacity features through alpha-mean and beta-enhancement preprocessing. The study employed deep learning models like ResNet-50, VGG-16, and XGBoost to analyze these images, tested on the ActualMed COVID-19 Chest X-ray and COVID-19 Radiography datasets. Enhanced neutrosophic images achieved the highest accuracy of 97.33% in comparative analysis, demonstrating the effectiveness of this approach. In [148], researchers proposed fast and reliable automatic technologies to help healthcare teams during the COVID-19 pandemic. With standard blood testing, the study screens for COVID-19 using ML. Multiple ML classifiers were tested on a dataset from Hospital Albert Einstein in São Paulo, Brazil, including 608 patients, 84 of whom had COVID-19. A local Decision Tree Explainer (DTX) and a Criteria Graph provide local and global explanations, respectively. The top classifier was the Random Forest (RF) with 0.88 accuracy, 0.76 F1-score, 0.66 sensitivity, 0.91 specificity, and 0.86 AUROC. Patterns from DTX and the Criteria Graph helped clinicians analyse blood parameters. The findings show electronic health record systems might use the methodology.

In [149], developed a machine learning model to detect tuberculosis (TB) using 1196 chest X-ray images sourced from Kaggle.com, trained using Google’s Collaboration Platform. The model efficiently identified TB patterns, achieving an average precision of 0.934 and precision and recall rates of 94.1%. It also recorded sensitivity and specificity of 96.85% and 91.49%, respectively, with an F1 score of 0.941 and overall accuracy of 94%. These results highlight the model's effectiveness in TB detection, emphasizing the need for further research and validation for clinical use. Researchers in [150] proposed a method to improve COVID-19 classification using chest X-rays by developing a model called CovidGAN, based on an Auxiliary Classifier Generative Adversarial Network (ACGAN). Since obtaining sufficient training data for deep learning networks like CNNs is challenging due to the pandemic's recent onset, CovidGAN generates synthetic chest X-ray images to supplement the limited data. Their results show that incorporating these synthetic images

increased CNN accuracy from 85% to 95%, suggesting this approach could enhance radiology systems and speed up COVID-19 detection. In [151], developed a hybrid deep learning CNN model for identifying COVID-19 using chest X-rays. Combining VGG16 and VGG19 architectures with max and average pooling layers, the model also features three dense layers, various activation functions, and a dropout layer to combat overfitting. It was compared with architectures like EfficientNetB0 and ResNet50 and classification techniques including KNN, Naive Bayes, RF, SVM, and Neural Networks. The hybrid model, especially with average pooling and SVM-linear, achieved an accuracy of 92%, aiming to help radiologists and physicians reduce misdiagnosis and confirm COVID-19 cases effectively. Researchers in [152], proposed a method to improve the diagnosis of Primary Sjögren's syndrome associated with interstitial lung disease (pSS-ILD) using serum Raman spectroscopy and machine learning algorithms. Serum samples from 30 pSS patients, 28 pSS-ILD patients, and 30 healthy controls were pre-processed and analyzed using principal component analysis (PCA) for dimensionality reduction. The classification was performed using SVM, KNN, and RF models. The SVM (poly) model achieved an average accuracy of 89.52%, sensitivity of 91.27%, precision of 89.52%, and an AUC value of 0.921, demonstrating its superior discriminative effect.

A computer-aided diagnosis (CAD) system that uses an ANN optimized by an artificial bee colony ABC algorithm has been proposed in [153] for detecting COVID-19 from lung CT images. The system based on region fractal analysis using the ABC optimization algorithm provided an excellent performance level (92.37%) when dealing with images from 470 lung CT slides, being applicable in automating the diagnostic process of COVID-19 virus. Researchers in [154] proposed a method using ML and radiomics features to quickly detect COVID-19 from chest X-rays (CXR), even with other pneumonia types and varying severity levels. The study used 378 CXR images, selecting 71 radiomics features per lung segment via Z-score heatmaps and ANOVA tests. Three machine learning algorithms—fine Gaussian SVM, fine KNN, and ensemble bagged model (EBM) trees—were evaluated on a test set of 115 COVID-19 and 100 other pneumonia/normal CXR images. Severity scores from 0 to 4 were assigned by radiologists. The EBM with selected features was most effective, achieving 87.8% sensitivity, 97% specificity, 95.2% accuracy, and a 0.9228 AUC, and could detect COVID-19 where radiologists saw no abnormalities. The method takes under two minutes for feature extraction and diagnosis and can be integrated with standard X-ray reporting

systems for efficient, cost-effective early diagnosis. Researchers in [155] proposed a series of deep-learning-based methods to detect COVID-19 in chest X-ray images. The study utilized deep feature extraction and the fine-tuning of pre-trained CNNs, such as ResNet18, ResNet50, ResNet101, VGG16, and VGG19. For classification, a SVM classifier with different kernel functions (Linear, Quadratic, Cubic, and Gaussian) was employed. Additionally, the paper introduced a novel CNN model developed for end-to-end training. The dataset consisted of 180 COVID-19 and 200 normal chest X-ray images. The most notable outcome was achieved using the ResNet50 model with SVM employing a Linear kernel, reaching a classification accuracy of 94.7%. This result highlighted the effectiveness of deep learning, particularly when compared to methods relying on local texture descriptors, in diagnosing COVID-19 from X-ray images.

Table 2.1 provides a comprehensive overview of various studies that have proposed methods for diagnosing diseases using machine learning and radiomics features. Each study is summarized with the key methods employed, the dataset used, and the achieved accuracy. The methods range from multi-model classification and segmentation algorithms to advanced neural networks and optimization techniques, applied to datasets such as chest X-rays, CT scans, and ECG data. The accuracies reported highlight the effectiveness of these approaches in improving diagnostic precision across different medical conditions.

Table 2.1: Different Related Works on Diagnosing Diseases Detection.

Ref	Methods	Dataset	Accuracy
[146]	Multi-model classification with VGG16 and SVM, Radial Basis function	Minimal dataset of X-ray images	95%
[147]	Neutrosophic classification with ResNet-50, VGG-16, XGBoost	ActualMed COVID-19 Chest X-ray, COVID-19 Radiography dataset	97.33%
[148]	ML classifiers, Decision Tree Explainer, Criteria Graph	Hospital Albert Einstein dataset (608 patients)	88%
[149]	Machine learning model for TB detection	1196 chest X-ray images from Kaggle	94%

Table 2.1: Different Related Works on Diagnosing Diseases Detection "Table Continued".

[150]	CNN with synthetic images (CovidGAN)	Real + Synthetic CXRs	95%
[151]	Hybrid deep learning CNN model using VGG16, VGG19, max and average pooling, various dense layers, and dropout	COVID-19 radiology database	92%
[152]	Serum Raman spectroscopy, PCA, SVM, KNN, RF	Serum samples from 88 patients	89.52%
[153]	ANN optimized by ABC, region fractal analysis	470 lung CT slides	92.37%
[154]	ML and radiomics features, Z-score heatmaps, ANOVA tests, SVM, KNN, EBM	378 CXR images	95.2%
[155]	Deep feature extraction with ResNet and VGG models, SVM classification	180 COVID-19, 200 normal chest X-ray images	94.7%

3. METHODOLOGY

3.1 INTRODUCTION

The employed methodology in the creation and assessment of the suggested pneumonia diagnosis model is outlined in this chapter. It starts with a summary of the key techniques and processes used in this study, then goes into great detail into the processes used to build and refine the diagnostic model. The chapter is organized to give readers a clear grasp of the procedure, beginning with an in-depth discussion of the dataset that was used to train and test the model. It delves deeper into the methods used for data preparation in order to get the dataset ready for analysis. The chapter describes the data splitting technique used to divide the dataset into training, validation, and test sets after data preparation, guaranteeing a reliable assessment of model performance. The modelling stage, which defines and fine-tunes the CNN's architecture and parameters, is the central component of the technique. A hybrid optimization strategy combining Fick's Law Optimizer (FLA) and Dung Beetle Optimization (DBO) is presented to improve model performance. The optimization methodologies used are explained in detail. The chapter seeks to provide a clear and methodical way to create a highly successful pneumonia diagnosis model through this organized approach.

3.2 PROPOSED PNEUMONIA DIAGNOSIS MODEL

A structured process follows, as shown in the flowchart of Figure 3.1, to accurately detect pneumonia from CXR images using our proposed model for diagnosis of infection with pneumonia. The model starts with preprocessing, where images in the chest X-ray dataset are converted from BGR to RGB. The images are resized, normalized, and one-hot encoded to be suitable for training. After preprocessing, The dataset was partitioned into training, validation, and test subsets. Convolution Neural Networks (CNNs), MobileNet and a composite of MobileNet with dung beetle optimization or Fick's law were developed using the training set so that they could be further improved. Similarly, once the models are trained, we store them and then check their performance on the testing dataset. Accuracy, precision, recall and F1-score are the evaluation metrics that guarantee an adequate measurement of how strong our model can perform. This process aims to create a powerful diagnostic AID that will help health professionals detect pneumonia from CXR images with high accuracy.

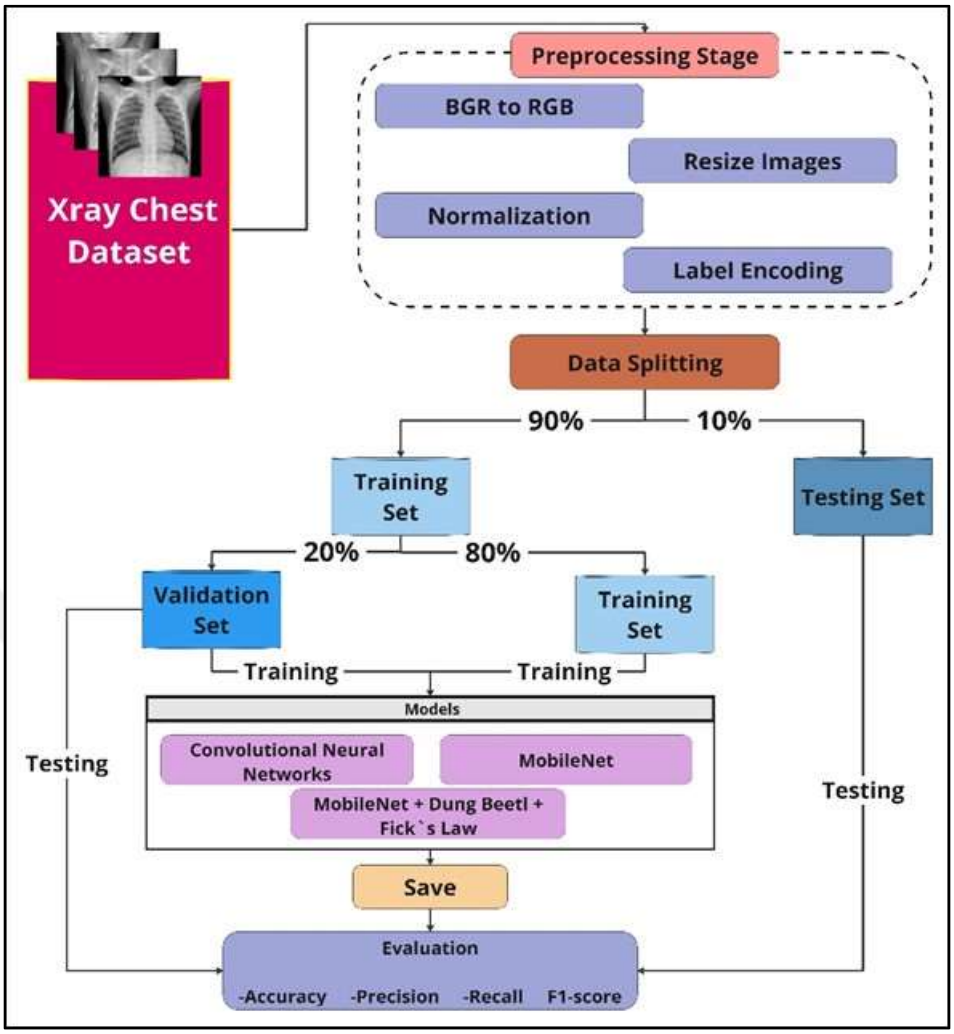


Figure 3.1: Proposed Pneumonia Diagnosis Model.

3.2.1 Dataset Description

The deep Learning Model used the CXR dataset containing 7,750 images from Kaggle. The deep learning model dataset comprises 7,750 CXR images categorized into two main classes: PNEUMONIA and NORMAL. There are an equal number of images per category. It is necessary to have some balance in the model and prevent it from becoming partial towards one category, thereby helping our model discern between a healthy lung image and pneumonia marks. The images also provide invaluable analytical clues on lung patterns that help detect multiple pathologies. The PNEUMONIA image shows a typical situation of increased lung opacities reflecting the presence of fluid (consolidation) or other infection-related changes. In normal images, in contrast to typical clear, both lung fields are clear, and the major abnormalities have been ruled out, as shown in Figure 3.2.

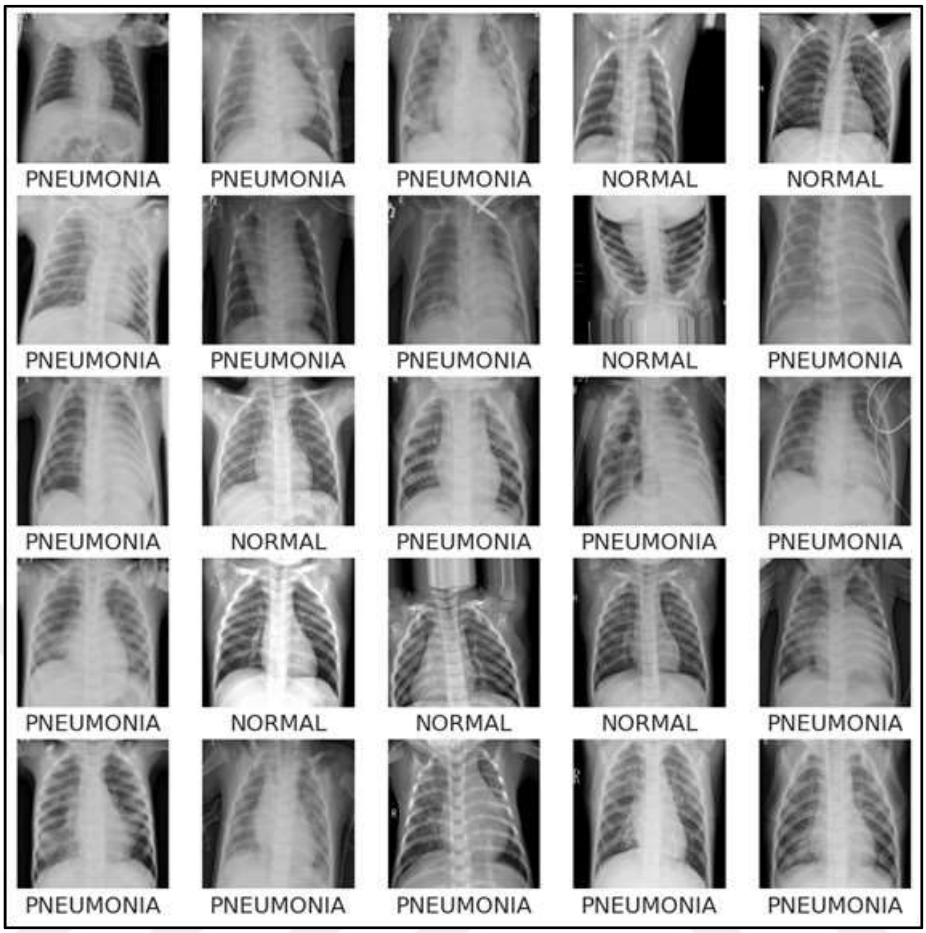


Figure 3.2: Dataset Sample Overview.

This is very important for interpreting the model because there are many images, and they must be of high quality to do successful critical feature extraction from it. The best quality images help the model to identify and correctly determine small changes between healthy lung tissue and abnormal tissues. Since this feature extraction process is quite detailed, it becomes necessary for us to have an efficient diagnostic tool to be able to capture changes related to pneumonia at an early stage. By training and performing tests on this dataset, the model aims to achieve a system capable of aiding doctors in making proper diagnoses to provide timely medication with an accurate treatment plan to help save lives.

3.2.2 Data Preprocessing

Pre-processing the CXR image dataset is crucial before training deep learning models. This process entails meticulously cleaning and structuring the data, ensuring it is in a usable format. Properly prepared data enhances the model's learning effectiveness by providing a

clean and organized foundation for algorithm training. By doing Exploratory Data Analysis (EDA) and applying different preprocessing steps will help to normalize the images, balance the dataset as well as improve model performance metrics. This section explains the systematic detailing of the approach used for data pre-processing, describing a robust framework that can be followed to develop a pneumonia diagnostic tool.

3.2.2.1 Image resizing

On the other hand, image resizing is a basic operation as part of our preprocessing pipeline, one on which we depend heavily to prep image data for training deep learning models. The selected image dimensions to resize (in this instance, 224 by 224 pixels) are determined by the variable `IMAGE_SIZE`. This size in particular is popular, specially within computer vision where a medium-size image usually enough to preserve key features and yet not very computationally intensive. The resizing process is implemented using the OpenCV library, a robust tool for various image-related tasks. The function cv2.resize(image, IMAGE_SIZE) adjusts the image size by interpolating pixel values, which helps maintain the aspect ratio and preserves important visual information. This standardized resizing across the dataset is vital for improving the learning efficiency and performance of the model, as it ensures consistent feature scales for subsequent extraction and analysis.

3.2.2.2 Converting image

In the majority of cases, when working with deep learning models, we need our images in RGB format. An important requirement to ensure that they are is converting them from BGR before feeding it into those algorithms as a preprocessing step. OpenCV uses BGR (Blue, Green, Red) as a default mode, while most machine learning frameworks should use RGB ([Red], [Green], [Blue]) instead, and the change can be seen in Figure 3.3. This is implemented by the CV2 function in OpenCV. cvtColor (image, cv2. COLOR_BGR2RGB). This task involves converting the image's color channels from BGR to RGB, maintaining its visual accuracy, and aligning it with the model architecture's expected input. Performing this alignment during both training and inference is crucial, as it teaches the model to effectively utilize pixel values in images to comprehend their representation of the training data. The use of color representation standardization in this step helps add consistency and efficiency to the overall image preprocessing pipeline.

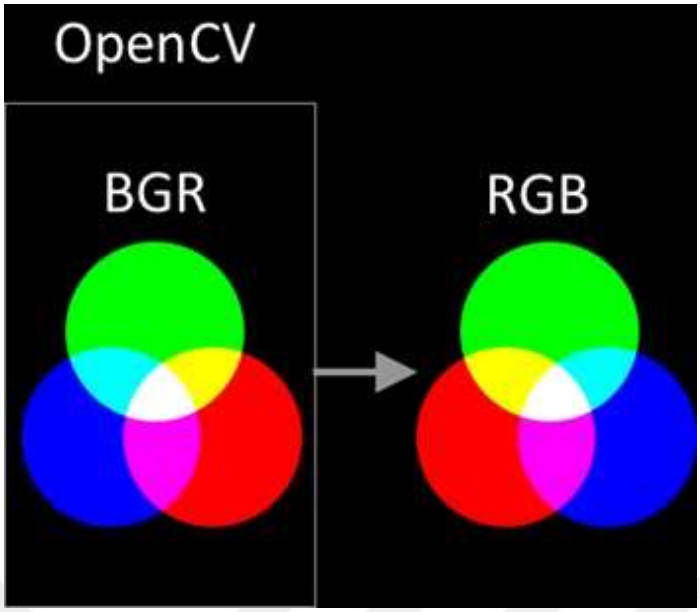


Figure 3.3: Difference between BGR and RGB.

3.2.2.3 Normalization

A very important part of data preprocessing is normalization, which scales pixel values to a range between 0 and 1, making them converge faster during training. By performing this step before any other processing, we standardize the input data and ensure it doesn't undergo any additional transformations that could potentially stabilize and accelerate our learning process. We commonly do normalization by dividing each pixel value with the maximum possible pixel value (usually 255 for an 8-bit image), thereby converting them to the [0,1] scale. This allows for faster gradient descent optimization and avoids numerical instability that can occur due to exploding gradients while training deep learning models. Normalizing the Pixel values will enhance the accuracy of our model by enabling it to process data from a limited and consistent range of datasets, thereby aiding in the training of models to better understand feature patterns over time.

3.2.2.4 One-hot encoding

One hot encoding is a key piece in data preprocessing, mainly preparing class labels for machine learning models. This method binarizes the categorical labels of each class, treating them as a binary vector, with one representing the category at the *i-th* position and zero for all other array values. This conversion is required during model training to interpret class labels appropriately. For this conversion, we utilize the to_categorical function from the

Keras library, which converts the numerical labels of the training, validation, and test sets into a one-hot format. For example, if there are three classes, then Label 0 will convert to [1, 0, 0], '1' to [0, 1, 0], and '2' to [0, 0, 1]. This encoding gives the model a larger learning space, indicating that each class is independent of the others and warrants separate treatment. One hot encoding aids in clearly classifying the output, leading to a more accurate classification of our final model.

3.2.3 Data Splitting

The data splitting process for the chest X-ray image dataset ensures a balanced distribution of images across the training and test sets. Initially, the dataset comprises 7,750 images, evenly divided into 'PNEUMONIA' and 'NORMAL' labels. These images are organized within folders corresponding to their labels. Data splitting (training, testing, and validation) as the first step makes our raw data suitable for model training and evaluation. Specifically, training consists of 90% of the data, and testing is done on the remaining 10%; the training data is split again into 80% training and 20% validation, as presented in Figure 3.4. Therefore, this stratified splitting method contains a uniform proportion of 'PNEUMONIA' and 'NORMAL' images in both the training dataset and the testing dataset. The blue bars are the training set distribution, while the orange ones refer to the test data distribution; notice we have a balanced representation of both classes on each side. This stratified split is very important to keep the model from learning any bias and evaluate how well it will perform robustly.

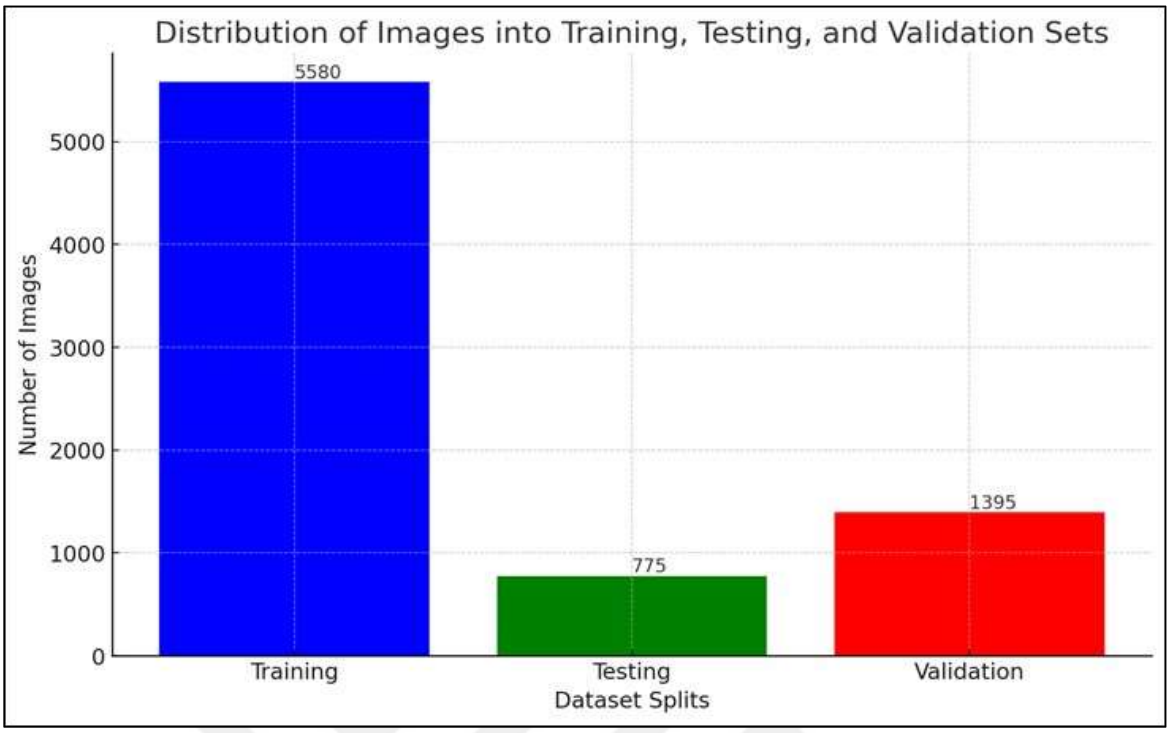


Figure 3.4: Distribution of Images into Training, Testing, and Validation Sets.

3.2.4 Modelling

Deep learning models, especially CNN and MobileNet, have significantly transformed the field of image recognition and analysis. CNNs are popular for tasks such as image classification and pixel-wise segmentation object detection. MobileNet is a light and efficient variation of convolutional neural networks (CNNs) designed for mobile and embedded vision applications, while MobileNet is extremely computationally efficient. In the last decade, deep learning models have played an important role in revolutionizing medical imaging and detecting diseases from X-ray and CT images, to name a few.

3.2.4.1 CNN architecture

Figure 3.5 shows the architecture of a CNN with one input layer where we select images having dimensions (224 x 224) along three color channels, which are used for learning weights during forward and backward propagations iteratively. The first layer of this architecture is the input layer, which then flows into the 1st convolutional layer. The first convolutional layer consists of 32 filters, each with a size of 3x3 and activated by the ReLU function to introduce non-linearity back into our model. After this max-pooling layer is applied with a kernel of (2, 2) to decrease the spatial dimensions of feature maps, it also

reduces the computational cost. The number of filters in this structure is equal to 32, the same size as the following convolutional layer and another max-pooling with a window size of 2x2. A second pooling operation After the other convolution, and then to add a dropout layer with an 80% dropout rate in order to avoid overfitting. The dropout layer is able to randomly deactivate a large number of the feature detectors during training, forcing the network to learn more powerful features. The network output after the convolutional and pooling layers is flattened into a one-dimensional array to assist it in moving directly onto becoming input to the fully connected neural network segment. This part contains a 128 neutron dense layer and a ReLU activation function for non-linearity. Finally, the architecture finishes with an output layer of 2 neurons (one for each class prediction). This layer uses a softmax activation function to obtain the predicted probability distribution of classes.

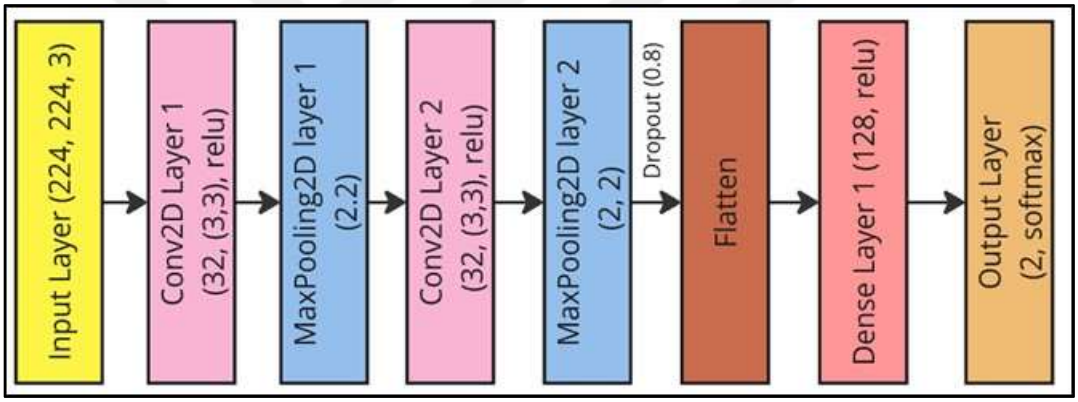


Figure 3.5: Proposed CNN Architecture.

3.2.4.2 MobileNet architecture

The MobileNet architecture, depicted in the provided diagram, employs a pretrained MobileNet model as the backbone. MobileNet is especially favored for its efficiency in mobile and embedded applications, utilizing depthwise separable convolutions. These convolutions significantly reduce the number of parameters compared to traditional convolutional layers, making it lightweight without compromising on performance. In this specific setup, the MobileNet model is customized for an input shape of 224x224 pixels across three color channels, ideal for processing color images. The pretrained model is configured to not include the top layer, enabling the addition of custom layers tailored for specific tasks. Here, it uses average pooling. The architecture extends with two dense layers.

The first dense layer has 1024 neurons and includes a dropout of 0.1 to prevent overfitting, followed by ReLU activation for non-linearity. A second dense layer with 512 neurons features a higher dropout rate of 0.5, again with ReLU activation. This configuration is designed to enhance the model's capability to learn complex features without overfitting.

The output layer consists of two neurons with a softmax activation function, which is typical for binary classification tasks. This layer outputs the probability distribution over the two classes, facilitating effective and interpretable classification. As shown in Figure 3.6, this architecture builds upon MobileNet by carefully designing additional layers and utilizing dropout methods to enhance classification performance.

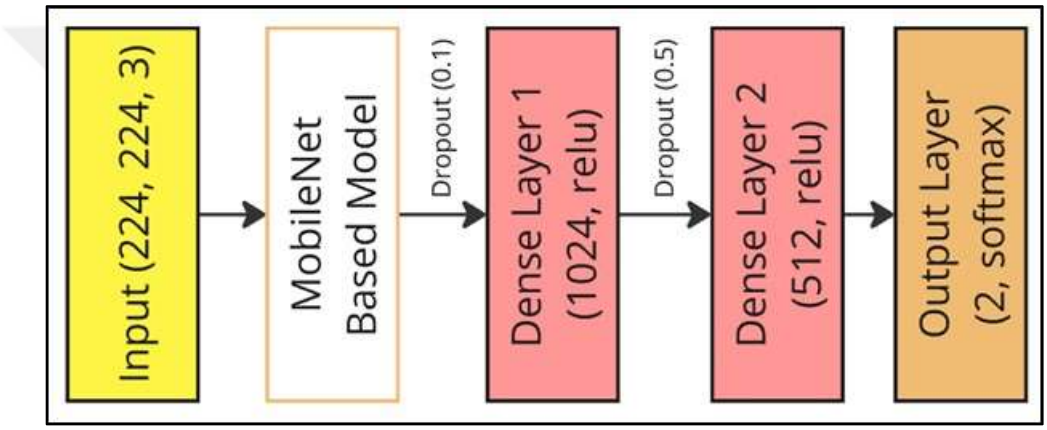


Figure 3.6: The Proposed MobileNet Architecture.

3.2.5 The Hyper Optimization DBO and FLA

This model demonstrates an innovative way to improve the efficiency of a CNN for image classification purposes by merging optimization approaches inbuilt parallelly, which is purported based on Dung Beetle Optimization (DBO) and Fick's Law Optimizer (FLA). The CNN is based on the MobileNet architecture, and pre-trained weights were used during fine-tuning, following a transfer learning style; this approach allows us to start with layers that have good feature extraction capabilities and add dense layers at the end of it so that it becomes specialized for our two class classification tasks. The optimization framework strives to make four specific hyperparameters more effective: the number of neurons in dense layers, dropout rate, learning rate and batch size. These parameters have a high-stake role to play in terms of the model learning and generalization processes. The loss function in the optimization process is to train a CNN on data and compute its performance as categorical

cross-entropy. This initialization is performed by randomly generating a population of candidate solutions in the bounds before optimising each parameter. The two complementary optimization techniques, DBO and FLA, will be recursively refined in this population. The DBO mechanism enables chaos by imitating the foraging behaviour of dung beetles and thus affords intensive searching in the solution space. By using a step size that reflects the difference between solutions and the best-so-far solution, we can pull a search towards better regions of state space while maintaining some exploration.

After the exploration, FLA provides exploitation by gradient-based change of solutions due to the fitness landscape (modelled in a way reminiscent of Fick's laws of diffusion). The dynamics of the proposed model are based on relocating states toward those with higher fitness, weighted by a decreasing attractiveness parameter that a bigger tint between their fitness implies less relocation, leading to exploration-exploitation balancing. The Optimization of multi-levels combines the processes, iteratively improving target performance one solution at a time via objective function assessment. Whichever configuration yields the best loss minimization performance is the optimal CNN adoption. The coupling of DBO with FLA guarantees a strong search strategy that successfully searches over the complicated, high-dimensional parameter space, yielding an efficient and widely general model. Algorithm 1 sketches the method of hybrid optimization based on Dung Beetle Optimization (DBO) and Fick's Law Optimizer (FLA). This method aims to find the best parameters for a given objective function within constraints in fewer evaluations.

The basic idea starts with a population of candidate solutions (individuals) represented by their characteristics, such as value and parameters, which, given the initial condition, are randomly generated within the domain limits. These candidates are an assessment of the problem through a function objective, and they will indicate in their aptitude if each one is, in other words, could be considered good or bad. Optimization is generally divided into two steps: Exploration and Exploitation. We leverage DBO to help us explore the search space. This requires changing the location of each individual as per the best solution which will lead to global exploration by moving the population towards potential regions. This procedure keeps every position inside the predetermined parameter range designed to represent valid configurations of parameters. Once the exploration is done, the algorithm

switches to the exploitation phase with the help of FLA. The group results discovered during the exploration phase are then honed further into the best solutions. The FLA method enhances local search capabilities by adjusting individual progeny and initial performance. This adjustment allows the FLA to concentrate on exploring all potentially optimal solutions. After updating, the population is re-evaluated to determine if there has been any improvement. During the iterations, balletic is devoted to saving the pupil resigned and any new atonement that further improves them. Mechanisms for balancing global and local search allow the algorithm to navigate the optimization landscape by exploring and exploiting iteratively. In the end, the algorithm deduces and returns a set of sensitive values for corresponding parameters, assuring that the optimal solution is performed. This methodology combines the strengths of DBO and FLA to provide a complete solution for complicated optimization problems.

Algorithm 1: Detailed Hybrid DBO and FLA Optimization Process

1: Input: Population size N, parameter bounds parameter_bounds, objective function obj_func, maximum iterations max_iterations

2: Output: Best found parameters and corresponding loss

3: Initialize population with N individuals randomly within parameter_bounds

4: Evaluate fitness of each individual in population using obj_func

5: Identify best_idx with the lowest fitness

6: for iteration ← 1 to max_iterations do

7: Exploration phase using DBO

8: for i ← 1 to N do

9: Apply DBO update rule to population[i] using the best individual population[best_idx] See Eq. (2.1)

10: Ensure population[i] respects parameter_bounds by clipping or wrapping

11: end for

12: Evaluate fitness of the updated population

13: Sort population by fitness ascending

14: Exploitation phase using FLA

15: for i ← 1 to N − 1 do

16: if fitness[i] > fitness[i + 1] then Condition for FLA interaction

17: Apply FLA update between population[i] and population[i + 1] based on their fitness See Eq. (2.2)

18: end if

19: end for

20: Re-evaluate fitness of the modified population

21: Update current_best_idx if a new lower fitness is found

22: if fitness[current_best_idx] < fitness[best_idx] then

23: best_idx ← current_best_idx

24: Update global best solution with population[best_idx] and its fitness

25: end if

26: Optionally record history or intermediate outputs for analysis

27: end for

28: return population[best_idx], fitness[best_idx]

The proposed flowchart represents this advanced hybrid optimization algorithm, which integrates elements from Fick's Law Algorithm (FLA) and the Dung Beetle Optimizer (DBO). Figure 3.7 presents the stages involved in the proposed hybrid optimization approach. The approach begins with initialization, followed by calculating the fitness values for each agent to assess solution quality. The agents are then divided into groups, N1 and N2, for diversified exploration. A transition factor TF^t, determines the phase each group will enter, dynamically adjusting their movements to optimize convergence. Three phases are outlined: Diffusion (DO), equilibrium (EO), and steady-state (SSO). Agents move based on concentration gradients in the DO phase, encouraging broad exploration. In the EO phase, exploration and exploitation are balanced, fostering solution diversity and avoiding premature convergence. In the SSO phase, agents stabilize, aiming to refine their positions in local optimum zones. The algorithm also includes a mechanism to handle boundary constraints, employing distinct equations for obstacle-free and obstacle-mode environments,

enhancing robustness across diverse problem landscapes. This iterative process repeats until all iterations are complete, after which the optimal solution is outputted. This structured integration of DBO and FLA strategies effectively enhances the algorithm's exploration-exploitation balance, potentially yielding more accurate and reliable solutions in complex optimization problems.

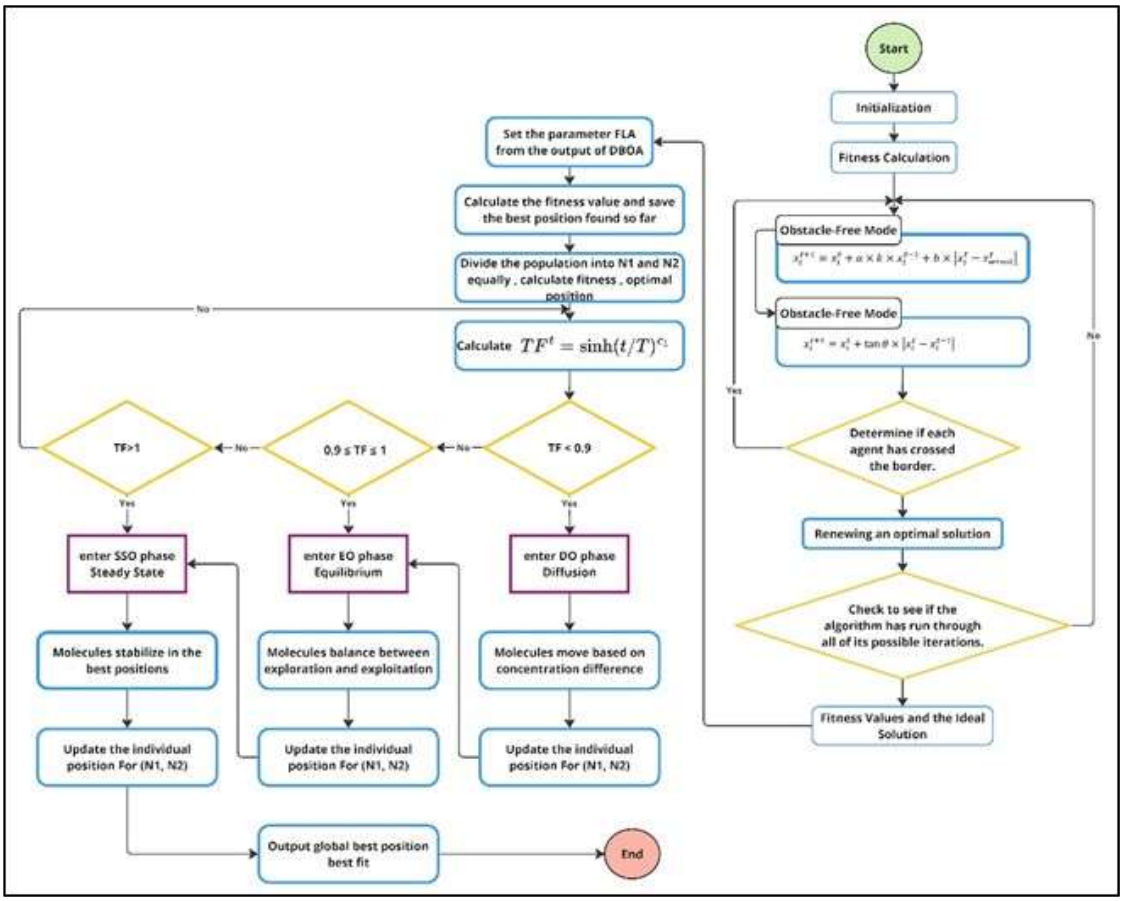


Figure 3.7: Detailed Flowchart Illustrating the Sequential Stages and Decision-Making Process of the Hybrid Optimization Algorithm.

4. RESULTS AND DISCUSSION

4.1 INTRODUCTION

This chapter presents a comprehensive evaluation of the experimental results from this study, which focused on the classification of medical images using various deep-learning models. We will go through a step-by-step evaluation for each model in specific detail, focusing on accuracy, precision, recall, and F1-score, which are the primary metrics to evaluate the performance of any classification task. The discussion delves into the detailed outputs of the Convolutional Neural Network (CNN) model, the MobileNet model, and the enhanced MobileNet model that utilizes Dung Beetle Optimization (DBO) and Fick's Law Optimizer (FLA). Another point we emphasize is the comparison of these models to provide insights into how different optimization strategies contribute to classification performance improvements. Finally, the results and discussions in this chapter validate our methods of interest and clarify the progress we have made by optimizing with new techniques.

4.2 EVALUATION METRICS

Evaluation metrics are critical for assessing the performance of machine learning models across various tasks. There are three main types: classification metrics, which measure how well models classify data; uncertainty estimation metrics, which measure how confident you are in your predictions; and qualitative evaluation methods, which give you more information about the situation by using case studies or expert opinions. Together, these metrics offer a comprehensive understanding of model performance, guiding researchers and practitioners in making informed decisions and improvements.

4.2.1 Classification Metrics

In this study, we trained and validated the DL models to assess their performance in coverage detection on the testing dataset. We further evaluated the models by comparing them using five performance matrixes: accuracy, sensitivity (recall), specificity, precision, and F1-score. Table 4.1 provides details on these metrics, which together offer a full assessment of the performance of each model in classifying images. In effect, these metrics contribute to a much better understanding of the efficiency of a model, answering different questions related to its predictive power. Accuracy refers to the overall correctness of the model across all

Classes. Precision and recall turn critical in cases in which there can be very different costs associated with false positives versus false negatives. The F1-Score is an averaged measure of precision and recall, which provides a balanced view of model performance in most cases, particularly for unbalanced datasets. Confusion matrices aim to deconstruct the model's predictions, providing a deeper understanding of the model's performance across various classes. Each of the metrics has its own specific purpose but provides great value in being able to understand a predictive model.

4.2.1.1 Accuracy

This measurement Computes the frequency at which predictions are equal to labels. This metric generates two local variables, total and counts, which are used to calculate the frequency at which the output value of a model corresponds to the given real value. The frequency is expressed as binary accuracy. It is mathematically represented as:

$$\boldsymbol{Acc} = \frac{\boldsymbol{Number\ of\ Correct\ Predictions}}{\boldsymbol{Total\ Number\ of\ Predictions}} \tag{4.1}$$

In this respect, accuracy is particularly useful as a measure of overall model performance in problems with balanced classes. On the other hand, accuracy might be misleading for problems with unbalanced classes since it can reflect the distribution rather than how good the model is at identification.

4.2.1.2 Precision

Precision also referred to as positive predictive value, is the proportion of true positives against all predictions forecasted as positive. It is measured as a ratio of true positive predictions to total predicted positives, which comprise true and false positives. The equation for precision is:

$$\boldsymbol{Precision} = \frac{\boldsymbol{True\ Positives}}{\boldsymbol{True\ Positives + False\ Positives}} \tag{4.2}$$

It is particularly useful when the cost of false positives is high, like in spam detection or disease screening, where false alarms can be expensive or worrisome.

4.2.1.3 Recall

Recall, or sensitivity, depends on how many relevant cases a model can retrieve in a given dataset. Formal definition: Ratio of true positives to actual positives, the latter being their sum with false negatives:

$$\boldsymbol{Recall} = \frac{\boldsymbol{True\ Positives}}{\boldsymbol{True\ Positives + False\ Negatives}} \tag{4.3}$$

Recall is important when a positive example is missing, which is much worse than a false positive, as in fraud detection or cancer diagnosis.

4.2.1.4 F1-Score

The F1-Score is calculated as the harmonic mean of Precision and Recall, providing a balanced measure that penalizes excessive results. It is particularly advantageous when there is an imbalanced class distribution, and you must balance Precision and Recall. The F1-Score is calculated using the following formula:

$$\boldsymbol{F1 - score} = \mathbf{2} * \frac{\boldsymbol{Precision * Recall}}{\boldsymbol{Precision + Recall}} \tag{4.4}$$

4.2.1.5 Confusion matrix

A confusion matrix is a table that describes the performance of a classification model over a test data set, where the true values are known. Hence, it describes the performance of an algorithm in a pleasing visual form. Each row of instances represents the instances of an actual class, and each column represents the instances in a predicted class. The matrix aids in comprehending the classifier's errors and, crucially, identifies the nature of these errors. Figure 4.1 illustrates the specifics of the confusion matrix.

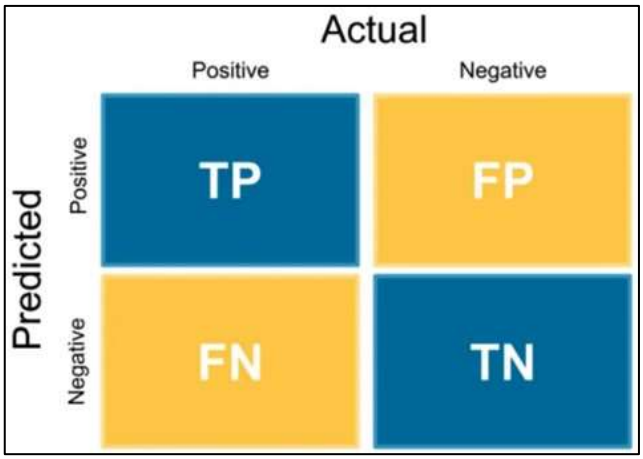


Figure 4.1: Confusion Matrix.

For the classification task involving normal versus pneumonia patients, the evaluation metrics are computed based on four key categories: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). In this context:

a. True Positives (TP) represent the number of pneumonia images correctly identified as pneumonia.

b. True Negatives (TN) denote the number of normal images correctly identified as normal.

c. False Positives (FP) indicate the number of normal images incorrectly classified as pneumonia.

d. False Negatives (FN) refer to the number of pneumonia images mistakenly classified as normal.

Table 4.1: Evaluation Metrics.

Metrics	**Equation**
Accuracy	$\frac{TP + TN + FP + FN}{\text{TP} + \text{TN}}$
Sensitivity /Recall	$\frac{TP}{TP + FN}$
Specificity	$\frac{TN}{TN + FP}$
Precision	$\frac{TP}{TP + FP}$
F1-score	$\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

4.2.2 Uncertainty Estimation

The area of uncertainty estimation is very important within model evaluation, as it gives the extent of the uncertainty related to machine learning predictions. It tries to compute the confidence of a model by going through the statistical process of uncertainty estimation, hence bettering decision-making where risks and reliability are on the table. Such techniques include confidence intervals, which supply the statistical range for answers, and Bayesian methods, which rack in prior knowledge for uncertainty. Appropriate uncertainty evaluation can also be useful in building up model interpretability so that the users can understand the predictability impulse. In cases where further investigation is needed, they know where to look.

4.2.2.1 Confidence intervals

Confidence intervals are normally reported at levels such as 95% or 99%. They are, however, very important in the evaluation of the uncertainty of a deep learning model's predictions. A 95% confidence interval means that if this process is conducted many times, about 95% of the calculated intervals would contain the true parameter value. In other words, this means that, in a practical sense, there is a 95% certainty that a model's predicted values are within a certain range from the true outcome. The 99% confidence interval would, of course, imply an even higher degree of certainty; in other words, it means that 99 out of 100 times, the true parameter would fall within this wider interval.

This has deep implications for the performance of models and decision-making processes. For example, in medical diagnostics, a model predicts the probability of disease and can provide a 95% confidence interval meaning health professionals can estimate the risk with high assurance, hence informing treatment options and patient management strategies. The larger the interval, as seen with a 99% confidence level, could mean a more conservative estimate, which might be appropriate in cases where incorrect predictions have severe consequences.

These confidence intervals may be calculated in many ways. Bayesian Neural Networks directly output a distribution over the predictions from which one may derive a confidence interval by using the spread of the predicted values. Monte Carlo Dropout approximates the intervals by making many forward passes through the model with dropout active, capturing

the prediction variance. Ensemble methods combine the predictions from many models, calculating their standard deviation and thereby letting the model compute intervals based on the range of outputs.

4.2.3 Qualitative Evaluation

Qualitative Evaluation serves as an essential complement to traditional quantitative metrics, aiming to interpret the behavior of the model and understand the reasoning behind its decisions. In deep learning, the Grad-CAM (Gradient-weighted Class Activation Mapping) technique helps meet this need, significantly improving our understanding of how deep models, especially convolutional neural networks, make decisions. This technique works by analyzing the gradients related to the target class to generate a heatmap that highlights the areas most influential in the final outcome. By providing a visual representation of the parts that the model relies on, both researchers and users can gain deeper insights into the model's learning patterns and assess if it is focusing on the correct features or irrelevant elements.

One of the main advantages of using Grad-CAM in qualitative evaluation is its ability to reveal potential biases or incorrect learning within the model. For instance, if a model successfully diagnoses pneumonia from a chest X-ray, Grad-CAM can show whether it focuses on the relevant area, such as the shaded or congested areas indicating inflammation, or on unrelated parts like the chest boundary or unaffected sections of the lungs. This type of analysis helps ensure that the prediction is based on genuine medical indicators rather than incidental correlations. Thus, relying solely on quantitative metrics is insufficient to guarantee quality and effectiveness; qualitative evaluation is equally important to ensure that the model's decisions are founded on sound and reasonable grounds.

Moreover, the visual explanation offered by Grad-CAM is particularly significant in critical applications like healthcare and autonomous driving, where clear and justified decisions are required. In diagnosing diseases from chest X-rays, for example, the visual explanation helps doctors understand which regions the model is focusing on, allowing them to ensure that it aligns with sound medical knowledge. This interpretability can increase users' and practitioners' trust in deep learning models. The decisions made by AI systems are often complex and hard for humans to understand, which limits their adoption in high-stakes environments. Techniques like Grad-CAM can help enhance transparency and build a more

trustworthy framework for models, allowing researchers to make necessary adjustments based on the visual analysis of the features used in the model's decisions.

4.3 EVALUATION RESULTS OF CNN MODEL

The training and evaluation results of the CNN model, as depicted in Figure 4.2, demonstrate a notable progression in accuracy a nd loss over the 25 training epochs. The model, optimized with the Adam optimizer at a learning rate of 0.001 and using categorical cross-entropy loss, showed steady improvement. The model continued to perform consistently, improving and ending at an accuracy of Training of 94.27 % and validation of 96,63%. The model generalizes to unseen data, a key for meaningful real-world applications. This is further supported by the loss curves that depict a gasp in training and validation loss, from initial 0.4242 and 0.4143 to final for train of 0.1484 and validation of 0.0887. The decrease in loss values means the model learns well from data. This shows how nicely it learns and generalizes without overfitting.

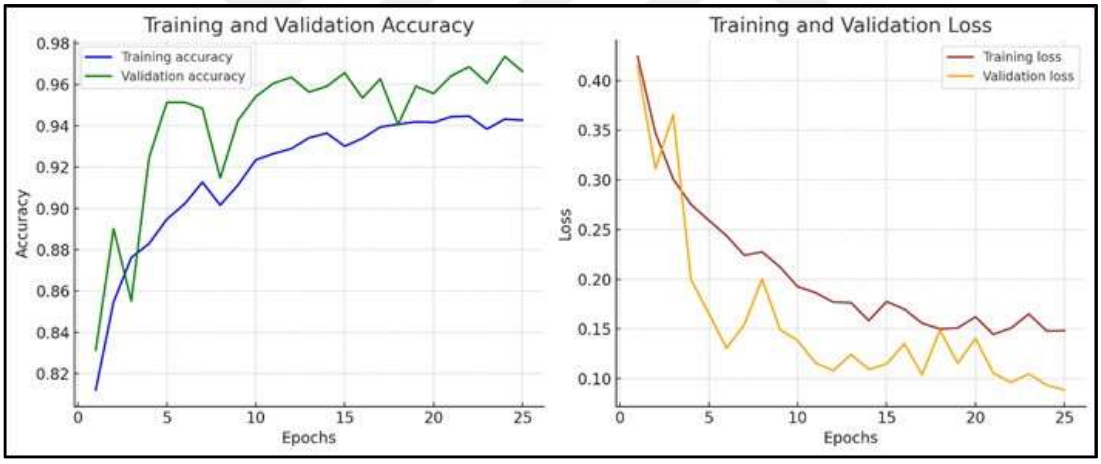


Figure 4.2: Accuracy and Loss of Cnn Models During Training and Validation.

The confusion matrix in Figure 4.3 is used to evaluate the CNN model and classify chest X-ray images into "NORMAL" and "PNEUMONIA" categories. The matrix shows that the model had successfully predicted 366 out of 388 normal cases and 373/387 pneumonia-positive images. Nonetheless, it had 22 healthy false positive cases, such as pneumonia, and 14 misclassified alienated infection cases. Thus, the results here clearly present a very high classification precision between these two conditions with minimal errors that point from one class to another. This follows the intuition that a good balance of true positives and true

negatives would say something about how successful this model is at doing well on predicting these data points.

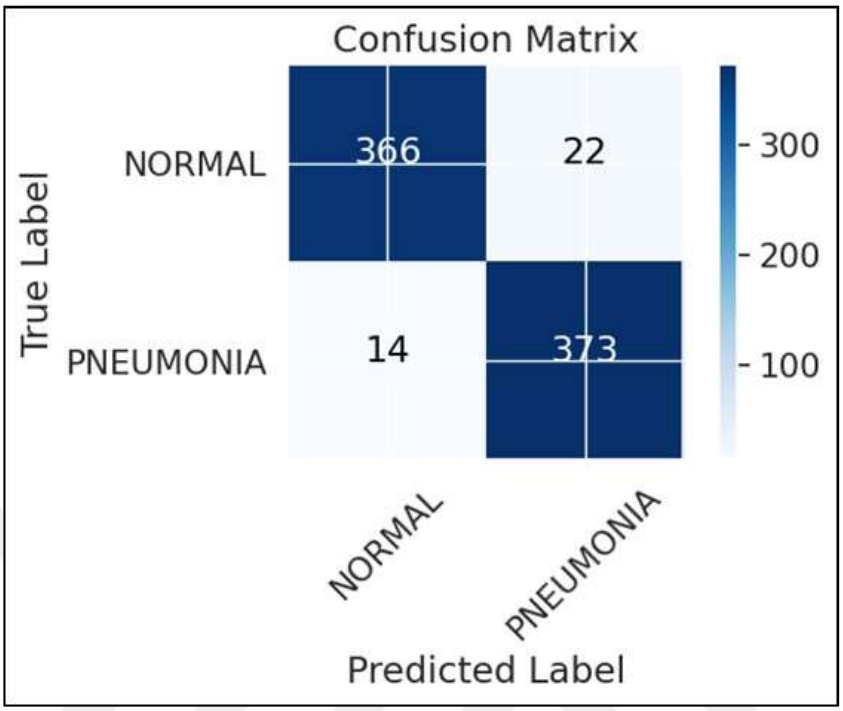


Figure 4.3: CNN Model Confusion Matrix.

Table 4.2 showcases the classification report of the CNN model, which exhibits good ability to separate NORMAL and PNEUMONIA classes. The precision, recall and F1-score for both categories are impressively high: NORMAL having 96 ± 1.38%, 94.38 ± 1.63% and 95 ± 1.53%. while PNEUMONIA has a confounding numbers precision of 94 ± 1.67%, Recall of 96.38 ± 1.32%, and an F1- score of 95 ± 1.53%. The model has an accuracy of 95.35 ± 1.48%, which means it could correctly classify the images with a high percentage of cases done. Overall, the precision is 95.37 ± 1.48%, 95.36 ± 1.48% recall, and 95.35 ± 1.48% f1-score balanced performance over both classes. The specificity metrics validate the model performance with NORMAL at 96.38±1.32% and PNEUMONIA at 94.33±1.63%. Conclusion This study has emphasized the ability of the CNN model to detect pneumonia in chest X-ray images, providing an important aid for assisted diagnosis and improved clinical decisions.

Table 4.2: Report on CNN Model Classification.

Class	Precision	Recall	F1-Score	Specificity
NORMAL	96 ± 1.38%	94.38 ± 1.63%	95 ± 1.53%	96.38 ± 1.32%
PNEUMONIA	94 ± 1.67%	96.38 ± 1.32%	95 ± 1.53%	94.33 ± 1.63%
Accuracy	95.35 ± 1.48%			
Overall	95.37 ± 1.48%	95.36 ± 1.48%	95.35 ± 1.48%	

4.4 EVALUATION RESULTS OF MOBILENET MODEL

Figure 4.4 illustrates the accuracy of training and testing, as well as a review of loss, over a period of up to 25 epochs, in the context of the MobileNet model architecture. Validation accuracy also has an upward trend by the final epoch, demonstrating close to 97.93% accuracy. This indicates that the model learns generalizable behavior from training examples to validation. The loss plot, on the other hand, shows that training and validation losses decrease by reducing epochs. The training loss decreases a lot initially in the epochs, reaching a minimum, and this then stabilizes at some low value, meaning that it is effectively minimizing its error on the data that was used for learning. The testing loss exhibits a similar behavior, albeit with higher variability and occasional downward trends. The variability seen in the testing loss plot can be attributed to the inherent variance of test data, but a decreasing trend overall shows that a model is increasing in strength. The MobileNet model did well on both the training and testing datasets, with very little loss. This means that it can very accurately tell the difference between chest X-rays that look normal and those that look like pneumonia.

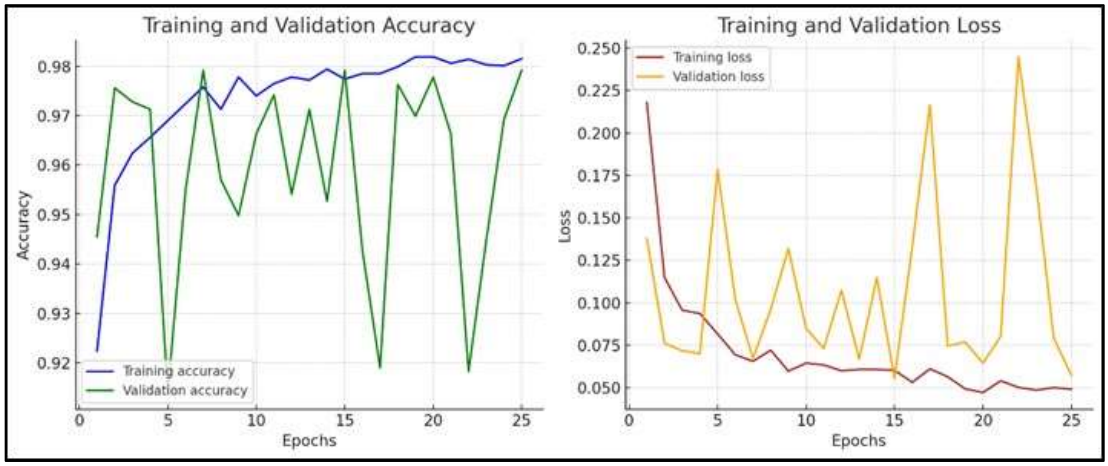


Figure 4.4: Training and Validation Accuracy and Loss of MobileNet Model.

Figure 4.5 shows the confusion matrix for the MobileNet model, which describes that its classification of normal and pneumonia chest X-ray images was used as a validation set. The matrix is for 388 NORMAL images; only 376 were correctly classified, and the remaining are misclassified as PNEUMONIA. 387 images of the PNEUMONIA class could be detected, with 383 correct and only 4 incorrect labels as NORMAL. This leads to high precision and recalls for both classes, which means the model is pretty confident in distinguishing between healthy and pneumonia-affected images. A smaller number of misclassifications demonstrates the extent to which MobileNet has proven effective in performing medical image classification using a dataset such as this one.

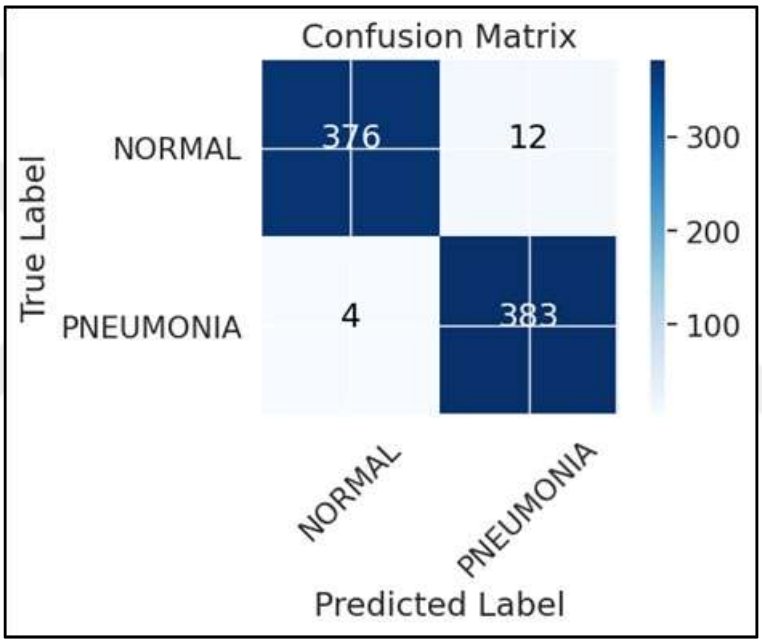


Figure 4.5: Confusion Matrix of MobileNet Model.

The classification report from the execution of the MobileNet model is depicted in Table 4.3 below. The model has significantly high values of precision, recall, and F1-score in both the CLASS NORMAL and PNUEMONIA. The values are approximately equal to 0.98, meaning that the MobileNet model has a strong ability to classify both normal and pneumonia cases successfully. More specifically, the NORMAL class had a 99 ± 0.70% precision, recall of 97 ± 1.20%, and F1-score of 98 ± 0.99%. The specificity value was 98.97 ± 0.71%. The PNEUMONIA class exhibited a precision of 97 ± 1.20%, recall of 99 ± 0.70%, and F1-score of 98 ± 0.99%, while the values for specificity were around 96.91 ± 1.22%. The overall accuracy rate of the model was 97.94 ± 1%, with an overall precision of 97.95

± 1%, recall at 97.93 ± 1%, and F1-score of 97.93 ± 1%. Therefore, the execution of the MobileNet model demonstrated highly robust diagnostic performance in differentiating between normal and pneumonia cases.

Table 4.3: Report on MobileNet Model Classification.

Class	Precision	Recall	F1-Score	Specificity
NORMAL	99 ± 0.70%	97 ± 1.20%	98 ± 0.99%	98.97 ± 0.71%
PNEUMONIA	97 ± 1.20%	99 ± 0.70%	98 ± 0.99%	96.91 ± 1.22%
Accuracy	97.94 ± 1%			
Overall	97.95 ± 1%	97.93 ± 1%	97.93 ± 1%	

4.5 EVALUATION RESULTS OF OPTIMIZED MOBILENET MODEL

Model evaluation: By utilizing Dung Beetle and Fick's Law optimization for parameter selection, MobileNet successfully selected efficient parameters to minimize loss. The results showed that the parameters optimized through integrated optimization yielded the best loss of 0.0571, highlighting the advantages of this optimization approach. The model displayed loss and accuracy improvements over five epochs in the training process for all trial runs. When one training session began, the model had an accuracy of 0.8878 and ended with 0.9746. Simultaneously, the loss parameter decreased from 0.2687 to its final value at the end of the epoch. Many trials accurately depicted this trend, allowing us to generalize the model using those parameters. When you use different biological optimization algorithms from the literature along with other powerful tools like Dung Beetle Optimization (DBO) and Fick's Law Optimizer (FLA), you can effectively explore the parameter space while saving a lot of time on computation. This rigorous optimization process not only identified the best parameters, but also yielded the lowest observed loss and highest accuracy metrics. The integrated optimization process yielded the best parameters as [373.0, 0.1244, 0.000291780989, 36.5608342] with an achieved best loss of 0.057161688804626465.

This method helps produce some of the lightest deep learning models, with a significant improvement in accuracy, when combined with Dung Beetle Optimization (DBO) and Fick's Law Optimizer (FLA) into integrated optimization of MobileNet. This demonstrates its viability in fine-tuning parameters to ensure peak performance. For the MobileNet model, optimized with Dung Beetle Optimization (DBO) and Fick's Law Optimizer (FLA), our

evaluation uncovers several interesting findings regarding its performance in both training and testing phases.

Figure 4.6 illustrates the training and testing's accuracy and loss over 25 epochs. The training accuracy consistently improves, starting at approximately 88% and reaching a plateau at around 98%. This trend indicates the model's ability to learn the training data effectively. The testing accuracy, which measures the model's performance on unseen data, closely follows the training accuracy, stabilizing between 97% and 98%. This close alignment suggests that the model generalizes well and is not overfitting. The training loss exhibits a rapid decline from an initial high value to below 0.1 within the first few epochs, eventually stabilizing around 0.05. This loss reduction indicates effective optimization and convergence of the model during training. The testing loss also exhibits a decreasing trend, albeit with greater fluctuations than the training loss. This variability in the testing loss is typically due to the inherent differences in the validation set.

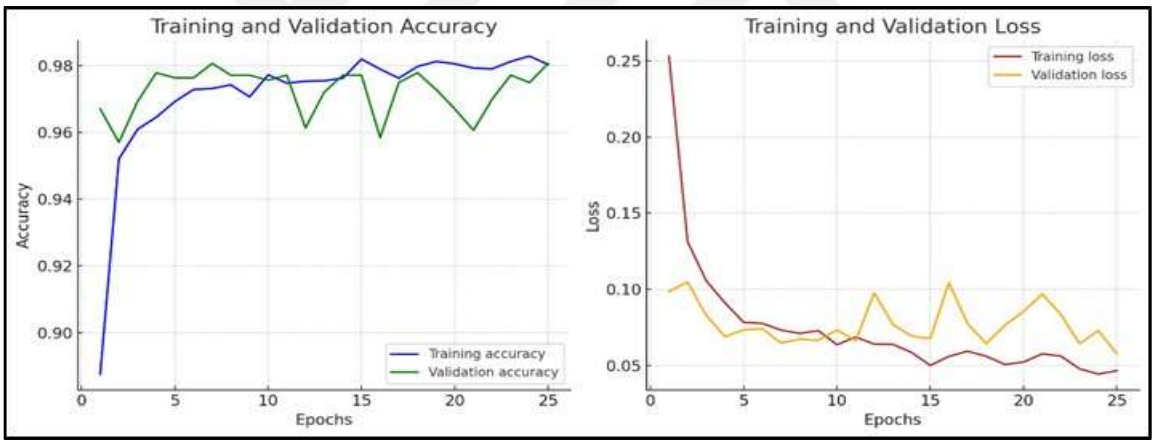


Figure 4.6: Accuracy and Loss of Optimization Model during Training and Validation.

Figure 4.7 presents the confusion matrix for the optimized MobileNet model utilizing DBO and FLA algorithms. The matrix illustrates the model's performance in classifying two categories: NORMAL and PNEUMONIA. Out of 388 NORMAL cases, the model correctly classified 375 instances and misclassified 13 as PNEUMONIA. Conversely, out of 387 PNEUMONIA cases, the model accurately identified 386 instances, with only one misclassification as NORMAL. This high level of accuracy in both categories indicates the model's strong ability to distinguish between NORMAL and PNEUMONIA cases, highlighting its reliability and robustness in medical diagnosis tasks.

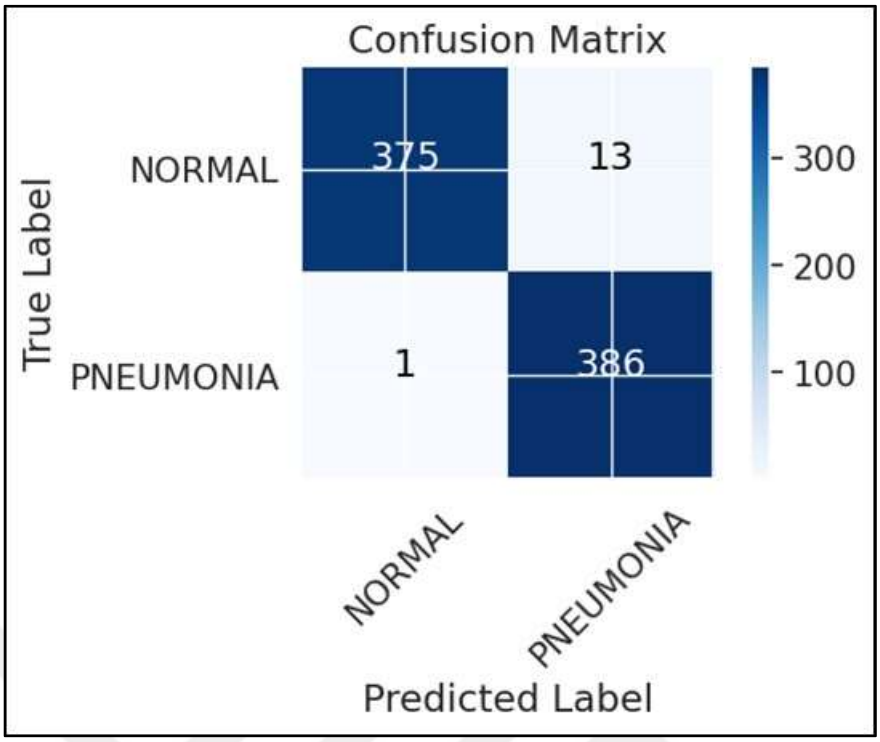


Figure 4.7: Confusion Matrix of the Optimization Model.

The classification report for the optimization model, incorporating MobileNet architecture combined with the Dung Beetle and Fick's Law optimization, is presented in Table 4.4. This report demonstrates the model's exceptional performance across various metrics. In particular, the model exhibits high accuracy in identifying normal cases, with a precision of 100% and an F1-score of 98 ± 0.99%, complemented by a recall of 97 ± 1.20%. These performance metrics are familiar, indicating a return to using balanced classes for pretraining. Additionally, the specificity values for the Normal class are 99.74 ± 0.31%, demonstrating the model's effectiveness in correctly identifying both true positives and true negatives. The PNEUMONIA class has a precision of 97 ± 1.2%, a recall of 100%, and an F1-score of 98 ± 0.99%, which supports our model's effectiveness in detecting pneumonia cases. In pneumonia vs. normal, the specificity of this class is 96.65 ± 1.27%, demonstrating that our model is highly trustworthy in confirming pneumonia cases compared to non-pneumonia fellows. The model achieves an accuracy of 98.19 ± 0.94%, with overall precision, recall, and F1-scores of 98.23 ± 0.93%, 98.19 ± 0.94%, and 98.19 ± 0.94%, respectively. These metrics collectively indicate the robustness and accuracy of the optimization model in classifying medical images, making it a valuable tool for diagnostic purposes.

Table 4.4: Report on Optimization Model Model Classification.

Class	Precision	Recall	F1-Score	Specificity
NORMAL	100%	97 ± 1.20%	98 ± 0.99%	99.74 ± 0.31%
PNEUMONIA	97 ± 1.2%	100%	98 ± 0.99%	96.65 ± 1.27%
Accuracy	98.19 ± 0.94%			
Overall	98.23 ± 0.93%	98.19 ± 0.94%	98.19 ± 0.94%	

4.6 COMPARISON RESULTS

The experiment results presented in Table 4.5 show the effectiveness of the different classification models, CNN, MobileNet, and Optimized MobileNet, regarding accuracy, precision, F1, sensitivity, and specificity. The CNN model had an accuracy of 95.35 ± 1.48%, with 96 ± 1.38% and 94 ± 1.67% precision for NORMAL and PNEUMONIA, respectively. Its F1-scores for 95 ± 1.53%, with 94.38 ± 1.63% and 96.38 ± 1.32% recall for NORMAL and PNEUMONIA, respectively. Which indicates excellent performance. The CNN has a specificity of 96.38 ± 1.32% for the NORMAL class. For PNEUMONIA, the former is 94.33 ± 1.63%.

MobileNet is the category with the best performance compared to CNN, with an accuracy of 97.94±1%. In the case of NORMAL, it extracted precision values to 99 ± 0.70% and PNEUMONIA with 97 ± 1.20%, which finally took F1-measures at 98 ± 0.99%; for NORMAL sensitivity and specificity resulted in 97 ± 1.20% and 98.97 ± 0.71%, respectively, PNEUMONIA got a better performance as it reached to the top of our detection rate with sensitivity equal to 99 ± 0.70%. Still, its specificity was slightly lower at 96.91 ± 1.22%. This result was further improved with the proprietary Optimised MobileNet. It had an accuracy of 98.19 ± 0.94%; in the case of NORMAL, it extracted precision values to 100% and PNEUMONIA with 97 ± 1.2%, which finally took F1-measures at 98 ± 0.99%, either through sensitivity or its specificity, which was excellent with 97 ± 1.20% and 99.74 ± 0.31%. This model had a precision of 97 ± 1.2%, a sensitivity level of about 100% for PNEUMONIA, and a specificity as high as 96.65 ± 1.27%. The performance improvements in the other model optimization effectiveness on model performance are in the full-size image.

Table 4.5: Comparison Models.

Metric	CNN	MobileNet	Optimized MobileNet
Accuracy	95.35 ± 1.48%	97.94 ± 1.00%	98.19 ± 0.94%
	NORMAL	NORMAL	NORMAL
Precision	96 ± 1.38%	99 ± 0.70%	100%
Recall	94.38 ± 1.63%	97 ± 1.20%	97 ± 1.20%
F1-score	95 ± 1.53%	98 ± 0.99%	98 ± 0.99%
Specificity	96.38 ± 1.32%	98.97 ± 0.71%	99.74 ± 0.31%
	PNEUMONIA	PNEUMONIA	PNEUMONIA
Precision	94 ± 1.67%	97 ± 1.20%	97 ± 1.2%
Recall	96.38 ± 1.32%	99 ± 0.70%	100%
F1-score	95 ± 1.53%	98 ± 0.99%	98 ± 0.99%
Specificity	94.33 ± 1.63%	96.91 ± 1.22%	96.65 ± 1.27%

Table 4-5 compares the proposed CNN and MobileNet architectures, leading to their higher classification accuracy than others stated in the literature for detecting pneumonia. CNN model accuracy: 95.35 ± 1.48%. Our results for precision, recall and F1-scores were consistent across different categories, showing balanced performance of the various classes in predicting pneumonia cases using this model. Post-training fine-tuning MobileNet showed an accuracy of the improved 97.94 ± 1% with higher precision, recall and F1 scores, showing great efficiency in model performance advancements. MobileNet was further optimized with a hybrid optimization strategy, which raised the mean accuracy from 97.94 ± 1% to a slightly higher 98.19 ± 0.94%. It performs well on all metrics compared to other approaches. In other words, this indicates how much of an impact precise training strategies can have on model speed and accuracy, especially in medical imaging tasks.

Figure 4.8 provides a comparative analysis of the performance metrics for three classification models: the CNN Model, the Transfer Learning Model (MobileNet), and the Optimized MobileNet Model. These results underscore the superior performance of the Optimized MobileNet Model, demonstrating the significant improvements achieved through optimization.

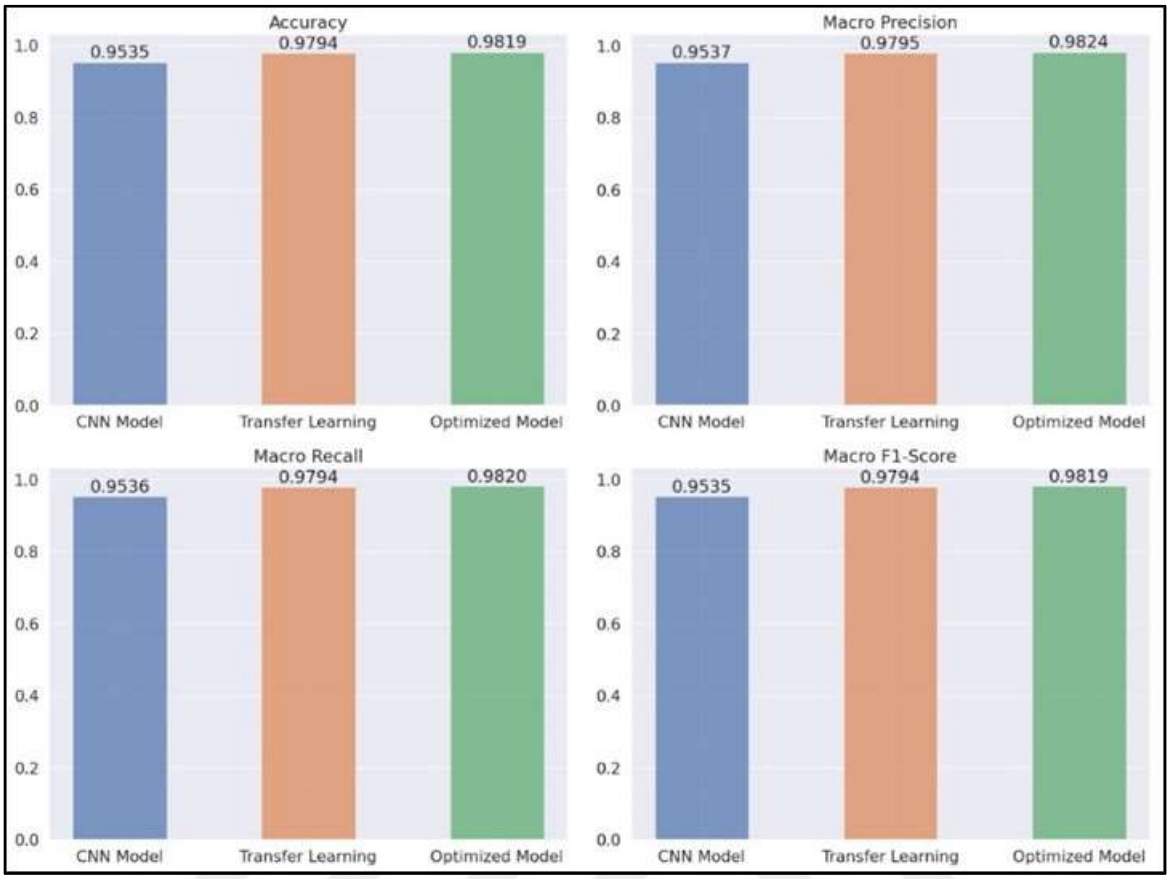


Figure 4.8: Comparative Analysis of the Performance Metrics for Three Classification Models.

Table 4.6 presents a comparative analysis of the proposed pneumonia detection model with other existing methods. The proposed model, which combines CNN and MobileNet architectures along with advanced optimization techniques such as dung beetle optimization and Fick's law, achieves superior results with an accuracy of 98.19%, precision of 98%, recall of 98%, and an F1-score of 98%. These results significantly outperform prior models, such as the machine learning model for tuberculosis detection [149], which achieved 94% accuracy, and the hybrid CNN model for COVID-19 diagnosis [151], with an accuracy of 92%. The proposed model also surpasses the SVM-based approach for pSS-ILD diagnosis [152], which reached an accuracy of 89.52%; the model was based on artificial Neural networks and improved by the artificial bee colony algorithm to detect Covid-19 from CT images, where it achieved an accuracy ratio of 92.37% [153], and the ensemble bagged model for COVID-19 detection [154], which achieved 95.2%. The consistent improvement across all evaluation metrics highlights the robustness and effectiveness of the proposed

model, positioning it as a more reliable and accurate diagnostic tool for pneumonia detection using chest X-ray images.

Table 4.6: Comparison of Our AI Model with Established Techniques in Pneumonia Detection.

Ref.	Method	Accuracy
[149]	Machine learning model for TB detection using chest X-ray	94%
[151]	Hybrid CNN model for diagnosing COVID-19 using VGG16 and VGG19	92%
[152]	SVM (poly) model for diagnosing pSS-ILD using Raman spectroscopy	89.52%
[153]	ANN with ABC for region-based fractal analysis on lung CT	92.37%
[154]	Ensemble bagged model for COVID-19 detection using radiomics features	95.2%
Proposed	Hybrid model combining MobileNet with optimization techniques	98.19%

Figure 4.9 shows the optimized model's outstanding performance in instance classification. It is robust under all test conditions and can thus provide reliable results on high-stakes applications requiring accurate decision-making.

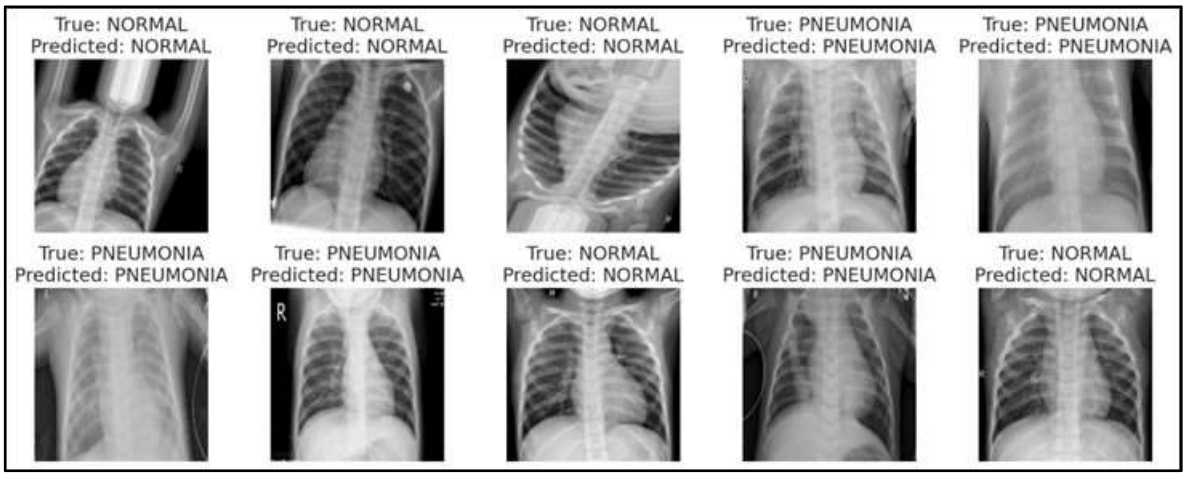


Figure 4.9: True vs Predicted class of Optimization Model.

Figure 4.10 shows a chest X-ray classified by a deep learning model. The image on the left is the original X-ray labeled as "NORMAL", while the right image represents the Grad-CAM heatmap used to illustrate the areas influencing the model's decision. The red and yellow

regions show the areas of greatest influence, while blue indicates less focus. The model primarily focuses on the central chest area, including the heart and trachea. This analysis helps improve transparency and trust in healthcare models by showing how decisions are made and identifying potential issues if irrelevant areas are highlighted.

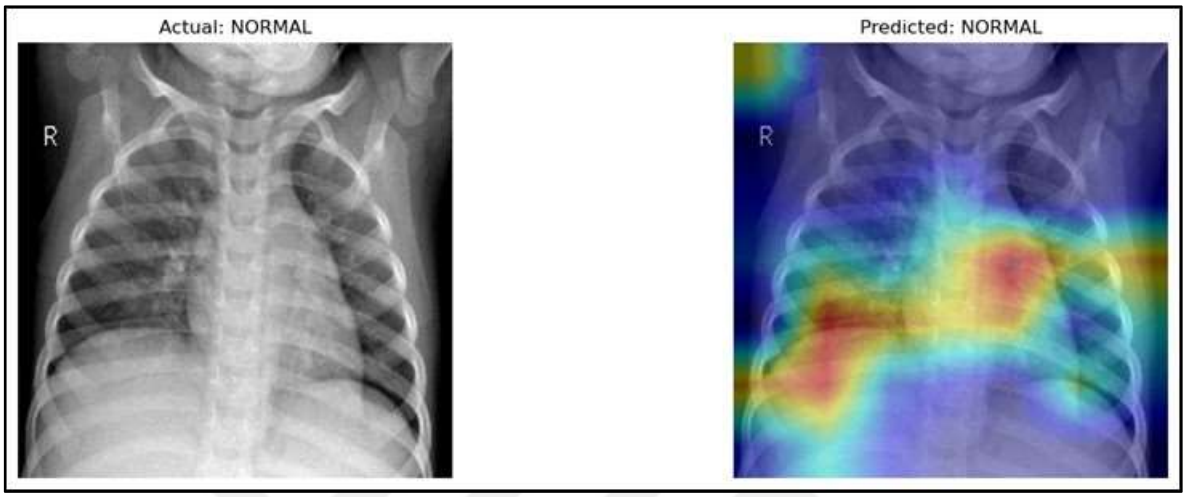


Figure 4.10: X-ray Actuals (NORMAL) and Model Predictions with Grad-CAM Heatmaps Highlighting Classification Influences.

5. CONCLUSIONS AND PROSPECTS FOR FUTURE WORK

5.1 CONCLUSIONS

Previous research in DL, particularly with CNNs and MobileNet models, has shown impressive results in recognizing pneumonia from X-ray images. These models accurately classified images as normal or pneumonia, crucial for efficient medical diagnostics during health crises. The CNN model achieved an overall accuracy of 95.35% with balanced precision, recall, and F1-scores for both classes. The accuracy improved to 97.93% with MobileNet architecture and reached 98.19% with the addition of a hybrid optimization method. These findings confirm the effectiveness of these models and highlight the potential of deep learning and optimization techniques to enhance diagnostic accuracy further.

5.2 FUTURE WORKS

Looking ahead, there are opportunities to expand on this research by incorporating multi-modal datasets and exploring additional metaheuristic algorithms to enhance AI capabilities in healthcare further. Using various optimization algorithms, such as Dung Beetle Optimization, and applying Fick's Law could offer new insights into algorithmic efficiency within digital image processing. These advancements could provide innovative approaches to complex diagnostics, pushing the boundaries of AI in medical image analysis and improving the diagnostic process for timely disease detection.

REFERENCES

[1] C. Sohrabi *et al.*, "World Health Organization declares global emergency: A review of the 2019 novel coronavirus (COVID-19)," *Int. J. Surg.*, vol. 76, pp. 71–76, 2020.

[2] W. H. Organization, "Weekly epidemiological update on COVID-19—29 June 2021, 46th edn." 2021.

[3] T. L. Sandri *et al.*, "Complementary methods for SARS-CoV-2 diagnosis in times of material shortage," *Sci. Rep.*, vol. 11, no. 1, p. 11899, 2021.

[4] M. Elsharkawy *et al.*, "Early assessment of lung function in coronavirus patients using invariant markers from chest X-rays images," *Sci. Rep.*, vol. 11, no. 1, p. 12095, 2021.

[5] B. E. Bejnordi *et al.*, "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer," *Jama*, vol. 318, no. 22, pp. 2199–2210, 2017.

[6] J. Amin, M. Sharif, M. Yasmin, and S. L. Fernandes, "Big data analysis for brain tumor detection: Deep convolutional neural networks," *Futur. Gener. Comput. Syst.*, vol. 87, pp. 290–297, 2018.

[7] L. A. Pastur-Romay, F. Cedrón, A. Pazos, and A. B. Porto-Pazos, "Deep artificial neural networks and neuromorphic chips for big data analysis: pharmaceutical and bioinformatics applications," *Int. J. Mol. Sci.*, vol. 17, no. 8, p. 1313, 2016.

[8] A. Bhandary *et al.*, "Deep-learning framework to detect lung abnormality–A study with chest X-Ray and lung CT scan images," *Pattern Recognit. Lett.*, vol. 129, pp. 271–278, 2020.

[9] M. Kuchana, A. Srivastava, R. Das, J. Mathew, A. Mishra, and K. Khatter, "AI aiding in diagnosing, tracking recovery of COVID-19 using deep learning on Chest CT scans," *Multimed. Tools Appl.*, vol. 80, pp. 9161–9175, 2021.

[10] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, 2017.

[11] K. El Asnaoui, Y. Chawki, and A. 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*, Springer International Publishing, 2021, pp. 257–284.

[12] Z. Yao *et al.*, “A machine learning-based pulmonary venous obstruction prediction model using clinical data and CT image,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 16, pp. 609–617, 2021.

[13] Z. Obermeyer and E. J. Emanuel, “Predicting the future - big data, machine learning, and clinical medicine,” *N. Engl. J. Med.*, vol. 375, pp. 1216–1219, 2016.

[14] J. Too, G. Liang, and H. Chen, “Memory-based Harris hawk optimization with learning agents: a feature selection approach,” *Eng. Comput.*, vol. 38, no. Suppl 5, pp. 4457–4478, 2022.

[15] S. Zhao, P. Wang, A. A. Heidari, X. Zhao, C. Ma, and H. Chen, “An enhanced Cauchy mutation grasshopper optimization with trigonometric substitution: engineering design and feature selection,” *Eng. Comput.*, vol. 38, no. Suppl 5, pp. 4583–4616, 2022.

[16] X. Zhang *et al.*, “Gaussian mutational chaotic fruit fly-built optimization and feature selection,” *Expert Syst. Appl.*, vol. 141, p. 112976, 2020.

[17] R. S. M. Ghosh R. Guha and A. Abraham, “A wrapper-filter feature selection technique based on ant colony optimization,” *Neural Comput. Applic.*, vol. 32, pp. 7839–7857, 2020.

[18] Y. Zheng *et al.*, “A novel hybrid algorithm for feature selection based on whale optimization algorithm,” *IEEE Access*, vol. 7, pp. 14908–14923, 2018.

[19] A. G. Hussien, A. E. Hassanien, E. H. Houssein, S. Bhattacharyya, and M. Amin, “S-shaped binary whale optimization algorithm for feature selection,” in *Recent trends in signal and image processing: ISSIP 2017*, Springer Singapore, 2019, pp. 79–87.

[20] M. Taradeh *et al.*, “An evolutionary gravitational search-based feature selection,” *Inf. Sci. (Ny).*, vol. 497, pp. 219–239, 2019.

[21] A. Ghosh, A. Datta, and S. Ghosh, “Self-adaptive differential evolution for feature selection in hyperspectral image data,” *Appl. Soft Comput.*, vol. 13, no. 4, pp. 1969–1977, 2013.

[22] M. Zhang Y. Chen and W. Susilo, “Ppo-cpq: a privacy-preserving optimization of clinical pathway query for e-healthcare systems,” *IEEE Internet Things J.*, vol. 7, pp. 10660–10672, 2020.

[23] A. A. A. Mohamed, S. A. Hassan, A. M. Hemeida, S. Alkhalaf, M. M. M. Mahmoud, and A. M. B. Eldin, “Parasitism–Predation algorithm (PPA): A novel approach for feature selection,” *Ain Shams Eng. J.*, vol. 11, no. 2, pp. 293–308, 2020.

[24] X. L. J. Z. W. C. B. Cao M. Li and Z. Lv, “Many-objective deployment optimization for a drone-assisted camera network,” *IEEE Trans. Netw. Sci. Eng.*, vol. 8, pp. 2756–2764, 2021.

[25] Y. Djenouri, D. Djenouri, A. Belhadi, P. Fournier-Viger, J. C. W. Lin, and A. Bendjoudi, “Exploiting GPU parallelism in improving bees swarm optimization for mining big transactional databases,” *Inf. Sci. (Ny).*, vol. 496, pp. 326–342, 2019.

[26] Q. B. Zhang, P. Wang, and Z. H. Chen, “An improved particle filter for mobile robot localization based on particle swarm optimization,” *Expert Syst. Appl.*, vol. 135, pp. 181–193, 2019.

[27] M. Chao, C. Kai, and Z. Zhiwei, “Research on tobacco foreign body detection device based on machine vision,” *Trans. Inst. Meas. Control*, vol. 42, no. 15, pp. 2857–2871, 2020.

[28] Y. Sun and Y. Chen, “Multi-population improved whale optimization algorithm for high dimensional optimization,” *Appl. Soft Comput.*, vol. 112, 2021.

[29] H. Chen, C. Yang, A. A. Heidari, and X. Zhao, “An efficient double adaptive random spare reinforced whale optimization algorithm,” *Expert Syst. Appl.*, vol. 154, p. 113018, 2020.

[30] B. Cao, Y. Gu, Z. Lv, S. Yang, J. Zhao, and Y. Li, “RFID reader anticollision based on distributed parallel particle swarm optimization,” *IEEE Internet Things J.*, vol. 8,

no. 5, pp. 3099–3107, 2020.

[31] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, “Slime mould algorithm: A new method for stochastic optimization,” *Futur. Gener. Comput. Syst.*, vol. 111, pp. 300–323, 2020.

[32] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, “Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts,” *Expert Syst. Appl.*, vol. 177, p. 114864, 2021.

[33] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, “Harris hawks optimization: Algorithm and applications,” *Futur. Gener. Comput. Syst.*, vol. 97, pp. 849–872, 2019.

[34] I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, and H. Chen, “RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method,” *Expert Syst. Appl.*, vol. 181, p. 115079, 2021.

[35] G. G. Wang, D. Gao, and W. Pedrycz, “Solving multiobjective fuzzy job-shop scheduling problem by a hybrid adaptive differential evolution algorithm,” *IEEE Trans. Ind. Informatics*, vol. 18, no. 12, pp. 8519–8528, 2022.

[36] R. Najjar, “Redefining radiology: a review of artificial intelligence integration in medical imaging,” *Diagnostics*, vol. 13, no. 17, p. 2760, 2023.

[37] J. Ker, L. Wang, J. Rao, and T. Lim, “Deep learning applications in medical image analysis,” *IEEE Access*, vol. 6, pp. 9375–9389, 2017.

[38] M. M. A. Monshi, J. Poon, and V. Chung, “Deep learning in generating radiology reports: A survey,” *Artif. Intell. Med.*, vol. 106, p. 101878, 2020.

[39] H. C. Shin, K. Roberts, L. Lu, D. Demner-Fushman, J. Yao, and R. M. Summers, “Learning to read chest x-rays: Recurrent neural cascade model for automated image annotation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2497–2506.

[40] Y. Li, X. Liang, Z. Hu, and E. P. Xing, “Hybrid retrieval-generation reinforced agent

for medical image report generation," in *Advances in neural information processing systems*, 2018.

[41] P. L. Schuyler, W. T. Hole, M. S. Tuttle, and D. D. Sherertz, "The UMLS Metathesaurus: representing different views of biomedical concepts," *Bull. Med. Libr. Assoc.*, vol. 81, no. 2, p. 217, 1993.

[42] C. P. Langlotz, "RadLex: a new method for indexing online educational materials," *Radiographics*, vol. 26, no. 6, pp. 1595–1597, 2006.

[43] D. Demner-Fushman *et al.*, "Preparing a collection of radiology examinations for distribution and retrieval," *J. Am. Med. Informatics Assoc.*, vol. 23, no. 2, pp. 304–310, 2016.

[44] B. Jing, P. Xie, and E. Xing, "On the automatic generation of medical imaging reports," *arXiv Prepr. arXiv1711.08195*, 2017.

[45] Y. Xue *et al.*, "Multimodal recurrent model with attention for automated radiology report generation," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I*, Springer International Publishing, 2018, pp. 457–466.

[46] X. Wang, Y. Peng, L. Lu, Z. Lu, and R. M. Summers, "Tienet: Text-image embedding network for common thorax disease classification and reporting in chest x-rays," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9049–9058.

[47] R. M. Thanki and A. Kothari, *Hybrid and advanced compression techniques for medical images*. Springer International Publishing, 2019.

[48] J. Irvin *et al.*, "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison," in *Proceedings of the AAAI conference on artificial intelligence*, 2019, pp. 590–597.

[49] N. H. S. England, "Statistics ≫ Diagnostic Imaging Dataset." 2024.

[50] H. Y. F. Wong *et al.*, “Frequency and distribution of chest radiographic findings in patients positive for COVID-19,” *Radiology*, vol. 296, no. 2, pp. E72--E78, 2020.

[51] B. K. Sahu and R. Verma, “DICOM search in medical image archive solution e-Sushrut Chhavi,” in *2011 3rd International Conference on Electronics Computer Technology*, 2011, pp. 256–260.

[52] O. Six and B. Quantib, *The ultimate guide to AI in radiology*. Artificial Intelligence in Healthcare Solutions, 2019.

[53] S. M. Lee *et al.*, “Deep learning applications in chest radiography and computed tomography: current state of the art,” *J. Thorac. Imaging*, vol. 34, no. 2, pp. 75–85, 2019.

[54] H. Mohsen, E. S. A. El-Dahshan, E. S. M. El-Horbaty, and A. B. M. Salem, “Classification using deep learning neural networks for brain tumors,” *Futur. Comput. Informatics J.*, vol. 3, no. 1, pp. 68–71, 2018.

[55] A. S. Chaudhari *et al.*, “Super-resolution musculoskeletal MRI using deep learning,” *Magn. Reson. Med.*, vol. 80, no. 5, pp. 2139–2154, 2018.

[56] E. Gibson *et al.*, “NiftyNet: a deep-learning platform for medical imaging,” *Comput. Methods Programs Biomed.*, vol. 158, pp. 113–122, 2018.

[57] M. Biswas *et al.*, “State-of-the-art review on deep learning in medical imaging,” *Front. Biosci.*, vol. 24, no. 3, pp. 380–406, 2019.

[58] J. Zhong, J. Tang, C. Ye, and L. Dong, “The immunology of COVID-19: is immune modulation an option for treatment?,” *Lancet Rheumatol.*, vol. 2, no. 7, pp. e428--e436, 2020.

[59] W. H. Organization, “WHO Coronavirus (COVID-19) Dashboard.” 2024.

[60] S. N. Jain, T. Modi, Y. Aswani, and R. U. Varma, “Chest radiography in adult critical care unit: A pictorial review,” *Indian J. Radiol. Imaging*, vol. 29, no. 04, pp. 418–425, 2019.

[61] E. Eisenhuber, C. M. Schaefer-Prokop, H. Prosch, and W. Schima, “Bedside chest

radiography," *Respir. Care*, vol. 57, no. 3, pp. 427–443, 2012.

[62] S. Rakshit, I. Saha, M. Wlasnowolski, U. Maulik, and D. Plewczynski, "Deep learning for detection and localization of thoracic diseases using chest x-ray imagery," in *Artificial Intelligence and Soft Computing: 18th International Conference, ICAISC 2019, Zakopane, Poland, June 16–20, 2019, Proceedings, Part II*, Springer International Publishing, 2019, pp. 271–282.

[63] H. Behzadi-Khormouji *et al.*, "Deep learning, reusable and problem-based architectures for detection of consolidation on chest X-ray images," *Comput. Methods Programs Biomed.*, vol. 185, p. 105162, 2020.

[64] T. Singhal, "A review of coronavirus disease-2019 (COVID-19)," *Indian J. Pediatr.*, vol. 87, no. 4, pp. 281–286, 2020.

[65] X. Wang *et al.*, "Pulmonary edema severity estimation in chest radiographs using deep learning," 2019.

[66] A. H. Association, "Heart Failure." 2024.

[67] J. B. Lamare, T. Olatunji, and L. Yao, "On the diminishing return of labeling clinical reports," *arXiv Prepr. arXiv2010.14587*, 2020.

[68] F. Gaillard, "Radiopaedia.org, the wiki-based collaborative Radiology resource." 2014.

[69] D. H. Livingston, B. Shogan, P. John, and R. F. Lavery, "CT diagnosis of rib fractures and the prediction of acute respiratory failure," *J. Trauma Acute Care Surg.*, vol. 64, no. 4, pp. 905–911, 2008.

[70] S. Xu, J. Guo, G. Zhang, and R. Bie, "Automated detection of multiple lesions on chest X-ray images: Classification using a neural network technique with association-specific contexts," *Appl. Sci.*, vol. 10, no. 5, p. 1742, 2020.

[71] M. Soellner and J. Koenigstorfer, "Compliance with medical recommendations depending on the use of artificial intelligence as a diagnostic method," *BMC Med. Inform. Decis. Mak.*, vol. 21, pp. 1–11, 2021.

[72] I. R. Mendo, G. Marques, I. de la Torre Díez, M. López-Coronado, and F. Martín-Rodríguez, "Machine learning in medical emergencies: a systematic review and analysis," *J. Med. Syst.*, vol. 45, no. 10, p. 88, 2021.

[73] I. S. Masad, A. Alqudah, A. M. Alqudah, and S. Almashaqbeh, "A hybrid deep learning approach towards building an intelligent system for pneumonia detection in chest x-ray images," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 6, pp. 5530–5540, 2021, doi: 10.11591/ijece.v11i6.pp5530-5540.

[74] F. Grillet, J. Behr, P. Calame, S. Aubry, and E. Delabrousse, "Acute Pulmonary Embolism Associated with COVID-19 Pneumonia Detected with Pulmonary CT Angiography," *Radiology*, vol. 296, no. 3, pp. E186–E188, 2020, doi: 10.1148/radiol.2020201544.

[75] R. Bhardwaj, A. R. Nambiar, and D. Dutta, "A study of machine learning in healthcare," in *2017 IEEE 41st annual computer software and applications conference (COMPSAC)*, 2017, pp. 236–241.

[76] L. Igual and S. Seguí, "Supervised learning," in *Introduction to Data Science: A Python Approach to Concepts, Techniques and Applications*, Springer, 2024, pp. 67–97.

[77] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, "Unsupervised learning," in *An introduction to statistical learning: with applications in Python*, Springer, 2023, pp. 503–556.

[78] D. Ernst and A. Louette, "Introduction to reinforcement learning," *Feuerriegel, S., Hartmann, J., Janiesch, C., Zschech, P.(2024). Gener. ai. Bus. Inf. Syst. Eng.*, vol. 66, no. 1, pp. 111–126, 2024.

[79] O. M. El Zein, M. M. Soliman, A. K. Elkholy, and N. I. Ghali, "Transfer learning based model for pneumonia detection in chest X-ray images," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 5, pp. 56–66, 2021.

[80] H. M. Ahmed and M. Y. Kashmola, "A proposed architecture for convolutional neural networks to detect skin cancers," *IAES Int. J. Artif. Intell.*, vol. 11, no. 2, pp. 485–493,

2022, doi: 10.11591/ijai.v11.i2.pp485-493.

[81] B. Ramsundar, P. Eastman, P. Walters, and V. Pande, *Deep learning for the life sciences: applying deep learning to genomics, microscopy, drug discovery, and more*. " O'Reilly Media, Inc.," 2019.

[82] S. Innani, P. Dutande, B. Baheti, U. Baid, and S. Talbar, "Deep learning based novel cascaded approach for skin lesion analysis," in *International Conference on Computer Vision and Image Processing*, Springer, 2022, pp. 615–626.

[83] Y. Hedhoud, T. Mekhaznia, and M. Amroune, "An improvement of the CNN-XGboost model for pneumonia disease classification," *Polish J. Radiol.*, vol. 88, no. 1, pp. e483–e493, 2023, doi: 10.5114/pjr.2023.132533.

[84] M. Au-Yong-Oliveira, A. Pesqueira, M. J. Sousa, F. Dal Mas, and M. Soliman, "The potential of big data research in healthcare for medical doctors' learning," *J. Med. Syst.*, vol. 45, no. 1, p. 13, 2021.

[85] Z. Zhang, Y. Genc, D. Wang, M. E. Ahsen, and X. Fan, "Effect of ai explanations on human perceptions of patient-facing ai-powered healthcare systems," *J. Med. Syst.*, vol. 45, no. 6, p. 64, 2021.

[86] C. Krittanawong, "The rise of artificial intelligence and the uncertain future for physicians," *Eur. J. Intern. Med.*, vol. 48, pp. e13--e14, 2018.

[87] A. Murugan, S. A. H. Nair, and K. S. Kumar, "Detection of skin cancer using SVM, random forest and kNN classifiers," *J. Med. Syst.*, vol. 43, no. 8, p. 269, 2019.

[88] D. Palani and K. Venkatalakshmi, "An IoT based predictive modelling for predicting lung cancer using fuzzy cluster based segmentation and classification," *J. Med. Syst.*, vol. 43, no. 2, p. 21, 2019.

[89] H. Zerouaoui and A. Idri, "Reviewing machine learning and image processing based decision-making systems for breast cancer imaging," *J. Med. Syst.*, vol. 45, no. 1, p. 8, 2021.

[90] K. Jafarian, V. Vahdat, S. Salehi, and M. Mobin, "Automating detection and

localization of myocardial infarction using shallow and end-to-end deep neural networks," *Appl. Soft Comput.*, vol. 93, p. 106383, 2020.

[91] Ö. Yıldırım, P. Pławiak, R. S. Tan, and U. R. Acharya, "Arrhythmia detection using deep convolutional neural network with long duration ECG signals," *Comput. Biol. Med.*, vol. 102, pp. 411–420, 2018.

[92] D. S. W. Ting *et al.*, "Artificial intelligence and deep learning in ophthalmology," *Br. J. Ophthalmol.*, vol. 103, no. 2, pp. 167–175, 2019.

[93] S. Govindarajan and R. Swaminathan, "Analysis of tuberculosis in chest radiographs for computerized diagnosis using bag of keypoint features," *J. Med. Syst.*, vol. 43, no. 4, p. 87, 2019.

[94] M. Khan *et al.*, "Applications of artificial intelligence in COVID-19 pandemic: A comprehensive review," *Expert Syst. Appl.*, vol. 185, p. 115695, 2021.

[95] H. Liz, M. Sánchez-Montañés, A. Tagarro, S. Domínguez-Rodríguez, R. Dagan, and D. Camacho, "Ensembles of convolutional neural network models for pediatric pneumonia diagnosis," *Futur. Gener. Comput. Syst.*, vol. 122, pp. 220–233, 2021.

[96] N. Barakat, M. Awad, and B. A. Abu-Nabah, "A machine learning approach on chest X-rays for pediatric pneumonia detection," *Digit. Heal.*, vol. 9, p. 20552076231180010, 2023.

[97] M. Goyal, R. Goyal, and B. Lall, "Learning activation functions: A new paradigm for understanding neural networks," *arXiv Prepr. arXiv1906.09529*, 2019.

[98] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2009.

[99] CV-Tricks, "ResNet, AlexNet, VGGNet, Inception: Understanding various architectures of Convolutional Networks." 2024.

[100] A. N. Khalifa, H. R. Ali, S. A. Jebur, and S. A. Jahefer, "Automate facial paralysis detection using vgg architectures," *Int. J. Curr. Innov. Adv. Res.*, pp. 1–8, 2024.

[101] M. Neshat, M. Ahmed, H. Askari, M. Thilakaratne, and S. Mirjalili, "Hybrid

Inception Architecture with Residual Connection: Fine-tuned Inception-ResNet Deep Learning Model for Lung Inflammation Diagnosis from Chest Radiographs," *Procedia Comput. Sci.*, vol. 235, pp. 1841–1850, 2024.

[102] S. Li, H. Qu, X. Dong, B. Dang, H. Zang, and Y. Gong, "Leveraging deep learning and xception architecture for high-accuracy mri classification in alzheimer diagnosis," *arXiv Prepr. arXiv2403.16212*, 2024.

[103] A. Keles, M. B. Keles, and A. Keles, "COV19-CNNet and COV19-ResNet: diagnostic inference Engines for early detection of COVID-19," *Cognit. Comput.*, vol. 16, no. 4, pp. 1612–1622, 2024.

[104] O. N. Mohammed, "Enhancing pulmonary disease classification in diseases: A comparative study of CNN and optimized MobileNet architectures," *J. Robot. Control*, vol. 5, no. 2, pp. 427–440, 2024.

[105] A. Alzahrani, "Digital Image Forensics: An Improved DenseNet Architecture for Forged Image Detection," *Eng. Technol. Appl. Sci. Res.*, vol. 14, no. 2, pp. 13671–13680, 2024.

[106] M. A. Thanoon, M. A. Zulkifley, M. A. A. Mohd Zainuri, and S. R. Abdani, "A Review of Deep Learning Techniques for Lung Cancer Screening and Diagnosis Based on CT Images," *Diagnostics*, vol. 13, no. 16, 2023, doi: 10.3390/diagnostics13162617.

[107] J. M. Czum, "Dive Into Deep Learning," *J. Am. Coll. Radiol.*, vol. 17, no. 5, pp. 637–638, 2020, doi: 10.1016/j.jacr.2020.02.005.

[108] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[109] M. Mhamed *et al.*, "A deep CNN architecture with novel pooling layer applied to two Sudanese Arabic sentiment data sets," *J. Inf. Sci.*, p. 01655515231188341, 2022.

[110] J. Jian *et al.*, "Multiple instance convolutional neural network with modality-based attention and contextual multi-instance learning pooling layer for effective differentiation between borderline and malignant epithelial ovarian tumors," *Artif. Intell. Med.*, vol. 121, p. 102194, 2021.

[111] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, “Dive into deep learning,” *arXiv Prepr. arXiv2106.11342*, 2021.

[112] L. Yin, X. Cao, and D. Liu, “Weighted fully-connected regression networks for one-day-ahead hourly photovoltaic power forecasting,” *Appl. Energy*, vol. 332, p. 120527, 2023.

[113] M. Latifi and H. Amindavar, “Forward-looking ground penetrating radar image reconstruction using fully connected neural network,” *Signal Processing*, vol. 203, p. 108788, 2023.

[114] A. G. Howard *et al.*, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv Prepr. arXiv1704.04861*, 2017.

[115] B. Han, M. Hu, X. Wang, and F. Ren, “A triple-structure network model based upon MobileNet V1 and multi-loss function for facial expression recognition,” *Symmetry (Basel).*, vol. 14, no. 10, p. 2055, 2022.

[116] K. Rajwar, K. Deep, and S. Das, “An exhaustive review of the metaheuristic algorithms for search and optimization: Taxonomy, applications, and open challenges,” *Artif. Intell. Rev.*, vol. 56, no. 11, pp. 13187–13257, 2023.

[117] F. S. Gharehchopogh, “Quantum-inspired metaheuristic algorithms: comprehensive survey and classification,” *Artif. Intell. Rev.*, vol. 56, no. 6, pp. 5479–5543, 2023.

[118] A. Bouaouda and Y. Sayouti, “Hybrid meta-heuristic algorithms for optimal sizing of hybrid renewable energy system: a review of the state-of-the-art,” *Arch. Comput. Methods Eng.*, vol. 29, no. 6, pp. 4049–4083, 2022.

[119] B. Morales-Castañeda, D. Zaldivar, E. Cuevas, F. Fausto, and A. Rodríguez, “A better balance in metaheuristic algorithms: Does it exist?,” *Swarm Evol. Comput.*, vol. 54, p. 100671, 2020.

[120] J. O. Agushaka and A. E. Ezugwu, “Initialisation approaches for population-based metaheuristic algorithms: a comprehensive review,” *Appl. Sci.*, vol. 12, no. 2, p. 896, 2022.

[121] S. Kaur, Y. Kumar, A. Koul, and S. Kumar Kamboj, "A systematic review on metaheuristic optimization techniques for feature selections in disease diagnosis: open issues and challenges," *Arch. Comput. Methods Eng.*, vol. 30, no. 3, pp. 1863–1895, 2023.

[122] M. Azizi, S. Talatahari, and A. H. Gandomi, "Fire Hawk Optimizer: A novel metaheuristic algorithm," *Artif. Intell. Rev.*, vol. 56, no. 1, pp. 287–363, 2023.

[123] B. Toaza and D. Esztergár-Kiss, "A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems," *Appl. Soft Comput.*, p. 110908, 2023.

[124] O. W. Khalid, N. A. M. Isa, and H. A. M. Sakim, "Emperor penguin optimizer: A comprehensive review based on state-of-the-art meta-heuristic algorithms," *Alexandria Eng. J.*, vol. 63, pp. 487–526, 2023.

[125] J. O. Agushaka, A. E. Ezugwu, and L. Abualigah, "Gazelle optimization algorithm: a novel nature-inspired metaheuristic optimizer," *Neural Comput. Appl.*, vol. 35, no. 5, pp. 4099–4131, 2023.

[126] E. H. Houssein, M. R. Saad, F. A. Hashim, H. Shaban, and M. Hassaballah, "Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 94, p. 103731, 2020.

[127] W. Zhao *et al.*, "Electric eel foraging optimization: A new bio-inspired optimizer for engineering applications," *Expert Syst. Appl.*, vol. 238, p. 122200, 2024.

[128] K. De Jong, "Learning with genetic algorithms: An overview," *Mach. Learn.*, vol. 3, pp. 121–138, 1988.

[129] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, pp. 341–359, 1997.

[130] D. Simon, "Biogeography-based optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, 2008.

[131] P. J. Van Laarhoven and E. H. Aarts, "Simulated annealing," in *Simulated annealing:*

Theory and applications, Springer Netherlands, 1987, pp. 7–15.

[132] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Inf. Sci. (Ny).*, vol. 179, no. 13, pp. 2232–2248, 2009.

[133] S. Kumar, D. Datta, and S. K. Singh, "Black hole algorithm and its applications," in *Computational intelligence applications in modeling and control*, Springer, 2015, pp. 147–170.

[134] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowledge-based Syst.*, vol. 191, p. 105190, 2020.

[135] F. A. Hashim, R. R. Mostafa, A. G. Hussien, S. Mirjalili, and K. M. Sallam, "Fick's Law Algorithm: A physical law-based algorithm for numerical optimization," *Knowledge-Based Syst.*, vol. 260, p. 110146, 2023.

[136] M. Y. Cheng and M. N. Sholeh, "Optical microscope algorithm: A new metaheuristic inspired by microscope magnification for solving engineering optimization problems," *Knowledge-Based Syst.*, vol. 279, p. 110939, 2023.

[137] J. Li, M. Noto, and Y. Zhang, "Improved Kepler Optimization Algorithm Based on Mixed Strategy," in *International Conference on Swarm Intelligence*, Springer, 2024, pp. 157–170.

[138] R. V Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems," *Comput. Des.*, vol. 43, no. 3, pp. 303–315, 2011.

[139] A. H. Kashan, "League championship algorithm: a new algorithm for numerical function optimization," in *2009 international conference of soft computing and pattern recognition*, 2009, pp. 43–48.

[140] Y. Shi, "Brain storm optimization algorithm," in *Advances in Swarm Intelligence: Second International Conference, ICSI 2011, Chongqing, China, June 12-15, 2011, Proceedings, Part I*, vol. 2, Springer Berlin Heidelberg, 2011, pp. 303–309.

[141] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems,"

Knowledge-based Syst., vol. 96, pp. 120–133, 2016.

[142] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, "Gradient-based optimizer: A new metaheuristic optimization algorithm," *Inf. Sci. (Ny).*, vol. 540, pp. 131–159, 2020.

[143] M. Abdel-Basset, D. El-Shahat, M. Jameel, and M. Abouhawwash, "Exponential distribution optimizer (EDO): A novel math-inspired algorithm for global optimization and engineering problems," *Artif. Intell. Rev.*, vol. 56, pp. 9329–9400, 2023.

[144] J. Xue and B. Shen, *Dung beetle optimizer: a new meta-heuristic algorithm for global optimization*, vol. 79, no. 7. 2023. doi: 10.1007/s11227-022-04959-6.

[145] Z. Xie, J. Wu, W. Tang, and Y. Liu, "Advancing image segmentation with DBO-Otsu: Addressing rubber tree diseases through enhanced threshold techniques," *PLoS One*, vol. 19, no. 3, p. e0297284, 2024.

[146] S. S. Verma, A. Prasad, and A. Kumar, "CovXmlc: High performance COVID-19 detection on X-ray images using Multi-Model classification," *Biomed. Signal Process. Control*, vol. 71, no. PB, p. 103272, 2022, doi: 10.1016/j.bspc.2021.103272.

[147] J. Sofia Jennifer and T. Sree Sharmila, "A Neutrosophic Set Approach on Chest X-rays for Automatic Lung Infection Detection," *Inf. Technol. Control*, vol. 52, no. 1, pp. 37–52, 2023, doi: 10.5755/j01.itc.52.1.31520.

[148] M. A. Alves *et al.*, "Explaining machine learning based diagnosis of COVID-19 from routine blood tests with decision trees and criteria graphs," *Comput. Biol. Med.*, vol. 132, p. 104335, 2021.

[149] K. K. Goswami, R. Kumar, R. Kumar, A. J. Reddy, and S. K. Goswami, "Deep learning classification of tuberculosis chest X-rays," *Cureus*, vol. 15, no. 7, 2023.

[150] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman, and P. R. Pinheiro, "Covidgan: data augmentation using auxiliary classifier gan for improved covid-19 detection," *Ieee Access*, vol. 8, pp. 91916–91923, 2020.

[151] M. Abdullah, F. berhe Abrha, B. Kedir, and T. T. Tagesse, "A Hybrid Deep Learning

CNN model for COVID-19 detection from chest X-rays," *Heliyon*, vol. 10, no. 5, 2024.

[152] X. Wu *et al.*, "Raman spectroscopy combined with machine learning algorithms for rapid detection Primary Sjögren's syndrome associated with interstitial lung disease," *Photodiagnosis Photodyn. Ther.*, vol. 40, p. 103057, 2022.

[153] R. K. M. M. M. S. K. S. Punitha T. Stephan and S. B. Belhaouari, "Detecting covid-19 from lung computed tomography images: A swarm optimized artificial neural network approach," *IEEE Access*, vol. 11, pp. 12378–12393, 2023.

[154] M. Tamal, M. Alshammari, M. Alabdullah, R. Hourani, H. A. Alola, and T. M. Hegazi, "An integrated framework with machine learning and radiomics for accurate and rapid early diagnosis of COVID-19 from Chest X-ray," *Expert Syst. Appl.*, vol. 180, p. 115152, 2021.

[155] A. M. Ismael and A. Şengür, "Deep learning approaches for COVID-19 detection based on chest X-ray images," *Expert Syst. Appl.*, vol. 164, p. 114054, 2021.