الجمهورية الجزَائِرية الديمقرَاطية الشعبية
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
وزَارة التعليم العَالي و البحث العلمي
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
جَامعة عمَار ثليجي بِالأَغوَاط
UNIVERSITY OF LAGHOUAT

كُلِّيَّة العُلُوم
FACULTY OF SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

Master's Thesis

Field : Mathematics and Computer Science
Department : Computer Sciences
Specialization : Artificial Intelligence and Data Science

Submitted by:

Baadj Rania **Guenou Souaad**

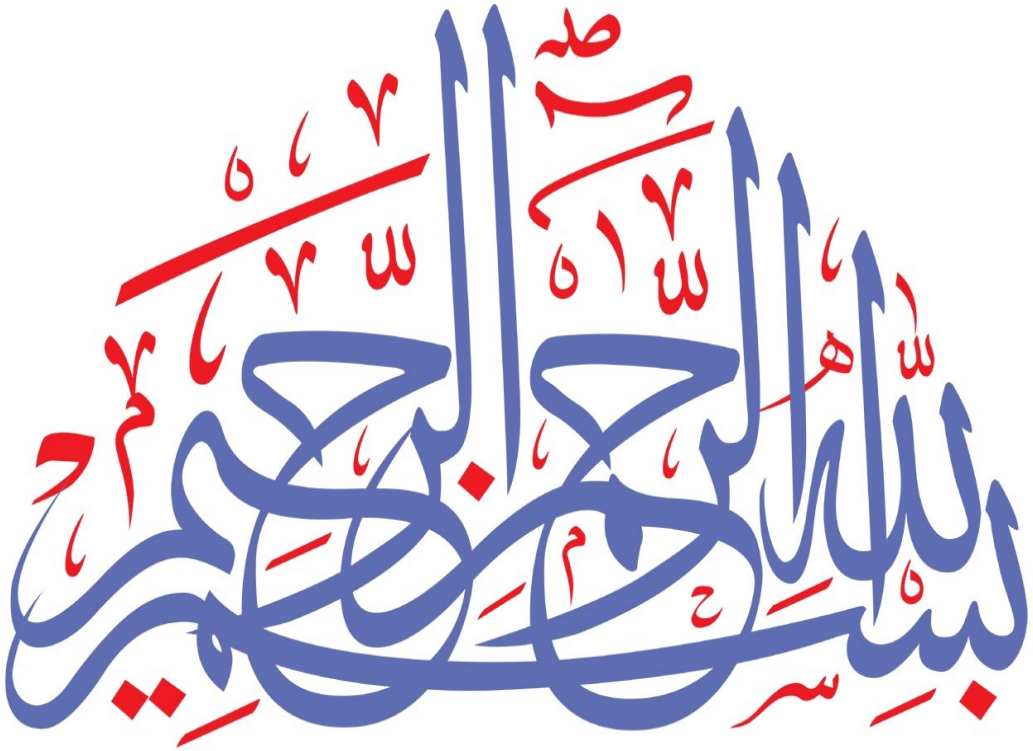
TOPIC

Optimizing Neural Networks for Healthcare Data: Ensemble Learning and Hyperparameter Optimization

Defended publicly on June 25th, 2025, before a jury composed of:

(University of Laghouat) President
(University of Laghouat) Examiner
(University of Laghouat) Supervisor

Thesis No. ..... – Academic Year 2024/2025


بسم الله الرحمن الرحيم

Acknowledgments

All our gratitude and thanks go to Allah Almighty, who granted us the strength, patience, and determination to carry out this work and see it through to completion.

First and foremost, we would like to express our sincere gratitude to our supervisor, Dr. GUELLOUMA Younes, for his continuous support, encouragement, and valuable guidance throughout this research journey. His trust and insightful advice have been essential to the progress and realization of this work. We are deeply grateful for his availability, kindness, and professionalism.

We would like to express our heartfelt gratitude to the respected members of the defense committee for their support and for taking the time to evaluate our work:
Dr. Hadda Cherroun, President of the Jury
Dr. Fatna Guibadj, Examiner 1
Dr. Saïda Sarra Boudouh, Examiner 2
Thank you for your time, your thoughtful attention to our thesis, and the valuable feedback you have shared. We truly appreciate your insights and suggestions, which will help us grow and further improve our research. It is an honor to have you as part of this important step in our academic journey.

We also wish to express our heartfelt appreciation to Dr. BOUDOUH Saida Sarra for her guidance and support at the early stages of this research, particularly in the field of artificial intelligence. Her insights helped lay a strong foundation for our study and inspired us to explore deeper into this evolving discipline.

Finally, we extend our gratitude to all the professors of the Computer Science Department, and to every teacher who has accompanied us throughout our academic journey. We also thank everyone who, directly or indirectly, contributed to the development and completion of this work.

Souaad, Rania, 2025

Dedications

I would like to dedicate this work to my parents, who have always supported and encouraged me to pursue my dreams. Your love and guidance have been invaluable throughout my academic journey. Thanks to them, I am here today.

To my dear sister, Fatima Zohra, and my brother, Belkacem thank you for always being there with advice and inspiration. You are my pillars of strength and motivation.

To all the colleges and to all my teachers.
I will always be grateful

RANIA
june 2025

Dedications

To my cherished parents, especially my father,
whose sacrifices shaped my strength and whose prayers
carried me through every step of this journey.

To my beloved siblings, with special affection for my sister Sarah,
your love, wisdom, and faith in me have been my silent anchors.

To my second family, my friends
your presence in this journey has meant more than words can say.
You'll know this is for you. Thank you for being the light in difficult days and
the joy in moments of doubt.

To the teachers whose guidance made this path clearer and gentler,
your support, patience, and dedication will always be remembered with gratitude.

With all my heart,

Souaad
June, 2025

ملخص

يتطلب تحسين أداء الشبكات العصبية على بيانات الرعاية الصحية التي تشمل الصور الطبية والسجلات السريرية والنتائج التشخيصية استراتيجيات تعزز الدقة، والقدرة على التعميم، والمتانة.

يستعرض هذا العمل دمج تقنيات التعلم التجميعي مثل التصويت الناعم، التصويت الصلب، والتجميع (Stacking) مع طرق هجينة لتحسين المعاملات، بما في ذلك الخوارزميات الجينية والبحث العشوائي. تم استخدام ثلاث شبكات عصبية تلافيفية مدرّبة ومعدّلة مسبقاً DenseNet201 وVGG16 وResNet101 كنماذج أساسية لمهام التصنيف الثنائي والمتعدد. وقد أظهرت طرق التجميع أداءً أفضل من النماذج الفردية من حيث الدقة، والاسترجاع، ودرجة F1.

على الرغم من هذه التحسينات، واجهنا عدة تحديات عملية، خصوصًا بسبب صغر حجم بعض مجموعات البيانات وعدم توازنها، مما أثّر سلبًا على استقرار النماذج ودقتها. كما أن القيود المادية مثل ضعف ذاكرة الـ GPU حدّت من عمق تحسين المعاملات.

بالإضافة إلى ذلك، استغرقت عملية التدريب وقتًا طويلاً، مما أبطأ من وتيرة التجارب. ومع ذلك، تُظهر نتائجنا أن استخدام تقنيات تجميع متنوعة مع أساليب تحسين ذكية يمكن أن يُحسّن بشكل كبير من أداء النماذج وموثوقيتها عند التعامل مع بيانات طبية معقدة.

الكلمات المفتاحية : بيانات الرعاية الصحية، الشبكات العصبية، التعلم التجميعي، تحسين المعاملات، التعلم العميق، الصور الطبية ، الخوارزمية الجينية، التجميع، التصويت.

Abstract

The integration of Artificial Intelligence (AI) in medical imaging has shown substantial potential, particularly in automating disease diagnosis using deep learning architectures. Chest X-ray classification has emerged as a critical application area, enabling early detection of thoracic diseases. However, standalone Convolutional Neural Networks (CNNs) are often limited in their generalization capabilities and sensitivity to suboptimal hyperparameters. This thesis investigates the performance impact of Hyperparameter Optimization (HPO) techniques specifically random search and genetic algorithms on three pretrained CNNs: DenseNet201, VGG16, and ResNet101. After optimization, Ensemble Learning (EL) strategies, including soft voting and stacking, were employed to enhance classification accuracy and model robustness. The proposed methodology, guided by findings in the existing literature, yielded notable improvements in multiclass diagnostic performance. Nevertheless, the research faced practical constraints including difficulties in acquiring suitable data and selecting the most appropriate dataset for the task, limited GPU memory, and extended training time, which influenced model stability and optimization depth.

Keywords: Healthcare Data,, Ensemble Learning (EL), Hyperparameter optimization (HPO), Deep Learning, Fine-Tuning, Genetic Algorithm, Stacking, Voting.

Résumé

L'intégration de l'intelligence artificielle (IA) dans l'imagerie médicale a démontré un potentiel considérable, notamment dans l'automatisation du diagnostic des maladies à l'aide d'architectures d'apprentissage profond. La classification des radiographies thoraciques constitue une application cruciale, favorisant la détection précoce des pathologies pulmonaires. Toutefois, les réseaux de neurones convolutifs (CNN) utilisés seuls présentent des limites en matière de généralisation et de sensibilité aux hyperparamètres sous-optimaux. Ce mémoire étudie l'impact de l'optimisation des hyperparamètres (HPO), en particulier la recherche aléatoire et les algorithmes génétiques, sur trois modèles CNN préentraînés : DenseNet201, VGG16 et ResNet101. Après optimisation, des techniques d'apprentissage ensembliste (EL), notamment le vote souple et le stacking, ont été appliquées pour améliorer la précision de classification et la robustesse des modèles. La méthodologie proposée, inspirée des travaux de l'état de l'art, a conduit à des améliorations notables des performances en diagnostic multiclasses. Néanmoins, plusieurs contraintes pratiques ont été rencontrées, notamment des difficultés dans la collecte et la sélection des données adéquates, des limites matérielles en mémoire GPU, et un temps d'entraînement prolongé, impactant la stabilité et l'optimisation des modèles.

Mots-clés : Données médicales, , EL, HPO, Apprentissage profond, Réglage fin, Algorithme génétique, Stacking, Vote.

Contents

List of Figures

List of Tables

List of Acronyms

AI Artificial Intelligence. 1, 4, 5, 16, 36

CNN Convolutional Neural Network. 1, 2, 4, 5, 6, 7, 10, 11, 12, 16, 18, 19, 20, 21, 23, 24, 26, 33, 34, 45, 47, 49, 50

DL Deep Learning. xii, 2, 5, 16, 17, 18, 20, 22, 26, 34, 35, 49, 50

EL Ensemble Learning. vi, vii, 1, 2, 4, 10, 16, 17, 18, 20, 21, 22, 28, 30, 33, 34, 35, 38, 40, 49, 50

FT Fine-Tuning. 4, 9, 16, 20

GA Genetic Algorithm. 38, 40, 41, 47, 48, 49

HPO Hyperparameter optimization. vi, vii, 1, 2, 4, 12, 16, 17, 23, 26, 27, 30, 31, 34, 38, 39, 49

ML Machine Learning. 5, 10, 13, 16, 18, 22, 23, 26, 28, 30, 33

TL Transfer Learning. 2, 4, 6, 7, 16, 20

1 Introduction

Contents

1.1 Context

In recent years, the application of Artificial Intelligence (AI) in the healthcare domain has grown substantially, particularly through the use of deep learning models for diagnostic support. Among these, chest X-ray analysis has emerged as a critical use case, offering valuable assistance in the early detection of pulmonary diseases and significantly easing the diagnostic workload of clinicians. Convolutional Neural Network (CNN) models have demonstrated remarkable effectiveness in processing medical images, yet their performance often varies across different patient populations and imaging conditions. This variability highlights the limitations of relying on individual models, which may lack robustness and generalization capacity. To mitigate these issues, EL techniques—such as soft voting and stacking—combine multiple predictive models to improve overall reliability and accuracy. Complementing EL, HPO systematically tunes critical parameters to ensure that each model performs optimally on the given dataset. The integration of EL and HPO thus offers a promising strategy for enhancing AI-driven diagnostic systems in healthcare.

1.2 Research Problem and Research Question

Although EL has shown considerable potential in improving performance in multiclass medical classification tasks, its effectiveness is often undermined by poorly tuned hyperparameters. Inadequate optimization can lead to overfitting, reduced accuracy, and inconsistent generalization across varied datasets. While ensemble methods have been widely explored, the specific contribution of structured HPO in enhancing their reliability remains under-investigated within the healthcare context. This gap motivates the central research question of this thesis:

Research Question *How does the use of hyperparameter optimization affect the performance and generalization of ensemble learning models in multiclass classification tasks within healthcare?*

1.3 Our Objectives

This thesis aims to investigate the effectiveness of combining EL and HPO to improve the performance of deep learning models for multiclass classification of chest X-ray images. Specifically, we evaluate three widely adopted pretrained CNN architectures—DenseNet201, VGG16, and ResNet101. Before training, hyperparameter tuning was conducted using two optimization strategies: random search and genetic algorithms, targeting key parameters such as learning rate, batch size, and the number of epochs. The optimized models were then trained and assessed individually using standard evaluation metrics, including accuracy, precision, recall, F1-score, and confusion matrices. To further enhance generalization and predictive stability, ensemble methods—namely soft voting and stacking—were employed to combine the outputs of the individual models. The study also discusses practical implementation challenges and outlines the methodological limitations encountered during experimentation.

1.4 Our thesis organization

This thesis is organized into five main chapters. The Introduction and Motivation chapter presents the research problem, objectives, and general structure. The Background chapter reviews key concepts such as medical imaging, DL, Transfer Learning (TL), EL, and HPO. The Related Work chapter, or The state-of-Art chapter explores previous research efforts involving ensemble strategies and optimization techniques in the healthcare domain. The Proposed Methodology chapter details the experimental setup, model training after the optimization processes, ensemble implementation, and performance evaluation. Finally, the Conclusion and Future Perspectives chapter summarizes the findings, discusses limitations, and outlines directions for future research.

2 Background

Contents

2.1 Introduction

This chapter provides the theoretical foundation necessary to understand and implement the methods used in this thesis. It begins with an overview of the healthcare domain and the critical role medical imaging plays in modern diagnostics, especially through X-ray analysis. It then introduces key deep learning concepts, focusing on TL and Fine-Tuning (FT) strategies that allow leveraging powerful pre-trained CNNs. Following this, the chapter explores EL, a method of combining multiple models to enhance prediction robustness especially relevant in healthcare. Specific ensemble techniques such as bagging, boosting, voting, and stacking are explained with emphasis on their application to CNNs. The final section presents HPO, highlighting its importance and outlining common HPO algorithms used to fine-tune model performance. Together, these sections lay the groundwork for the methodology and experiments that follow in later chapters.

2.2 Healthcare and Medical Images

Healthcare is increasingly becoming data driven, with medical imaging playing a crucial role in disease diagnosis, treatment planning, and patient monitoring. From X-rays and CT scans to MRIs and ultrasounds, these images provide vital insights that support clinical decision making.

2.2.1 Medical Images

Medical imaging refers to several different technologies that are used to view the human body in order to diagnose, monitor, or treat medical conditions [21].

It is also defined as the technique and process of imaging the interior of a body for clinical analysis and medical intervention, as well as visual representation of the function of some organs or tissues [22].

2.2.2 Xray Images

X-rays are a type of radiation called electromagnetic waves. X-ray imaging creates pictures of the inside of your body. The images show the parts of your body in different shades of black and white. This is because different tissues absorb different amounts of radiation. [23] In another word, an X-ray (also known in many languages as Röntgen radiation) is a form of high-energy electromagnetic radiation with a wavelength shorter than those of ultraviolet rays and longer than those of gamma rays. Roughly, X-rays have a wavelength ranging from 10 nanometers to 10 picometers, corresponding to frequencies in the range of 30 petahertz to 30 exahertz (3×1016 Hz to 3×1019 Hz) and photon energies in the range of 100 eV to 100 keV, respectively.

2.3 Artificial Intelligence and Machine Learning

AI is the science of making machines that can think like humans. It can do things that are considered "smart." AI technology can process large amounts of data in ways, unlike humans.

The goal for AI is to be able to do things such as recognize patterns, make decisions, and judgelike humans. A branch of AI known as Machine Learning (ML) is a technology of getting computers to learn and act as humans can do [24]. In general method is to train the ML algorithm using a collection of labelled or unlabelled training data to create a model, then introduce additional input data and have the system make predictions based on the model [25] .

2.4 Deep Learning

DL is a subset of ML that uses multilayered neural networks, called deep neural networks, to simulate the complex decision-making power of the human brain [26]. It is an aspect of data science that drives many applications and services that improve automation, performing analytical and physical tasks without human intervention. [26]

2.4.1 Convolutional Neural Networks (CNNs)

CNN, also known as ConvNet, is a specialized type of deep learning algorithm mainly designed for tasks that necessitate object recognition, including image classification, detection, and segmentation. CNNs are employed in a variety of practical scenarios, such as autonomous vehicles, security camera systems, and others [27]. The CNN presented with this mathematical expression:

$$h_j(x,y) = \sigma\left(\sum_{i=-m}^{m}\sum_{k=-n}^{n} I(x+i, y+k)\cdot K(i,k) + b_j\right)$$

Is based on convolution operation applied between the image and filter. Followed by a non-linear Activation function.
A typical CNN architecture (Figure 2.1) consists of several key components:

- **Convolutional layers:** Apply filters to input data to extract local features.
- **Pooling layers:** Reduce spatial dimensions, reducing computational cost and controlling overfitting.
- **Fully connected layers:** Perform high-level reasoning for classification or regression.
- **Output layer:** Produces final predictions, often using softmax for classification.

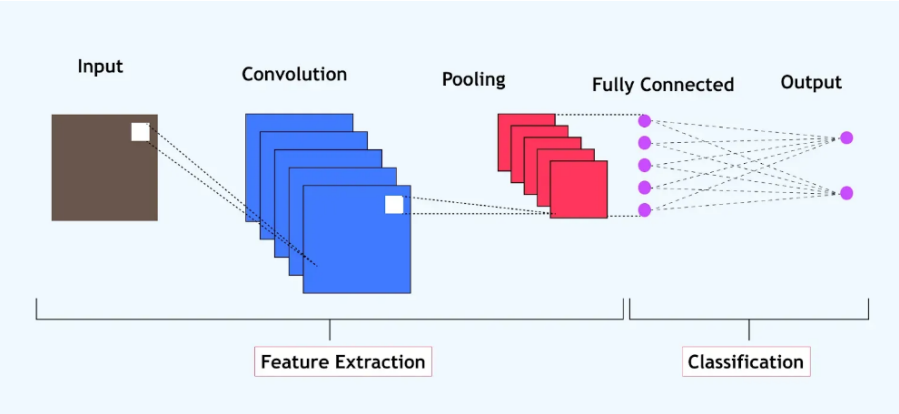


Figure 2.1: The Architecture of CNN [1]

Each layer transforms the input volume to an output volume of activations, which is then passed to the next layer. CNN have demonstrated exceptional performance in applications such as object detection, facial recognition, autonomous driving, and medical image analysis.

2.4.2 Transfer Learning

TL is considered a valuable tool for problems where the dataset is relatively small. The concept relies on leveraging a model that has been previously trained on a large-scale dataset—such as ImageNet—and adapting it to a different but related task. This approach has shown significant success in various domains, particularly in medicine [28] (Figure 2.7). It eliminates the need for extensive labeled data and reduces the computational burden and training time required to train a deep learning model from scratch.

Mathematically, let $f(x;\theta)$ be the pre-trained model with parameters θ learned from a source domain D_{source}. The goal of TL is to optimize the model on the target domain D_{target}:

$$\theta^* = \arg\min_{\theta} \mathcal{L}_{target}(f(x;\theta), y)$$

where $\mathcal{L}_{target}$ represents the loss function (e.g., binary cross-entropy) on the medical dataset, such as chest X-rays.

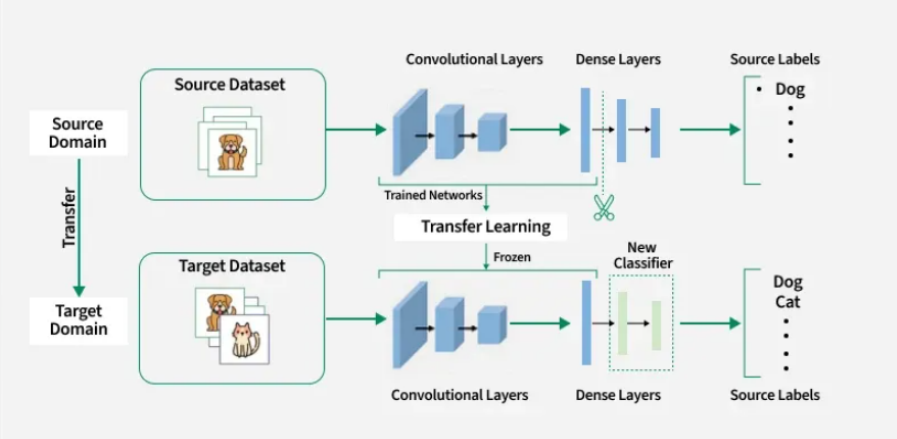


Figure 2.2: Transfer Learning [2]

Pre-trained CNN Architectures

Pre-trained model CNN architectures have become essential tools in computer vision, enabling efficient feature extraction through TL. By leveraging models trained on large datasets like ImageNet, tasks with limited data can benifit from deep representations without training from scratch. This section outlines three architectures: DenseNet,VGG and ResNet highlighting their core principale work.

DenseNet (Densely Connected CNN)

DenseNet DenseNet201 [29] is a type of CNN that utilises dense connections between layers [30], DenseNet introduces a paradigm shift by connecting each layer to every other layer in a feed-forward manner. Unlike traditional CNNs, which have a single connection between consecutive layers, DenseNet ensures that each layer receives inputs from all preceding layers and passes its output to all subsequent layers. This results in a network with L(L+1)/2 direct connections for L layers, significantly enhancing information flow [31] (Figure 2.3).

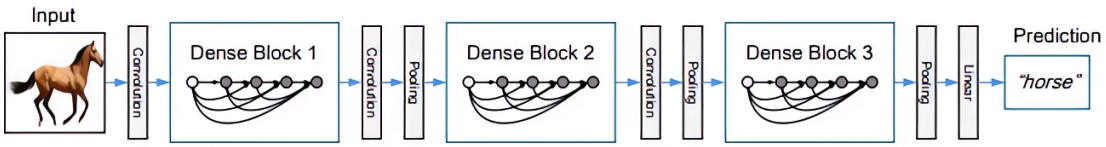


Figure 2.3: Architecture of DenseNet. [3]

When Dense blocks are the building blocks of DenseNet architectures. Each dense block consists of multiple convolutional layers [31].
The DenseNet architecture has several variants, including DenseNet-121, DenseNet-169, DenseNet-201, DenseNet-264. The number in each variant corresponds to the number of layers in the Network.

VGGNet (Visual Geometry Group)

VGGNet [32] is a typical deep CNN design with numerous layers [33], The VGGNet architecture is proposed by visual geometry group (VGG) team for ILSVRC 2014 and win this challenge [4]. The VGG Net consists of 16 or 19 layers, depending on the variant. The 16 layers is called VGG16, while the 19-layer version is called VGG-19.

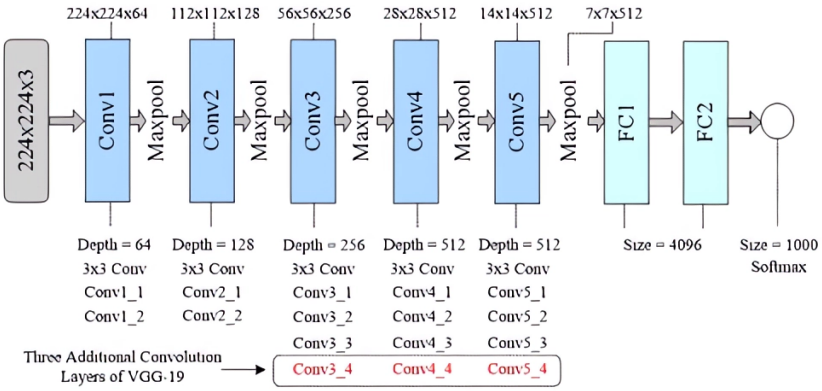


Figure 2.4: Architecture of VGG-16 and VGG-19. [4]

As shown in figure 2.4, VGG16 composed of 13 convolutional layers that employ 3x3 filters, each followed by a ReLU activation function, These Convolution layer are alternated with max-pooling layer using a filters 2x2 with a stride of 2. The network ends with two fully connected layers, each with 4096 neurons, and the final output layer with 1000 neurons. The softmax activation function is used to produce class probabilities.
The VGG19 have the same architecture with vGG16, with the exception that contient 3 extras convolutional layers.

ResNet (Residual Networks)
The residual neural network ResNet101 [34] is one of the most widely used networks recently. Based on a deep residual learning framework, ResNet can address the degradation problem and extract more information from the original data. [35] (Figure 2.5)

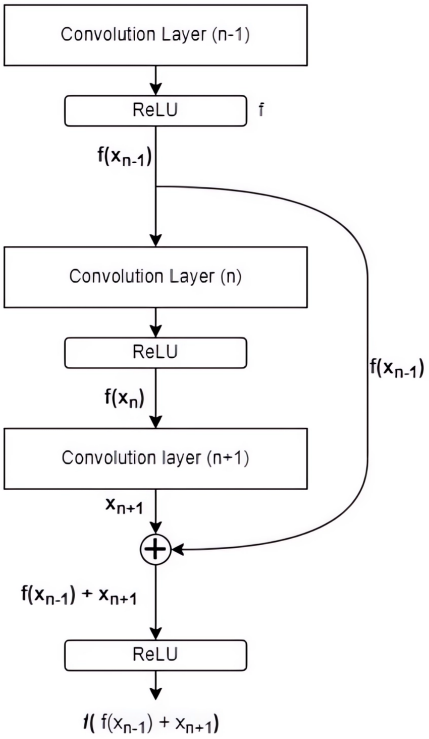


Figure 2.5: Structure of a residual block in ResNet. [5]

2.4.3 Fine Tuning

FT allows a pre-trained model to adapt to a new task. This approach (Figure 2.6) uses the knowledge gained from training a model on a large dataset and applying it to a smaller, domain specific dataset. FT involves adjusting the weights of the model's layers or updating certain parts of the model to improve its performance on the new task.

Formally, if $f(x) = g(h(x;\theta_1);\theta_2)$ where $h(x;\theta_1)$ are frozen base layers and $g(\cdot;\theta_2)$ is the new trainable classification head, FT may update both θ_2 and part of θ_1

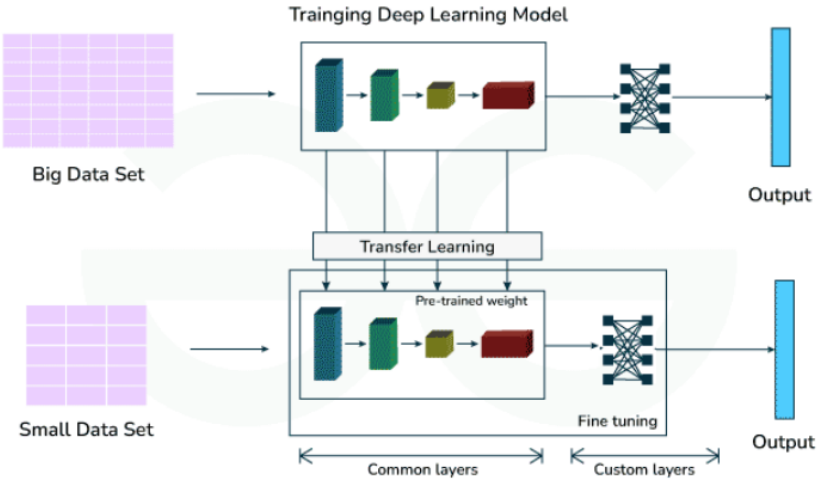


Figure 2.6: Fine Tuning. [6]

2.5 Ensemble Learning

EL is a ML strategy that enhances predictive performance by combining the outputs of multiple models. This approach aims to reduce generalization error and improve model robustness [36]. Various EL techniques exist, distinguished by how they aggregate predictions, including stacking, boosting, bagging, and voting [37].

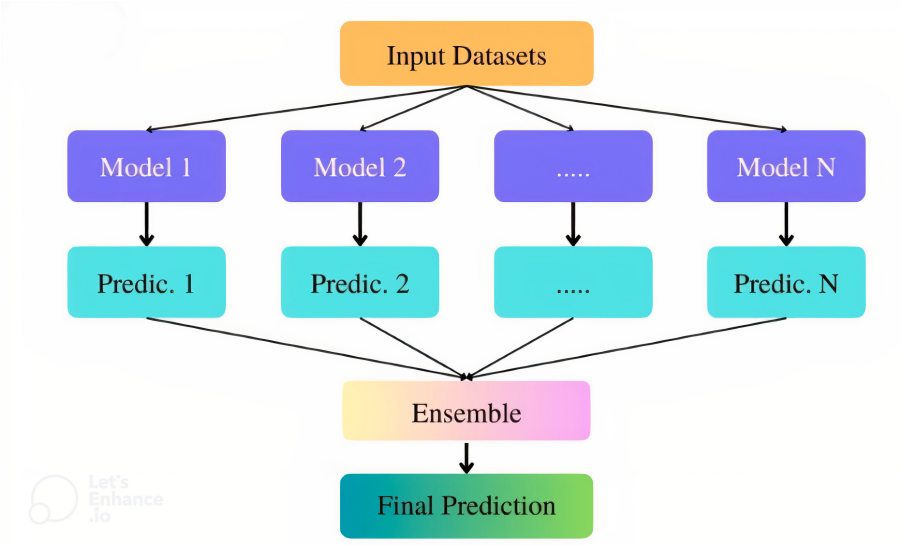


Figure 2.7: Ensemble Learning. [7]

Ensemble Learning Techniques

EL Techniques aim to improve model performance by combining the outputs of multiple learners, there are four types :

- **Bagging CNNs [38]**: This technique (Figure 2.8) trains multiple CNNs on different bootstrapped subsets of the dataset. The final prediction is made using majority voting (for classification) or averaging probabilities. Bagging helps reduce overfitting and increases model stability. *Use case: Classifying chest X-rays for pneumonia.*

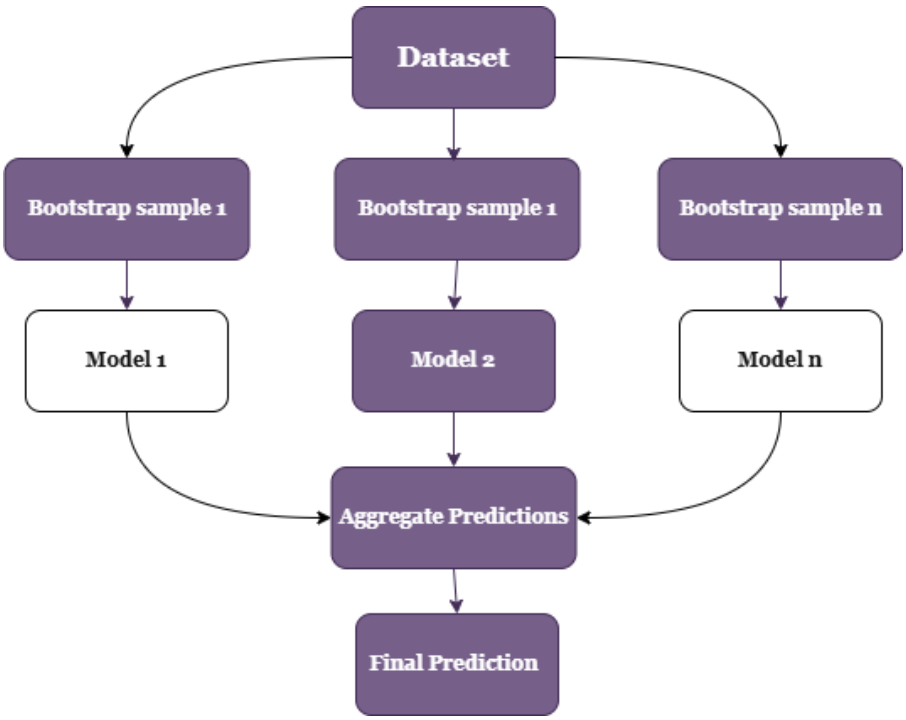


Figure 2.8: Bagging technique workflow.

- **Boosting CNNs [39]** : Boosting (Figure 2.9)involves training models sequentially, where each new model focuses on the errors made by its predecessors. Though less common with deep CNNs due to computational cost, it can be applied in hybrid setups like CNN + AdaBoost. The final prediction is a weighted combination of all models, which helps reduce bias and improve accuracy.

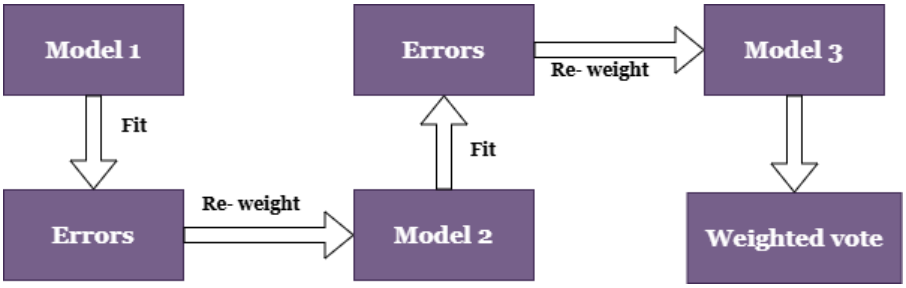


Figure 2.9: Boosting technique workflow.

- **Voting [40]**: This technique (Figure 2.10) aggregates predictions from multiple models to improve performance.
  - **Hard Voting**: Each model votes for a class label. The class with the most votes is the final prediction. For instance, if models predict (A, A, B), the result is A.
  - **Soft Voting**: Models output probability scores for each class. These are averaged and the class with the highest average probability is selected. Soft voting is effective when combining diverse models and is valuable in healthcare for integrating different data types.

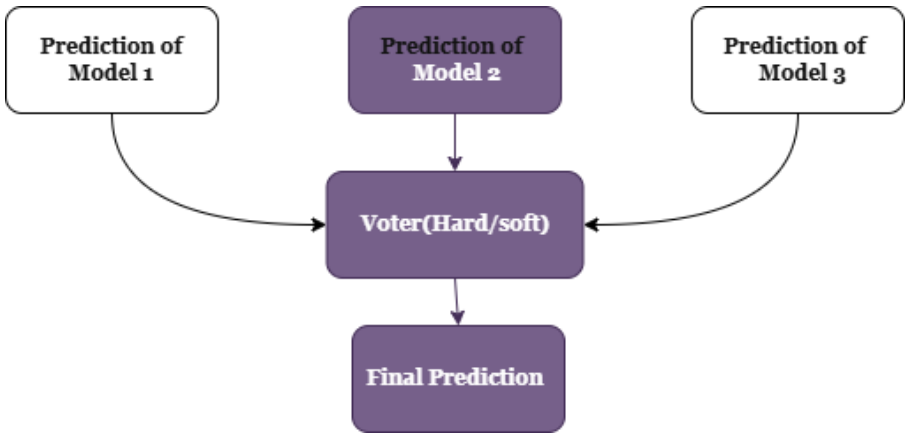


Figure 2.10: Voting technique workflow.

- **Stacking CNNs [41]**: Stacking (see Figure 2.11) uses the outputs of multiple CNNs as inputs to a meta-learner, such as logistic regression or a shallow neural network. This technique learns the best way to combine the predictions of base models and often leads to better performance in complex tasks like medical diagnosis.

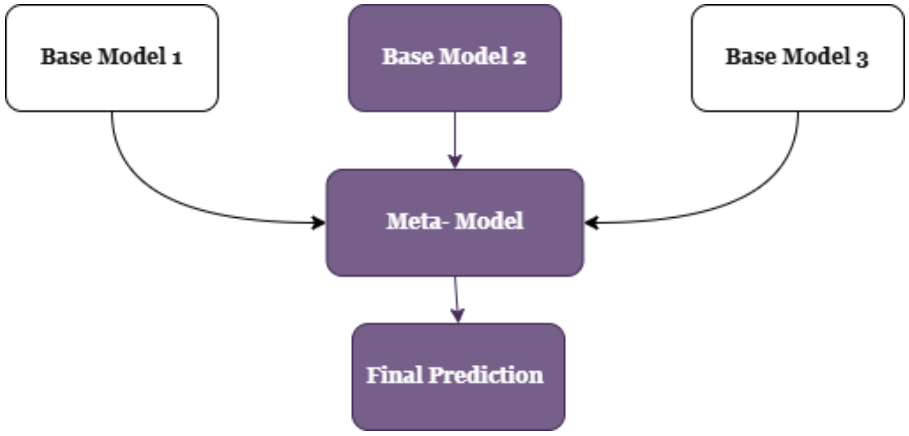


Figure 2.11: Stacking technique workflow.

2.6 Hyperparameter Optimization

Hyperparameters like learning rate, batch size, dropout rate, and number of layers greatly affect model performance. Rather than guessing, optimization methods such as Grid Search, Random Search, and Bayesian Optimization systematically explore parameter combinations to find the best configuration. In healthcare applications, fine-tuned hyperparameters help ensure models are not only accurate but also stable and generalizable across varying patient data.

2.6.1 Hyperparameter optimization algorithms

Several HPO algorithms have been introduced in different taxonomies over the years to optimize the network architecture of CNN. In this study, we briefly mention three different most popular algorithms, and chose to use two of them.

2.6.1.1 Grid Search

The traditional method of hyperparameters optimization is a Grid Search , which simply makes a complete search over a given subset of the hyperparameters space of the training algorithm 1. Because the ML algorithm parameter space may include spaces with real or unlimited values for some parameters, it is possible that we need to specify a boundary to apply a grid search. Grid search suffers from high dimensional spaces, but often can easily be parallelized, since the hyperparameter values that the algorithm works with are usually independent of each other [42].

Algorithm 1: Grid Search Algorithm

Function `grid_search`(*param_grid, model,* X_{train}, y_{train}, X_{val}, y_{val}, *metric*)**:**
- *best_score* $\leftarrow -\infty$
- *best_params* $\leftarrow$ `None`
- *combinations* $\leftarrow$ all combinations of *param_grid*
- **foreach** *params in combinations* **do**
  - *new_model* $\leftarrow$ initialize model with *params*
  - $new\_model.fit(X_{train}, y_{train})$
  - $predictions \leftarrow new\_model.predict(X_{val})$
  - $score \leftarrow metric(y_{val}, predictions)$
  - **if** $score > best\_score$ **then**
    - $best\_score \leftarrow score$
    - $best\_params \leftarrow params$
- **return** *best_params*, *best_score*

2.6.1.2 Random Search

Random Search overrides the complete selection of all combinations by their random selection. Its algorithm shown in 2 can be easily applied to discrete cases, but the method can be generalized to continuous and mixed spaces. Random search can outperform a grid search, especially if only a small number of hyperparameters affect the performance of the ML algorithm, [42].

Algorithm 2: Random Search for Hyperparameter Optimization

Function `random_search`(*param_distributions*, *n_iter*, *model*, X_{train}, y_{train}, X_{val}, y_{val}, *metric*)**:**

- *best_score* $\leftarrow -\infty$
- *best_params* $\leftarrow$ None
- **for** $i \leftarrow 1$ **to** *n_iter* **do**
  - *params* $\leftarrow$ sample randomly from *param_distributions*
  - configure *model* with *params*
  - $model.fit(X_{train}, y_{train})$
  - $y\_pred \leftarrow model.predict(X_{val})$
  - $score \leftarrow metric(y_{val}, y\_pred)$
  - **if** $score > best\_score$ **then**
    - $best\_score \leftarrow score$
    - $best\_params \leftarrow params$
- print("Best hyperparameters:", *best_params*)
- print("Validation score:", *best_score*)

2.6.1.3 Genetic Algorithm

The genetic algorithm is an evolutionary search technique employed to address complex optimization and modeling tasks by iteratively selecting, recombining, and mutating candidate solutions, mimicking the principles of natural selection. Inspired by biological evolution, it operates on the premise that individuals best adapted to their environment are more likely to survive and propagate. In this context, each individual in the population represents a candidate solution, encoded as a fixed length string commonly using binary, integer, or real valued representations analogous to a chromosome.

The process begins with a randomly generated population of such individuals, each corresponding to a unique set of hyperparameter values. A fitness function is then used to evaluate and rank these individuals, typically based on model performance metrics such as accuracy, loss, or F1-score on a validation dataset. Higher fitness scores indicate better performing configurations. Based on these evaluations, selection methods such as roulette wheel selection, tournament selection, or rank-based selection determine which individuals are most likely to pass their genetic material to the next generation.

Following selection, crossover and mutation operations are applied to generate new individuals (offspring). Crossover recombines segments of parent chromosomes, while mutation introduces random alterations to maintain diversity in the population. This evolutionary cycle continues across successive generations until a convergence criterion is met, such as reaching a maximum number of generations or attaining a satisfactory fitness level. This iterative refinement enables the genetic algorithm to effectively explore large and complex hyperparameter spaces, as highlighted in studies such as (Loussaief and Abdelkrim (2018)) [42]

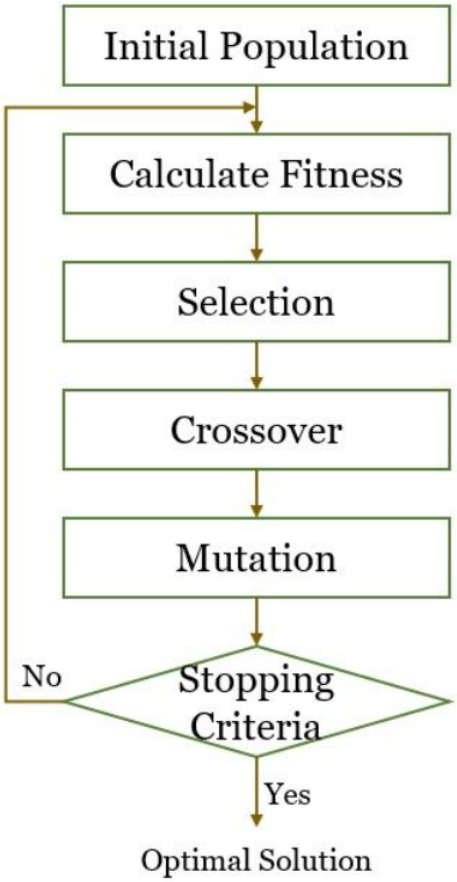


Figure 2.12: Genetic Algorithm Process Phases. [8]

2.7 Conclusion

In this chapter, we introduced the fundamental concepts underlying our research, starting with the role of healthcare and medical imaging, particularly the importance of chest X-rays in disease diagnosis. We then discussed the foundations of AI, ML, and DL, with a focus on CNNs and the principles of TL and FT.

Furthermore, we explored EL and HPO, highlighting their relevance in improving the performance and reliability of deep learning models in medical applications. This theoretical background lays the groundwork for the next chapter, where we review recent state-of-the-art approaches and related works in chest disease classification using DL and ensemble techniques.

3

Related Work

Contents

3.1 Introduction

The previous chapter provided essential background on medical imaging, Deep Learning,Transfer Learning, Ensemble Learning, and hyperparameter optimization techniques. Building on that foundation, this chapter reviews related work on the application of EL and HPO in the healthcare domain. It first analyses studies that explore each technique independently, then highlights the limited but emerging efforts to integrate both approaches for medical image classification.

One notable direction in this context is the use of HPO in EL, which refers to the systematic process of finding the best configuration of parameters that govern how multiple models combine their predictions. Rather than relying on default values or manual adjustments, HPO automates this process to enhance diagnostic performance. In medical applications, this may involve optimizing voting weights in classifier ensembles, tuning architectures in DL stacks, or selecting fusion strategies for multi-modal data.

3.2 Ensemble Learning in Healthcare

1. **Bagging technique**

   This study conducted by **Mahmoud S.al [16]**, the researchers explored DL techniques for the diagnosis of four eye desease: diabetic retinopathy, glaucoma,myopia and normal vision. They employed three CNN architectures, Classical CNN,VGG16 and InceptionV3. The contribution consists evaluation the accuracy of the models and their combinations by using the EL model: bagging(boostrap aggregating). The experiments were conducted using a GPU-enabled Google Colab environment on a dataset comprising 2781 retinal images, which included 975 images for diabetic retinopathy, 721 images for glaucoma, 486 images for myopia and 599 images for normal eye. This images divided into training with 1951 images and 415 images used for validation and 415 images for testing. The main contribution of the paper lies in the comparative evaluation of individual CNN models and their ensemble combinations, where table 3.1 in the original article presents the classification accuracies .

Table 3.1: Prediction accuracy on the testing set for same and different models using ensemble learning. [16]

Model	**Type of ensemble**	**Prediction Accuracy**
Three model of CNN	Bagging	77.1%
Three model of VGG16	Bagging	79.8%
Three model of InceptionV3	Bagging	87.2%
CNN, VGG16 and InceptionV3	Bagging	86.5%

2. **Boosting technique**

   In this study [9], S. Kalaivani et al. presented a three-stage ensemble boosted convolutional neural network model. ResUNet was employed to segment CXR images, which helped improve the model's overall performance. CNN was utilized to extract deep features from the segmented images for the diagnosis of COVID-19. These features were then integrated with various ML classifiers using an ensemble voting approach, achieving better classification results. They constructed a new dataset including CXR images from a variety of sources, such as GitHub, SIRM, TCIA, and Radiopaedia, to enhance the proposed model's classification and generalization capability for COVID-19 identification (see table 3.2).

Table 3.2: Training, validation, and testing of CXR images. [9]

Dataset	**Abnormal**		**Normal**	**Total**
	COVID 19	Other diseased		
Training	2982	756	3017	6755
Validation	852	108	862	1822
Testing	426	54	431	911
Total	**4260**	**918**	**4310**	**9488**

The proposed method (see Figure 3.4)follows structured multi-stage approach for classification of Chest-xray images. Initially, the system prepares the chest X-ray images by resizing them, normalizing the data, and segmenting the important parts to focus on key features. Next, it uses a CNN to automatically extract meaningful patterns from the images. In the final stage, the extracted features are passed through a group of classifiers—Decision Tree, Random Forest, AdaBoost, and SVM—which work together through an ensemble approach to make an accurate prediction, identifying whether the image shows signs of COVID or is normal

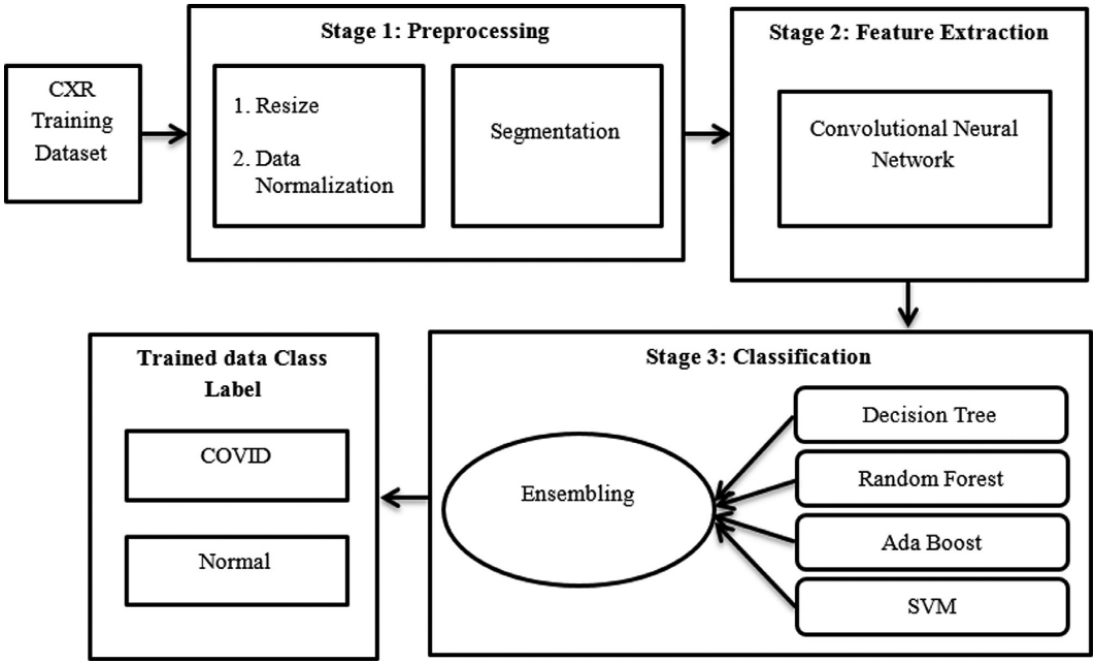


Figure 3.1: Overview of Proposed model. [9]

The Ensemble boosted CNN achieved an overall accuracy of 99.35% , precesion of 100% ,recall of 98.71% and 99.35% for the metric of F1-Score.

3. **Stacking technique**
The stacking technique is a widely used ensemble method that combines the outputs of multiple base learners through a meta-learner. It has been applied in several studies to enhance classification performance, particularly in medical imaging and diagnosis.

- **Stacking-Based Model by Mamar.K et al**

In this study [17], The authors developed and tested multiple DL-based approaches for automatic classification of medical imaging modalities and anatomical organs. They introduced a comparative framework involving deepTL using six popular CNN architecture, applied in both feature extraction and FT modes. additionally, they explored a third approach that leverages stacking EL to combine the outputs of the individual CNNs for improved predective performance. Two datasets used in their experiments: the first called **MC4_Dataset (Size and average challenges)** This image dataset was created by them from nine public image datasets (size of 35150 images). Then, this dataset divided into 4 classes depending on the imaging modality used (MRI 7023 images, ultrasound 2116 images, CT-Scan 12988 images, X-ray: 12023 images). Each class is divided into subclasses according to the different anatomical organs. The second one called **RSNA-MICCAI dataset (high size and challenges)** it is a public dataset "RSNA-MICCAI Brain Tumor Radio-genomic Classification includes 290923 images. The experimental results demonstrate the effectiveness of the proposed stacking method with 91% for accuracy, 95% for precesion, 89% dor recall and 92% for F1-Score.
The results of their DL models are presented in table 3.3.

Table 3.3: Performance comparison of different DL models. [17]

Model	Accuracy	Precision	Recall	F1-Score
VGG16	0.58	0.75	0.24	0.37
VGG19	0.50	0.73	0.24	0.36
Resnet 50	0.63	0.70	0.41	0.51
Inception 3	0.84	0.89	0.81	0.85
XCeption	0.88	0.89	0.87	0.88
NASNet	0.86	0.91	0.82	0.87
Stacking Model	0.91	0.95	0.89	0.92

- **Stacking base-model by Ruaa N. Sadoon et al**

In this Study [10], The researchers proposed an advanced model for devastating impact of the COVID-19 pandemic and numerous pulmonary pathologies, combining multiple CNN architectures (DenseNet, Xception, Inception) into a stacked ensemble model optimized throughTL. Using An ANN acts as a meta-learner to combine the outputs of the individual base learners and improve classification performance. Two large-scale chest X-ray datasets

were used to train and evaluate the model: the COVIDx CXR-4 dataset, comprising over 30.000 radiographic images used for COVID-19 and pneumonia detection, and the ChestX-ray14 dataset, containing more than 100.000 chest radiographs spanning a wide range of pulmonary condition. The diagram in Figure 3.2 presents the methodology of the study, the process starts with the collection and preprocessing the data, including exploratory data analysis. Following preprossing of three CNN architectures -DenseNet, Inception,Xception- are independently trained on the prepared data. Each model has an intial evaluation to assess individual performance. subsequently, Feauture extraction from these base models are combined in stacking ensemble framework, where a meta learner integrats their predection. The final stage involvesa comprehensive evaluation of the EL. The experimental results demonstrate the superiority of the stacking model over individual CNNs, achieving an accuracy of 98%, with corresponding precision, recall and F1-Score values also reaching 0.98. The best result of individual model for DenseNet201 with 95% accuracy, while InceptionV3 with 85% then Xception trailed behind with 81%.

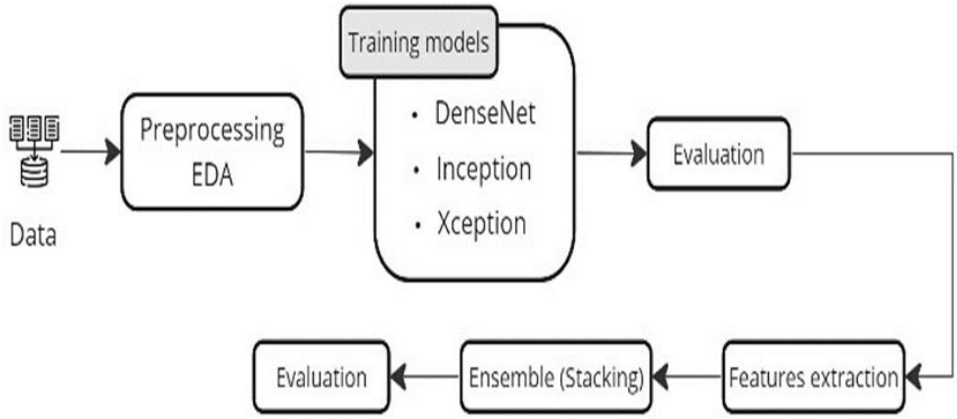


Figure 3.2: Methodology of the study. [10]

4. **Voting technique**

In this paper [11], **Nazik.Al et al** explored a hybrid system for brain tumor detection using MRI images. The researchers utilized a publicly available dataset from kaggle, which contains 3762 images classified into two categories: 1683 tumor and 2079 non-tumor cases. The figure 3.3 illustrates the proposed methodology. which follows a two stage approach: in the first scenario, all 13 available features were directly used to train classifiers. in the second, deep features were extracted using a custom CNN, and then a voting classifier combining a Logistic Regression(LR) and Stochastic Gradient Descent(SGD) was used for classification. The dataset was split into 70% for training and 30% for testing. The model's performance was evaluated using metrics such as accuracy, precision, recall and F1-Score demonstrating the effectiveness of the hybrid approach.

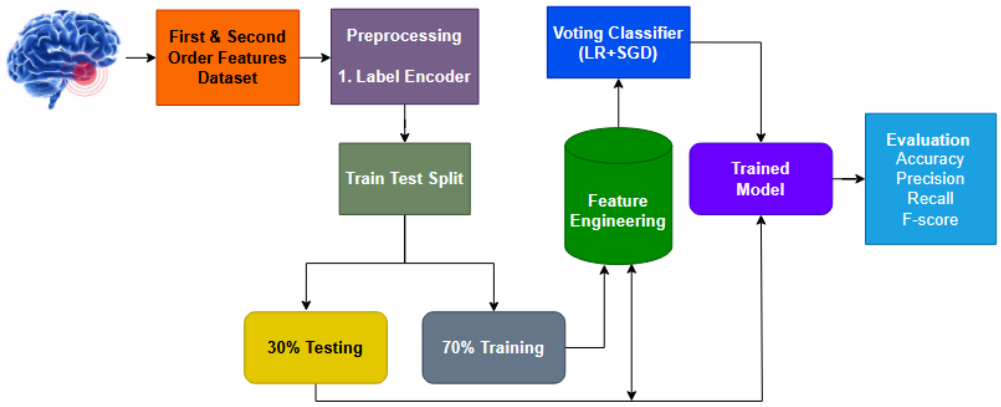


Figure 3.3: workflow diagram of the proposed voting classifier (LR+SGD) model. [11]

All the experiments was performed using a 7th generation Intel Corei7 machine with Windows 10 operating system. Python language was used for the implementation of the proposed approach and the other learning models. With the support of libraries such as Tensor Flow, Sci-kit learn, and Keras . The results demonstrated that the approach achieved a remarkable accuracy of 99.9%. This performance surpassed other traditional ML algorithms.

The table 3.4 below provides a comparative summary of selected state-of-the-art studies that applied EL techniques with various DL models in medical image classification.

Table 3.4: Comparative summary of related work.

Authors	**Year**	**Dataset(s)**	**Technique**	**Pre-Trained Models**	**Accuracy**
Mahmoud S. et al	2020	Diabetic Retin iChallenge-GON	Bagging	CNN, VGG16, InceptionV3	86.5%
S. Kalaivani et al	2022	CXR	Boosting	CNN, ResUNet	99.35%
Mamar K. et al	2023	MC4 RSNA-MICCAI	Stacking	VGG16, VGG19, ResNet50 Xception, Inception, NASNet	91%
Ruaa N. Sadoon et al	2024	COVIDx CXR-4, ChestX-ray14	Stacking	DenseNet201, Xception, InceptionV3	98%
Nazik A. et al	2023	Brain Tumor	Voting	CNN	86%

3.3 Hyperparameters optimization in Healthcare

HPO is a necessary step to ensure the best possible performance of ML algorithms, including CNNs [43].

In the context of CNNs, HPO involves selecting the optimal combination of parameters such as learning rate, batch size, and number of filters to achieve the highest classification accuracy. However, this process is particularly challenging due to the large number of parameters involved, making it time-consuming and computationally expensive [44]. **1.** In the first study [18], Hashi et al proposed a heart disease prediction system using multiple ML algorithms such as Logistic Regression, K-Nearest Neighbors, SVM, Decision Tree, and Random Forest. The study demonstrated the impact of hyperparameter tuning using grid search, showing notable accuracy improvements over untuned models when applied to the Cleveland dataset. This dataset [1-3] has been collected from the UCI ML repository that has used for both training and testing purposes. It contains 303 instances and 75 attributes, but this work considers a feature subset of 14 numerical valued attributes. The output level has two classes, where 0 represents not having heart disease, and 1 represents having heart disease. Their experiments revealed that hyperparameter tuning led to significant improvements in model accuracy. Specifically, Accuracy increased from 81.97% to 90.16% for untuned models and from 85.25% to 91.80% for tuned models using Grid Search for hyperparameter optimization, as illustrated in Table 3.5. [18].

Table 3.5: Performance comparison of classification models on training and test datasets. [18]

Algorithm	**Parameters**	**Training**				**Test**			
		Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
LR	C=1, solver='liblinear'	87.60	87.05	90.98	88.97	88.52	90.32	87.50	88.89
KNN	Neighbors=5, weights='uniform'	87.60	87.59	90.23	88.89	90.16	90.62	90.62	90.62
SVM	kernel='rbf', gamma=0.001, C=2.0	81.82	77.30	94.74	85.14	88.52	87.88	90.62	89.23
DT	criterion='gini', min_samples_leaf=1, min_samples_split=2, splitter='best'	100	100	100	100	81.97	86.21	78.12	81.97
RF	criterion='gini', min_samples_leaf=1, min_samples_split=2, estimators=1000	100	100	100	100	85.25	87.10	84.38	85.71

2. awotunde et al. Propose a paper with title 'An Enhanced Hyper-Parameter Optimization of a CNN Model for Leukemia Cancer Diagnosis in a Smart Healthcare System' [12]. This study introduced an IoMT-enabled CNN framework designed for accurate detection of leukemia

cancer. The CNN was optimized using the HORD algorithm, which combines Radial Basis Function modeling with Dynamic Coordinate Search. This optimization strategy significantly improved classification performance on high-dimensional gene expression data.
The following figure 3.4 displays the proposed framework for the IoMT-based cancer diagnostics architectural design using the proposed hyper-parameter-optimized CNN classifier [12]. The specific hyperparameters utilized in this model, along with their descriptions and corresponding value ranges, are summarized in Table 3.6.

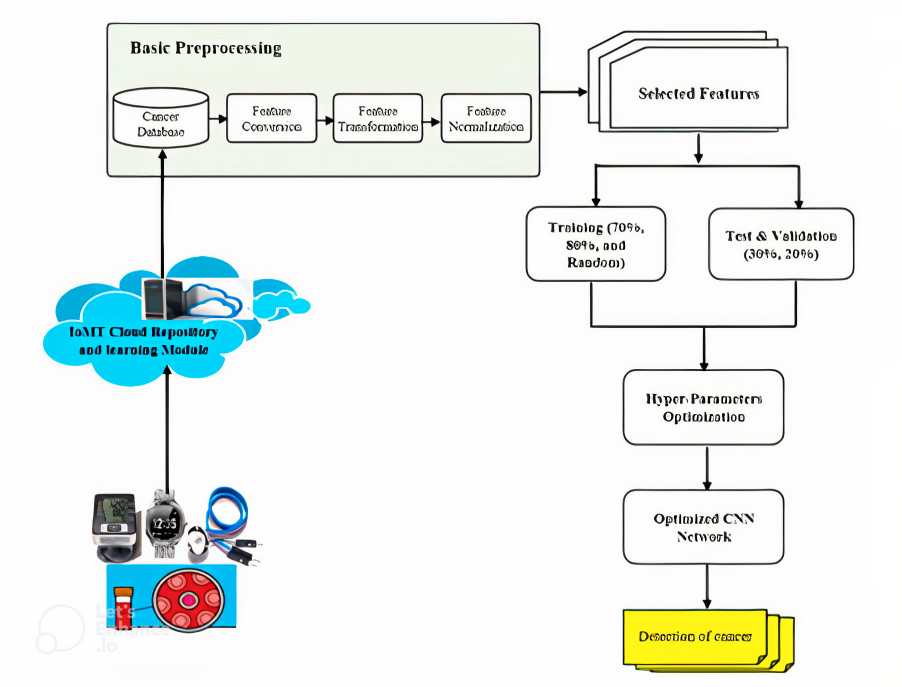


Figure 3.4: The proposed framework. [12]

Table 3.6: List of hyperparameters used in the model, including descriptions and value ranges [?].

Hyper-Parameter	Explanation	Range
Neuron Count	The number of neurons in the top convolutional layers	8, 16, 32
Layer Depth	The total number of layers in the network	1, 2, 3
Kernel Size	Size of the convolutional layer's kernel	1, 2, 3
Stride	Quantity of shifting kernel pixels during convolution	1, 2, 3
Activation Function	The process of activating neurons	Sigmoid, ReLU, SeLU
Batch Size	Number of training data divisions per group	8, 16, 32
Kernel Count	Number of convolutional layer kernels	8, 16, 32
Epoch	Numerous iterations of learning	20, 50, 100
Learning Rate	Updated weight during learning	0.01, 0.001, 0.0001
Loss Function	A method for calculating error	L2 loss, Binary cross-entropy

The dataset used in this study appealed The UCI leukemia dataset contains 7,129 genes from 72 patients (25 AML, 47 ALL). Pre-normalized data was split into 70-30 and 80-20 random train-test partitions to evaluate classification models handling high-dimensional genomic data and class imbalance. [12]

Table 3.7: Detail description of the leukemia datasets. [12]

Dataset	**Number of Genes**	**Samples**	**Classes**
AML-ALL	7129	72	2
AML-ALL-CML	7129	72	3
AML-ALL-CML-CLL	7129	72	4

For experimentals result the table 3.7 displays the proposed model's prediction performance for ALL and healthy cases,revealing the accuracy to be 99.9 and 100,respectively.Theprecision,recall,and F1score were also 100 or1.0. The prediction accuracy for CLL was 99.8%, the recall was 98.8%,the specificity was 100%,the F1-score was 99.8%,and the precision was 100%, respectively.The prediction accuracy rate for AML was 99.9%,and the precision,recall, and F1score were 100%.The dataset was divided into 80% for training and 20% for testing (validation),respectively. [12]

3. In this paper [13]. Asiri et al. proposed a customized CNN model for brain tumor classification, emphasizing the importance of tuning critical hyperparameters such as filter size,

stride, padding, pooling, activation functions, learning rate, and batch size. The model's hyperparameter optimization led to significant accuracy gains over baseline pre-trained models on multiple MRI datasets. Dataset 1 consists of 7,023 brain MRI images classified into four categories: glioma, meningioma, pituitary, and no tumor. Dataset 2 contains 253 additional brain MRI images for further evaluation. [13]
As a methodology, This diagram Figure 3.5 represents a systematic approach to hyperparameters tuning, often used in DL experiments to improve model performance.

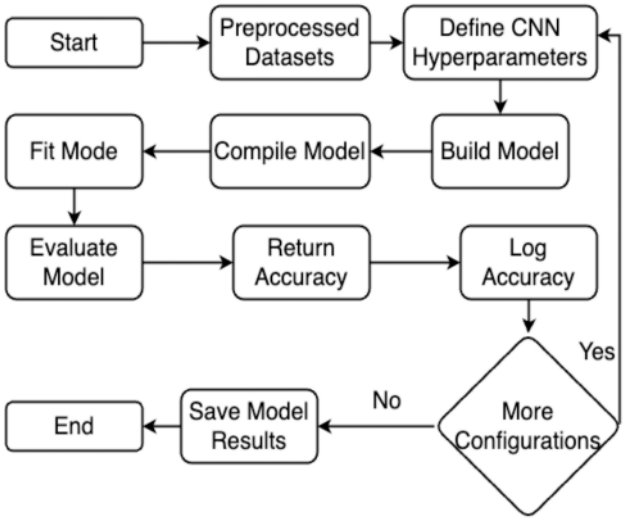


Figure 3.5: Flowchart of the proposed work. [13]

The process begins with dataset preprocessing followed by defining the CNN hyperparameters. The model is then buid , compiled, trained and evaluated. The accuracy of model is returned and logged. A decision point checks whether more configuration needs to be tested. if yes, the process loops back to defining a new hyperparameters. if no, The model results are saved and the workflow ends.
As results, the optimized CNN achieved 98.04% accuracy, with 98% precision, recall, and F1-score, outperforming VGG16, ResNet50, and others. [13]

4. Morales-Hernández et al. present an extensive survey on multi-objective hyperparameter optimization (MO-HPO) techniques, classifying them primarily into metaheuristic based and metamodel based approaches [14]. They begin by explaining how HPO algorithms work as wrappers around ML models, controlling their hyperparameters to optimize a chosen performance metric. This process is clearly illustrated in Figure 3.6, where an HPO algorithm configures the hyperparameters such as learning rate, number of neurons, and training epochs for an artificial neural network (ANN) predicting house prices from input features (e.g., square meters, rooms, region). The performance is evaluated based on the root mean square error (RMSE), highlighting how the choice of hyperparameters directly affects model accuracy. In metaheuristic based approaches, solution candidates are iteratively generated and evaluated using techniques like Genetic Algorithms, Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and NSGA-II. The latter, NSGA-II, is the most cited due to its efficiency in exploring Pareto optimal fronts using elitism and non-dominated sorting.

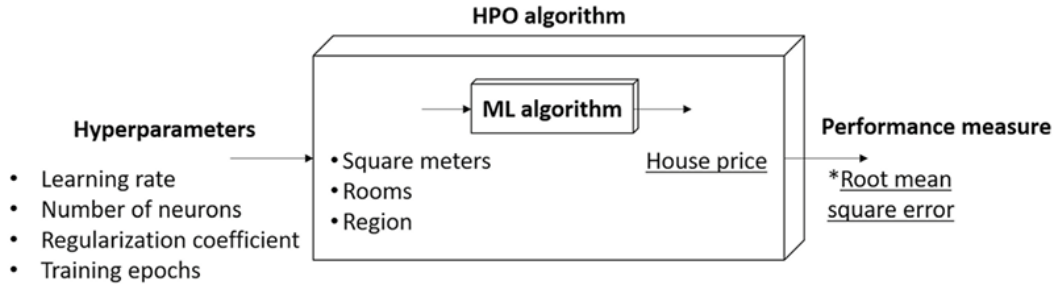


Figure 3.6: Example of the interplay between the HPO algorithm and the target ML algorithm. [14]

The authors also detail metamodel based methods, which reduce the computational cost of HPO by using surrogate models like Gaussian Processes (GP), Random Forests (RF), or Tree structured Parzen Estimators (TPE). These models approximate the performance landscape to guide the search using acquisition functions such as Expected Improvement (EI), Probability of Improvement (POI), Upper/Lower Confidence Bound (UCB/LCB), and Predictive Entropy Search (PES). Their methodology includes comparing various MO-HPO algorithms across numerous ML applications like image recognition, speech processing, and medical diagnostics, covering metrics like accuracy, complexity, runtime, and model size. they provide systematic comparisons and visual explanations of the procedures. The survey emphasizes the adaptability of methods like NSGA-II, MOEA/D, CMA-ES, and PESMO, noting their varying strengths depending on whether computational efficiency, solution diversity, or user defined constraints are prioritized. The table 3.8 presents the Hyperparameters optimization techniques in different studies.

Table 3.8: Comparison of Hyperparameter Optimization Techniques in Healthcare-Related Studies

Authors	**year**	**Dataset**	**Model Architecture**	**HPO Techniques**	**Performance Metrics**
Emrana Kabir Hashi et al.	2020	Cleveland Heart Disease dataset (303 samples)	Logistic Regression, KNN, SVM, Decision Tree, Random Forest	Grid Search	Accuracy: 90.16% (best model)
Joseph Bamidele Awotunde et al.	2022	Leukemia gene expression dataset (72 samples, 7129 features)	Convolutional Neural Network (CNN)	HORD (Radial basis + Dynamic coordinate search)	Accuracy: 99.9% Precision/Recall/F1: 100%
Alejandro Morales-Hernández et al.	2022	Not applicable	Various ML models	Multi-objective algorithms (e.g., NSGA-II, SPEA2)	Not applicable
Abdullah A. Asiri et al.	2024	Dataset 1: 7,023 MRI images Dataset 2: 253 MRI images	CNN	Manual tuning: filter size, stride, pooling, activations, learning rate, batch size	Accuracy: 96% (DS1), 88% (DS2)

S

3.4 HPO in EL within Healthcare

1. The study by Asif et al. aimed to enhance heart disease prediction by combining EL methods with hyperparameter optimization [15]. The authors developed a ML-based system that uses EL with hyperparameter optimization to predict heart disease. The authors combined three datasets from Kaggle, cleaned and normalized the data, and evaluated multiple ensemble models. They applied GridSearchCV and RandomizedSearchCV to tune the hyperparameters. The Extra Trees classifier showed the best results with 98.15% accuracy, outperforming other models. This work highlights the potential of optimized ensemble models for improving clinical prediction systems.

Dataset Description

Three datasets with similar features were merged to form a single dataset with 1625 samples. The datasets and their sizes are listed in the table 3.9.

Table 3.9: Merged Heart Disease Datasets. [15]

Dataset Name	**Source**	**Records**
Cleveland Heart Disease Dataset	Kaggle	297
Heart Disease UCI Dataset	Kaggle	303
Heart Disease Dataset	Kaggle	1025
Total Combined Dataset	—	**1625**

Before modeling, preprocessing was performed to remove duplicates and missing values. Min-Max normalization was applied to bring all features into a 0–1 range. Finally, the dataset was split into 80% for training and 20% for testing.

The methodology followed a typical ML pipeline, but focused strongly on optimization and EL. First, three datasets were collected and merged based on matching feature columns. Data cleaning included removing any duplicates or missing entries. Then, normalization was done using Min-Max scaling to standardize feature values. After splitting the data, the authors trained four ensemble classifiers: Random Forest, Extra Trees, XGBoost, and CatBoost. For each classifier, both default and optimized models were tested. Hyperparameter optimization was conducted using GridSearchCV and RandomizedSearchCV techniques. The models were then evaluated using metrics like accuracy, recall, precision, F1-score, ROC-AUC, PR curves, and Cohen's kappa score. Visual tools like violin plots and ROC curves were also used to compare the models' performance and consistency. The proposed model of this study is presented in the Figure 3.7

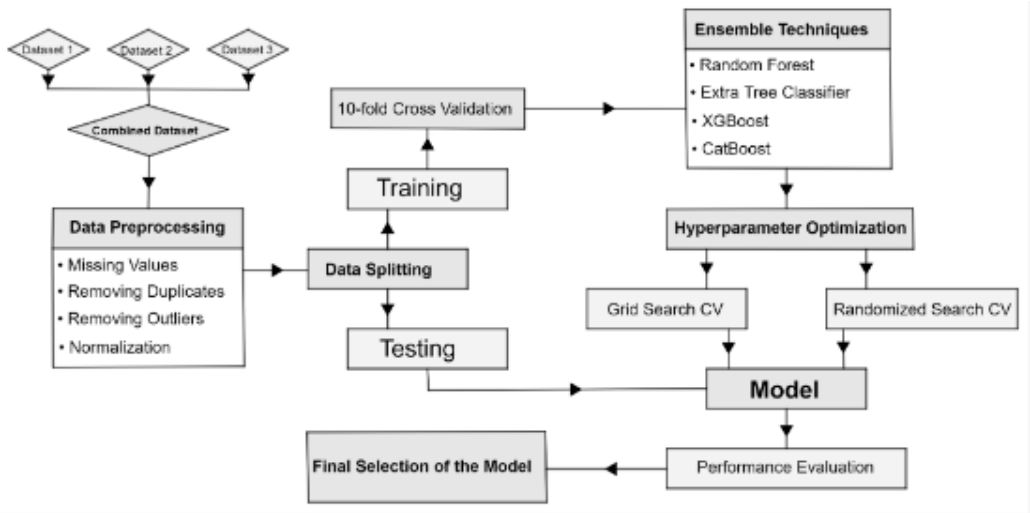
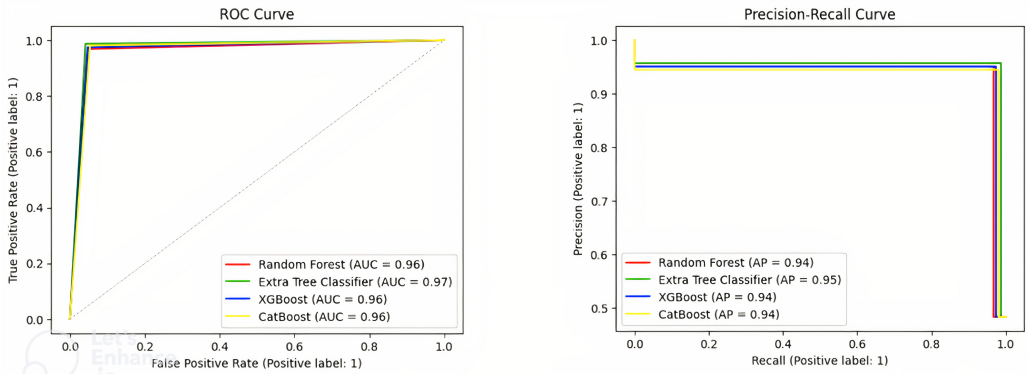


Figure 3.7: The proposed model developed to predict heart disease. [15]

The Experimental Results demonstrated that, amoung all tested models, the Extra Trees classifier showed the best results, especially when optimized using GridSearchCV. The results of all models are presented in the following table 3.10

Table 3.10: Model Performance Comparison. [15]

Model	**Tuning Method**	**Accuracy (%)**
Random Forest	Default	96.70
Random Forest	GridSearchCV	97.06
XGBoost	RandomizedSearchCV	96.38
CatBoost	GridSearchCV	97.00
Extra Trees	Default	97.23
Extra Trees	RandomizedSearchCV	97.54
Extra Trees	**GridSearchCV**	**98.15**



(a) The area under the ROC curve.

(b) The precision–recall curve.

Figure 3.8: ROC and PR curves of all algorithms using default hyperparameter settings. [15]

Figure 3.8 shows the ROC-AUC and PR curves of the evaluated models using their default hyperparameters. These curves indicate that the Extra Trees classifier consistently outperformed the others in terms of both precision-recall balance and discriminatory power. Additionally, violin plots (not shown here) demonstrated stable score distributions across cross-validation folds, and Cohen's kappa score further validated the reliability of the model's predictions.

The study showed that optimized ensemble models, especially Extra Trees with tuned hyperparameters, can achieve high prediction accuracy for heart disease. This approach, with proper preprocessing and parameter tuning, can be integrated into clinical tools to support early diagnosis and reduce the risks associated with late detection.

2. The paper [19] titled "Regularized Boosting with an Increasing Coefficient Magnitude Stop Criterion as Meta-Learner in Hyperparameter Optimization Stacking Ensemble" presents a novel approach to enhance the performance of ML models by integrating EL with HPO. Traditionally, HPO procedures select and retain only the single best-performing model, discarding all others. In contrast, this study proposes leveraging all models generated during HPO by incorporating them into a stacking ensemble, where a meta-learner is trained to combine their predictions. The authors introduce a new meta-learner called RBOOST (Regularized Boosting), which applies boosting in a feature-wise manner, effectively handling multicollinearity—a common issue in HPO generated ensembles. To prevent overfitting, they also propose a novel stop criterion named ICM (Increasing Coefficient Magnitude), which halts the boosting process when the magnitude of coefficients begins to rise, signaling potential overfitting.

The table 3.11 provides an overview of the datasets used in the experiments, including their names, the number of instances (Inst.), and the number of features (Feat.) for each. A total of 15 datasets were selected from the UCI repository. These datasets vary in size and complexity: some have a small number of instances, such as Fertility (100 instances) and Slump (103), while others, like White wine quality (4898) and Abalone (4177), contain several thousand. Feature dimensionality also differs across datasets, ranging from as few as 5 features (Servo) to as many as 119 features (Communities and Crime). This collection of datasets ensures that the experimental evaluation covers a broad range of real-world data scenarios.

Table 3.11: Number of instances and features for UCI repository datasets. [19]

Dataset	**Inst.**	**Feat.**	**Dataset**	**Inst.**	**Feat.**
Abalone	4177	11	Forest	517	13
Airfoil Self Noise	1503	6	Qsar	908	7
Auto MPG	392	8	Servo	167	5
Automobile	158	26	Slump	103	8
Concrete Data	1030	9	Traffic	135	18
Com. and crime	1993	119	Red wine quality	1599	12
Fertility	100	10	White wine quality	4898	12
Flow	103	8			

The models used in the experiments include Ridge Regression, Support Vector Regression, and Random Forest Regressor, while HPO strategies include Grid Search, Random Search, Bayesian Optimization, Particle Swarm Optimization, and Hyperband.

The Table 3.12 presents the averaged Friedman ranks of various ensemble methods across different hyperparameter optimization (HPO) techniques (GS, RS, BO, PSO, HB) and machine learning models (Ridge, SVR, RFR). The lower the rank, the better the method performs in terms of relative mean squared error (MSE). RBOOST (ICM) consistently achieves the best performance, with the lowest overall mean rank of 2.34 across all configurations. This outperforms Caruana's ensemble, which is the second-best with a mean rank of 2.65, and significantly surpasses simpler ensemble methods like BEM (4.84), IEW (3.98), and GEM (3.14).

Notably, RBOOST (ICM) achieves the best average rank in nearly all rows. For example, under PSO optimization with Ridge, SVR, and RFR models, it achieves exceptionally low ranks of 2.20, 1.53, and 1.80, respectively — clearly outperforming all competitors. Even in more challenging scenarios like GS or BO, RBOOST (ICM) maintains strong performance, with ranks such as 2.00 (GS-Ridge), 2.20 (GS-SVR), and 2.47 (BO-Ridge/SVR).

These results highlight the advantage of RBOOST (ICM)'s regularization-driven stacking, which avoids overfitting while effectively combining base learners. Compared to the Best single-model selector, which has a mean rank of 4.04, RBOOST (ICM) demonstrates that meta-learning with adaptive regularization significantly improves generalization across different optimization strategies and base models.

Table 3.12: Averaged Friedman ranks for the relative mean squared error over all datasets. [19]

HPO	**MLS**	**Best**	**BEM**	**IEW**	**GEM**	**Caruana**	**RBOOST (ICM)**
GS	Ridge	2.97	4.93	4.47	2.97	2.00	**2.00**
	SVR	3.40	5.07	4.27	3.47	2.60	**2.20**
	RFR	3.80	4.77	3.87	2.90	**2.67**	3.00
Mean	GS	3.39	4.92	4.20	3.34	2.42	**2.40**
RS	Ridge	3.20	4.87	4.53	3.37	**2.67**	**2.37**
	SVR	3.27	4.90	3.67	3.40	**2.40**	2.50
	RFR	3.33	4.93	3.93	3.00	**2.90**	**2.90**
Mean	RS	3.27	4.90	4.04	3.26	**2.66**	**2.59**
BO	Ridge	3.10	4.40	4.00	3.67	**2.33**	**2.47**
	SVR	3.60	4.60	4.00	3.33	**2.93**	2.47
	RFR	3.87	4.87	4.00	2.73	**2.73**	**2.53**
Mean	BO	3.52	4.62	4.00	3.24	**2.66**	**2.49**
PSO	Ridge	5.20	4.60	3.87	2.93	2.47	**2.20**
	SVR	5.93	4.53	3.67	3.00	**2.17**	**1.53**
	RFR	6.00	4.53	3.47	2.73	**2.20**	**1.80**
Mean	PSO	5.71	4.56	3.67	2.89	**2.28**	**1.84**
Mean	→ total	**4.04**	**4.84**	**3.98**	**3.14**	**2.65**	**2.34**

3.5 Conclusion

The reviewed literature highlighted the increasing use of EL and hyperparameter optimization techniques in healthcare related ML tasks. Ensemble approaches including Stacking, Bagging, Random Forest, Extra Trees, and XGBoost demonstrate strong performance by integrating predictions from multiple base models to enhance accuracy and robustness. In parallel, hyperparameter optimization methods such as GridSearchCV, RandomizedSearchCV, and Genetic Algorithms are widely employed to systematically tune model parameters and adapt learning processes to specific clinical challenges.
These insights have shaped our strategy, which leverages optimized ensemble methods to improve classification outcomes on chest X-ray datasets. By combining fine-tuned CNN architectures with voting and stacking strategies, and integrating hyperparameter tuning, our system aims to address the challenges of generalization and model selection. The next chapter describe the implementation and evaluation of this approach.

4

Our contribution

Contents

4.1 Introduction

Building upon the previous chapter, which reviewed existing literature on DL applications for image classification in healthcare and highlighted common limitations such as reliance on single models and insufficient generalization, this chapter proposes a DL-based approach to enhance disease classification performance.

We present our main contributions: a structured framework that begins by selecting three pretrained CNN models—DenseNet201, ResNet101, and VGG16—and a multiclass chest X-ray dataset. Hyperparameter optimization (HPO) techniques, including Random Search and Genetic Algorithm, are then employed to fine-tune each model. The optimized networks are trained on the dataset, and their outputs are integrated using ensemble learning (EL) strategies such as soft voting and stacking to enhance classification performance and robustness.

This chapter details the methodology, training strategy, and results, aiming to achieve more accurate and robust classification.

4.2 Approach and Experimentation

To demonstrate our contribution, we followed a structured experimental workflow (see Figure 4.1) that begins with data collection and preprocessing, setting up the development environment and tools. We then applied hyperparameter optimization (HPO) to fine-tune model

configurations, selected and trained DL models using the optimized parameters, and finally applied ensemble learning (EL) techniques to combine their outputs.

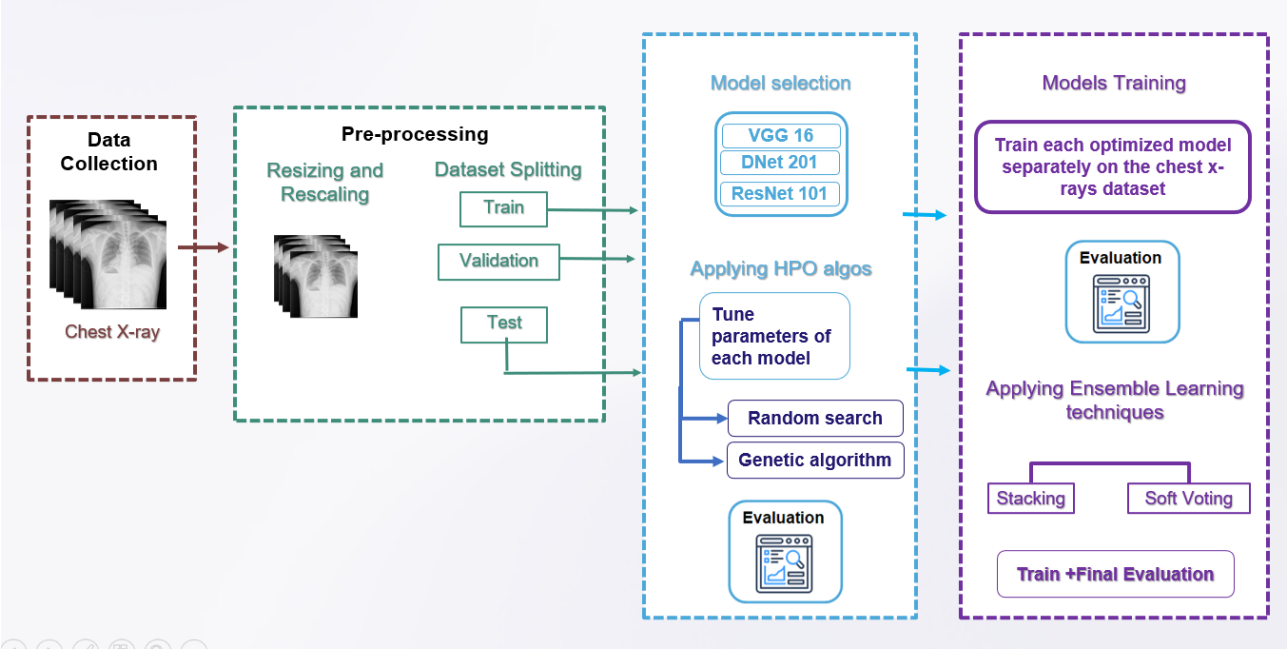


Figure 4.1: Our experimental workflow

4.2.1 Experimental Environment and Tools

As part of the experimental setup, it was essential to configure an appropriate software environment to support model development, training, and evaluation. The tools and libraries selected played a critical role in ensuring smooth implementation and reproducibility of the experiments, as outlined below.

Software Setup

The experiments employed widely used machine learning frameworks and programming languages to facilitate reproducibility and effective implementation:

- **TensorFlow:** An end-to-end open-source platform for machine learning that facilitates building and deploying ML models efficiently [45]. We used TensorFlow version 2.10.1 in our experiments.
- **Keras:** A high-level deep learning API, running on top of TensorFlow, used to define and train our neural network architectures. We utilized Keras version 2.10.0 [46].
- **Python:** All model development, training, and evaluation were carried out using Python version 3.8.20.

Hardware Setup

The computational resources employed for training and evaluating the models consisted of both high-performance local hardware and cloud-based platforms. These resources were selected

to balance computational efficiency and accessibility, enabling effective experimentation across different environments:

- **Local workstation:** We used an NVIDIA Quadro RTX 6000 GPU with 24GB of VRAM and Intel Xeon Gold 6138 CPU with 130Go of RAM for model training and evaluation.
- **Google Colab:** We utilized the free version of Google Colab, which provided access to T4GPU or running and testing our models in a cloud-based environment.

4.2.2 Data Collection

To evaluate the robustness and generalizability of our models, we conducted experiments on the Chest X-Ray Multiclass Dataset which is provided by Rifatul Islam Majumder publicly available on kaggle [20].The dataset originally includes both chest X-ray and non-X-ray images; however, for the purpose of developing AI models focused solely on medical imaging, all non-X-ray images were excluded. The refined dataset is thus limited to chest X-rays categorized into the following classes [20]: Normal, Pneumonia, Tuberculosis (TB).
This dataset combines chest X-rays from various global sources (Kermany, RSNA, NIAID, NLM, Belarus).

The following figures (4.2 to 4.4) presents examples of the images of the dataset

Figure 4.2: Normal images of the DATASET

[20]

Figure 4.3: Pneumonia images of the DATASET

[20]

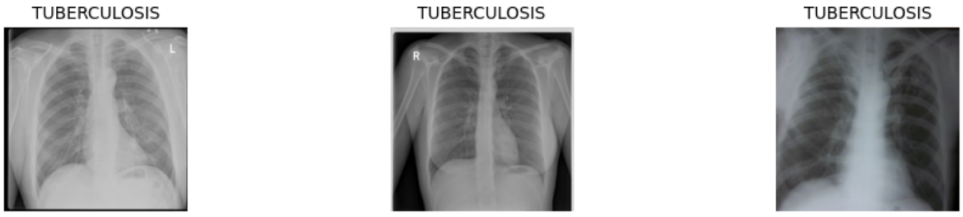


Figure 4.4: Tuberculosis images of the DATASET

[20]

Dataset preprocessing

Data splitting: The Chest X-ray dataset was divided into training, validation, and testing sets to ensure a robust evaluation pipeline. The training set was used to help the model learn meaningful patterns, the validation set was employed during training to fine-tune hyperparameters and reduce overfitting, and the testing set evaluated the model's generalization to unseen data. The detailed split is presented in table 4.1.

Table 4.1: Data Split for Chest X-ray Dataset. [20]

Classe	**Train**	**Val**	**Test**	**Total**
NORMAL	4667	548	247	5462
PNEUMONIA	3633	427	213	4273
TUBERCULOSIS	3573	417	207	4197
Total	11873	694	1392	13717

Data resizing and rescaling: To ensure consistency with the input requirements of pretrained convolutional neural networks, only basic preprocessing steps were applied to the chest X-ray images:

- **Resizing:** All images were resized to 224 × 224 pixels to match the expected input dimensions of the models.
- **Rescaling:** Pixel values were normalized to the range [0, 1] by dividing each value by 255.

4.2.3 Evaluation Metrics

Before initiating the training process and analyzing the results, it was essential to identify and establish the evaluation metrics that would objectively assess the performance of our classification models. These metrics serve as a foundation for interpreting how well the models distinguish between different classes, particularly in a medical imaging context where misclassifications can have serious implications.

- **True Positive (TP)**: correctly predicted positive cases.
- **True Negative (TN)**: correctly predicted negative cases.
- **False Positive (FP)**: negative cases incorrectly predicted as positive.
- **False Negative (FN)**: positive cases incorrectly predicted as negative.

The metrics are computed as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{4.1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\% \tag{4.2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\% \tag{4.3}$$

$$F1\text{-score} = \frac{2 \cdot (\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}} \tag{4.4}$$

These metrics are derived from the confusion matrix shown in table 4.2:

Table 4.2: The confusion matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

4.2.4 Hyperparameter Optimization

To ensure optimal performance, both Random Search and Genetic Algorithm (GA)-based HPO techniques were applied prior to model training and ensemble integration. These methods were used to tune key parameters such as learning rate, dropout rate, and dense layer size for each individual model—DenseNet201, VGG16, and ResNet101—on the multiclass chest X-ray dataset.
By conducting HPO beforehand, we ensured that each model operated under its best configuration, providing robust and well-calibrated components forEL. This setup also allowed us to investigate later the impact of optimized base models on the overall performance of ensemble techniques.

- **Random Search**

  The performance of a model depends on the choice of hyperparameters. We selected this technique to observe how changes in hyperparameters affect the model's accuracy and overall performance.
  In our approach, we defined the hyperparameter search space, which included the learning rate, base model, dropout rate, and number of dense units. The learning rate was explored within the following range:

```
learning_rate:  [1e-4, 5e-5, 1e-5]
```

After applying the Random Search technique, the best-performing configuration for each model was identified based on its performance on the validation set. The table 4.3 below summarizes the five top-performing random search runs, where different combinations of hyperparameters were evaluated to measure their impact on classification accuracy.

Table 4.3: Top 5 Random Search runs with optimal hyperparameters and corresponding accuracy.

Run	Learning Rate	Dropout	Dense Units	Base Model	Accuracy
1	1e-04	0.3	256	DenseNet201	91.82%
2	5e-05	0.4	512	VGG16	94.41%
3	1e-05	0.5	512	ResNet101	81.23%
4	1e-04	0.3	512	DenseNet201	95.13%
5	1e-04	0.6	256	VGG16	92.88%

Among the top configurations, the highest accuracy (95.13%) was achieved in Run 4 using the DenseNet201 base model, with a learning rate of 1×10^{-4}, a dropout rate of 0.3, and 512 dense units. Other models, such as VGG16 and ResNet101, also achieved reasonable performance, with accuracies ranging from 81.23% to 94.41These results suggest that hyperparameter tuning particularly of learning rate, dropout, and dense layer size can significantly influence model performance. While models like DenseNet201 and VGG16 benefited noticeably from HPO, others, such as ResNet101, showed more limited improvement. This indicates that the effectiveness of HPO may vary depending on the architecture and its compatibility with the dataset. Nonetheless, performing optimization before the training phase remains a critical step, especially when preparing models for integration into an ensemble framework.
The classification report in Table 4.4 confirms that the models tuned with the Random Search technique deliver strong, balanced performance across all three classes. Overall accuracy reaches 96 % on the validation set. For the NORMAL class, the model attains a precision of 0.96, a recall of 0.95, and an F1-score of 0.95. The TUBERCULOSIS class records a precision of 0.96, a recall of 0.97, and an F1-score of 0.96. The PNEUMONIA class achieves the best results, with a precision of 0.97, a recall of 0.98, and an F1-score of 0.97.

The corresponding confusion matrix shown in figure 4.5 reinforces these findings. The matrix shows that the vast majority of predictions lie on the main diagonal, indicating correct classification. Misclassifications are minimal: a moderate number of Normal images are confused with Tuberculosis, and to a lesser extent with Pneumonia, likely reflecting overlapping radiographic features between these conditions. Nonetheless, the low off-diagonal counts demonstrate that the optimized model distinguishes effectively among the three disease categories, delivering high reliability in practical diagnostic settings.

Table 4.4: Classification report for Random Search Technique.

Classe	**Précision**	**Recall**	**F1-score**	**Effectif**
NORMAL	0.96	0.95	0.95	548
TUBERCULOSIS	0.96	0.97	0.96	417
PNEUMONIA	0.97	0.98	0.97	427
Accuracy		0.96		

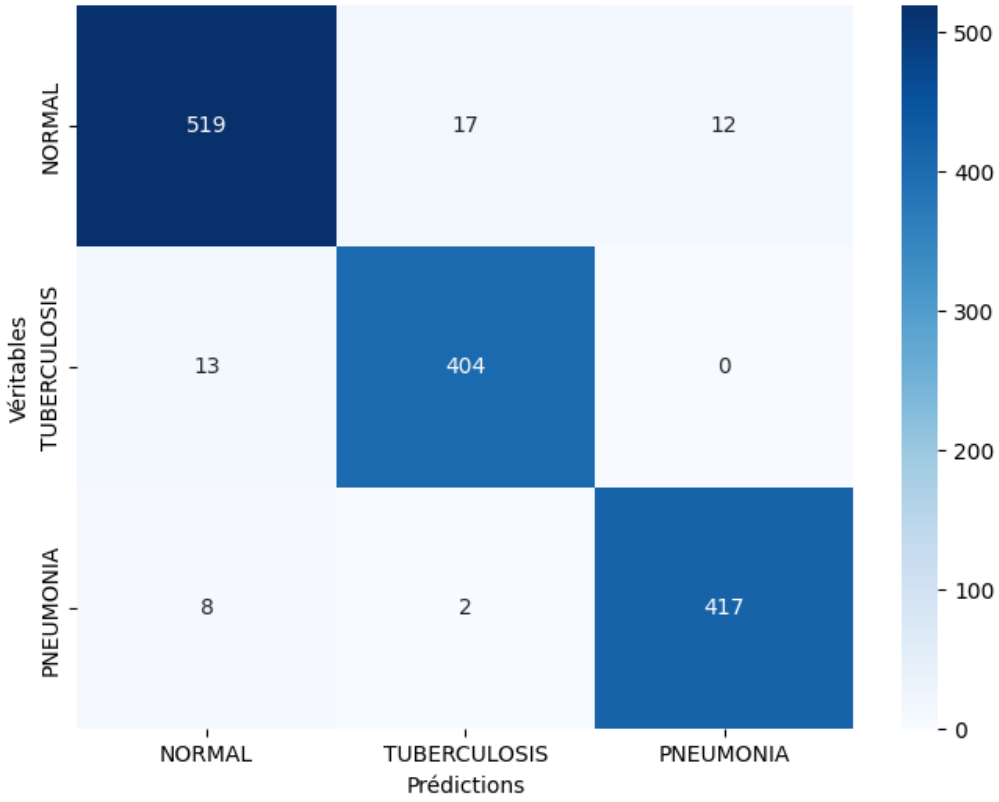


Figure 4.5: Confusion matrix of Random Search on the test set.

- **Genetic Algorithm**

  To combine predictions from the selected models, we employed a GA to determine the optimal weight distribution for a weighted average ensemble. GAs are evolutionary optimization techniques inspired by natural selection, and although they are not widely applied in EL for medical image classification, we chose to investigate their potential due to their adaptability and efficiency in navigating complex search spaces.

  The process begins with the generation of a random population of candidate weight vectors, each representing a possible combination of contributions from the three base models (VGG16, ResNet101, and DenseNet201). For each individual (weight set), we compute the ensemble's prediction and evaluate its accuracy using a fitness function. The

best-performing individuals are selected as parents, and crossover operations are applied to combine their weights into new offspring. To maintain diversity, mutation is introduced by randomly altering one weight in an individual. This evolutionary cycle (selection → crossover → mutation) is repeated over 20 generations, allowing the algorithm to converge toward an optimal solution.

The best result achieved by the GA yielded an ensemble with a validation accuracy of 96.77%. The final learned weights reflect each model's contribution:

- VGG16 had the most influence with 82.7%,
- ResNet101 contributed 12.4%, and
- DenseNet201 contributed 4.9%.

This indicates that VGG16 consistently performed best across folds and had the strongest generalization ability for this task.

Table 4.5: Classification report for Genetic Algorithm Technique.

Classe	**Précision**	**Recall**	**F1-score**	**Effectif**
NORMAL	0.96	0.96	0.96	548
TUBERCULOSIS	0.97	0.97	0.97	417
PNEUMONIA	0.98	0.98	0.98	427
Accuracy		0.97		

As illustrated in Figure 4.6, the confusion matrix provides a clear visualization of the model's classification performance using theGA-based ensemble. The model demonstrates strong predictive capabilities across all three classes: Normal, Tuberculosis, and Pneumonia, with minimal misclassifications.

For the Normal class, 526 out of 548 images were correctly classified, with only 12 and 10 instances misclassified as Tuberculosis and Pneumonia, respectively. This corresponds to a precision and recall of 96%, indicating both high accuracy in prediction and good sensitivity.

In the case of Tuberculosis, the model correctly classified 404 out of 417 instances. Notably, no Tuberculosis cases were misclassified as Pneumonia—an important result given the clinical similarity between these diseases. This yielded 97% precision and recall, highlighting the model's effectiveness in distinguishing Tuberculosis from other thoracic conditions.

The Pneumonia class exhibited the highest performance, with 417 of 427 images correctly predicted. Only 10 instances were misclassified as Normal, and none were confused with Tuberculosis. This led to a precision of 98% and a recall of 98%, reflecting excellent model sensitivity and specificity for Pneumonia detection.

Overall, the GA ensemble achieved an accuracy of 97% on the full multiclass dataset, as shown in Table 4.5. The model's strength lies in its ability to minimize confusion

between clinically overlapping categories. The absence of any misclassification between Tuberculosis and Pneumonia demonstrates the ensemble's robustness and the effectiveness of the GA-optimized weighting in improving predictive performance.

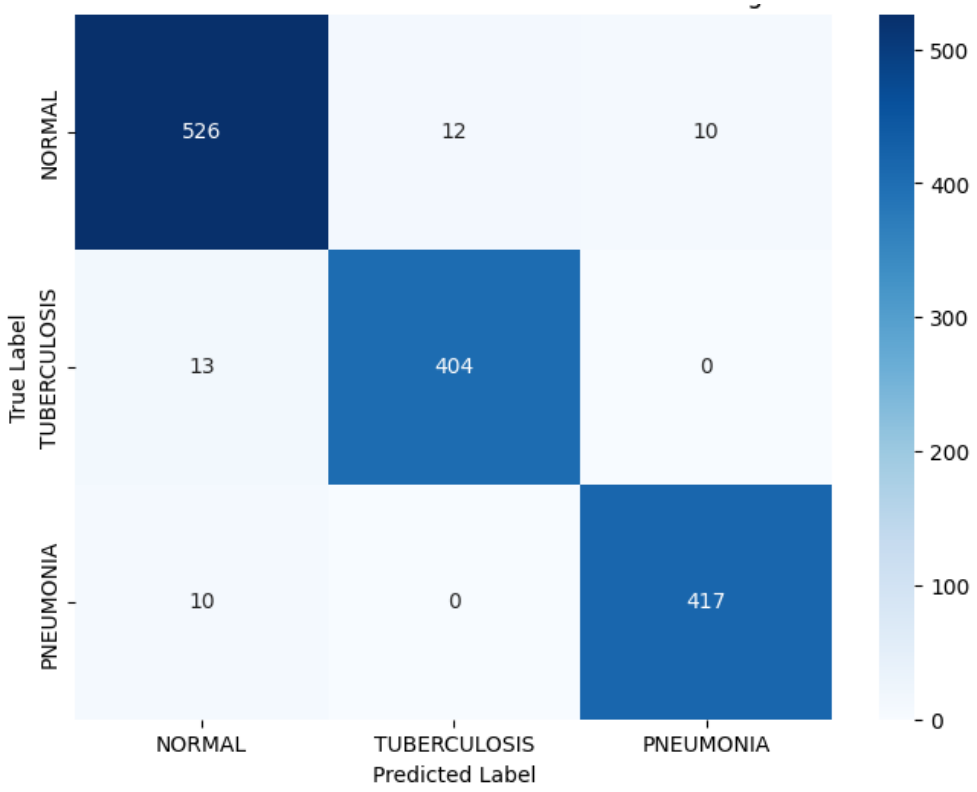


Figure 4.6: Confusion matrix of Genetic Algorithm

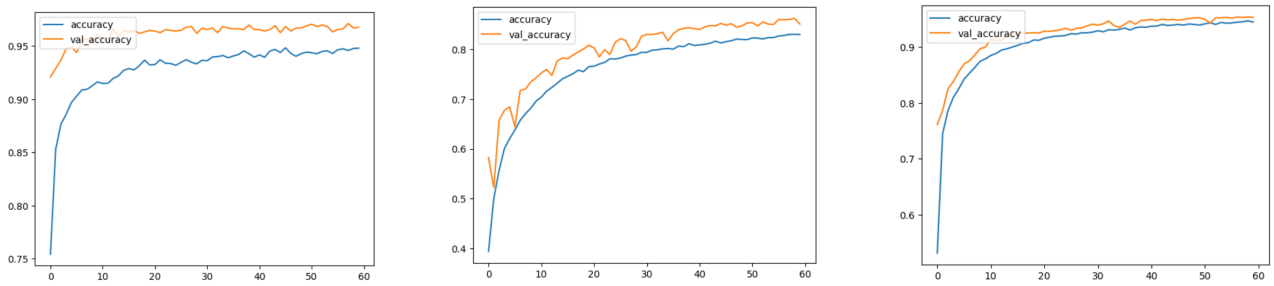
4.2.5 Training and Evaluation Using Optimized Hyperparameters

After optimizing the hyperparameters of the selected pretrained CNN models, we proceeded with training on the multiclass chest X-ray dataset. Each model was trained using its best-performing configuration obtained through hyperparameter optimization, enabling more effective feature extraction and classification. The evaluation process involved tracking training and validation accuracy and loss to assess model performance and generalization.

DenseNet201: The model training starts with accuracy 75% and steadily increase up to 95%, and validation accuracy begins with 92% ans still increase to 96.77%. The loss training starts with 62% and decrease to 15%, and validation loss is decrease from 20% to 8%.

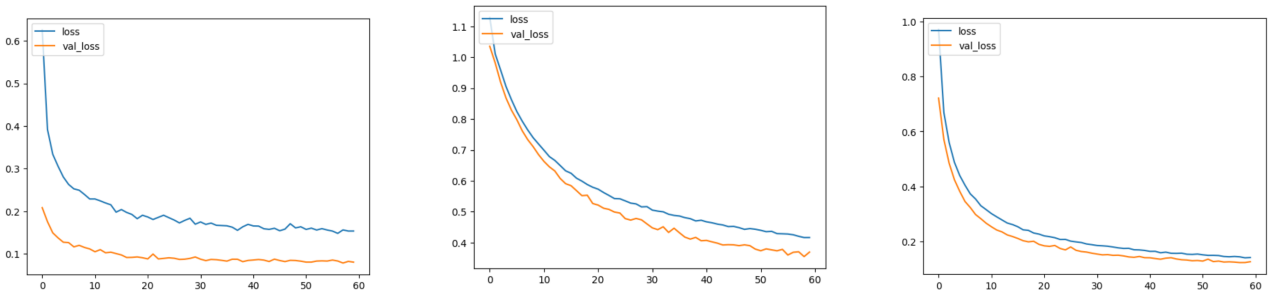
VGG16: The training accuracy of this model increase from 53% to 95%, and validation accuracy starts from 76% to 95.33%. The training loss begins with 95% and still decrease to 14%, The validation loss is decrease from 72% to 12%.

ResNet101: The accuracy of training process increase from 39.34% to 83.04%, and validation accuracy starts with 58.26% and stay increasing to 85.13%. The training loss starts with 100% and decreasing to 41%, validation loss increasing from 100% to 36%.



(a) DenseNet201 (b) ResNet101 (c) VGG16

Figure 4.7: Accuracy plots of the three models.



(a) DenseNet201 (b) ResNet101 (c) VGG16

Figure 4.8: Loss plots of the three models.

The **Classification Report** is a tool used in machine learning to evaluate the performance of a classification model. It displays various metrics such as precision, recall, and F1 score for each class in the classification task [47]:

- **DenseNet201**: The table 4.6 below presents the metrics of performance

Table 4.6: Classification report of DenseNet201 model

Classe	**Précision**	**Recall**	**F1-score**	**Effectif**
NORMAL	0.96	0.96	0.96	548
TUBERCULOSIS	0.97	0.97	0.97	417
PNEUMONIA	0.98	0.98	0.98	427
Accuracy		0.97		

The classification of DeneNet201 model demonstrates excellent perfermance across all three classes: Normal, Tuberculosis, and Pneumonia. it achieved the highest accuracy with 97%, For the precesion and recall metrics and F1-scores the maximun percentage

of predection with 98%, indicating both robustness in classifying the pneumonia cases. Also identify the tuberculosis images correctly with accuracy of 97% and the normal cases with 96%.

- **VGG16**: The table 4.7 below presents the metrics of performance providing the model ability to correctly classify Normal, Tuberculosis, and Pneumonia cases.

Table 4.7: Classification report for VGG16 Model

Classe	**Précision**	**Recall**	**F1-score**	**Effectif**
NORMAL	0.94	0.95	0.94	548
TUBERCULOSIS	0.95	0.97	0.96	417
PNEUMONIA	0.97	0.95	0.96	427
Accuracy		0.95		

The classification of VGG16 model was achieved an accuracy of 95%. The highest precesion presented in pneumonia cases with 97%, then tuberculosis with 95% and normal cases with 94% and that's indicating both excellent predection and robustness identifyin true pneumonia cases that other cases. And for recall, the model can evaluate and predect more the tuberculosis images than normal and pneumonia cases.

- **ResNet101**: After training and testing process of the model we calculate the pencetage of predection correct of cases .

Table 4.8: Classification Report for ResNet101 Model

Class	**Precision**	**Recall**	**F1-score**	**Support**
NORMAL	0.87	0.77	0.82	548
TUBERCULOSIS	0.80	0.87	0.83	417
PNEUMONIA	0.89	0.94	0.91	427
Accuracy		0.85		

The classification of ResNet101 model was evaluated on dataset comprising three categories: Normal, Tuberculosis, and Pneumonia, The overall accuracy of the model achieved 85%. Amoung the threee classes, pneumonia achieved the highest classification performance. with a precession of 89% and a recall of 94% indicating the model's strong ability to both correctly label and detect pneumonia cases. For the normal class, the model atteined a precesion of 87% and a recall of 77% demonstating that predections

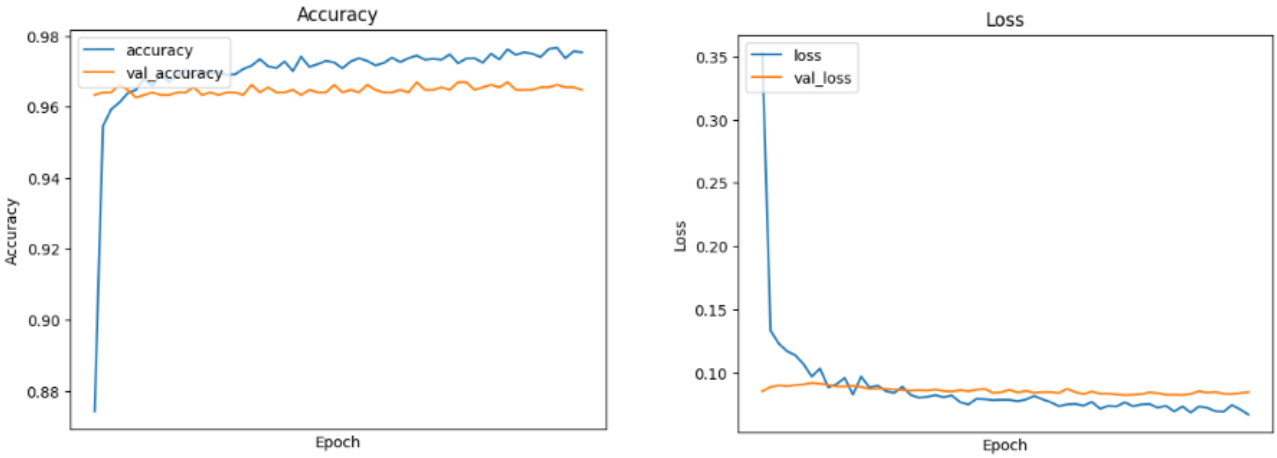
labeled as a normal were generally accurate.

4.2.6 Ensemble Learning Techniques

To leverage the strengths of individual models, we implemented stacking and soft voting ensemble strategies. These methods were employed to enhance model generalization and robustness.

- **Stacking Ensemble on Multiclass Dataset**

  A stacking ensemble was developed using three pre-trained CNN architectures: **DenseNet201**, **VGG16**, and **ResNet101**. Each network's final classification layer was removed to extract 256-dimensional features. These were concatenated into a 768-dimensional vector and passed to a meta-classifier composed of two dense layers and two dropout layers. The final output layer used softmax activation to classify images into Normal, Tuberculosis, and Pneumonia.



(a) Training and validation accuracy.

(b) Training and validation loss.

Figure 4.9: Performance of the stacking model.

The training and validation curves (Figure 4.9a and Figure 4.9b) exhibited smooth convergence with minimal overfitting. The difference between training and validation accuracy remained under 1%, and the loss values decreased steadily, demonstrating stable generalization throughout training.

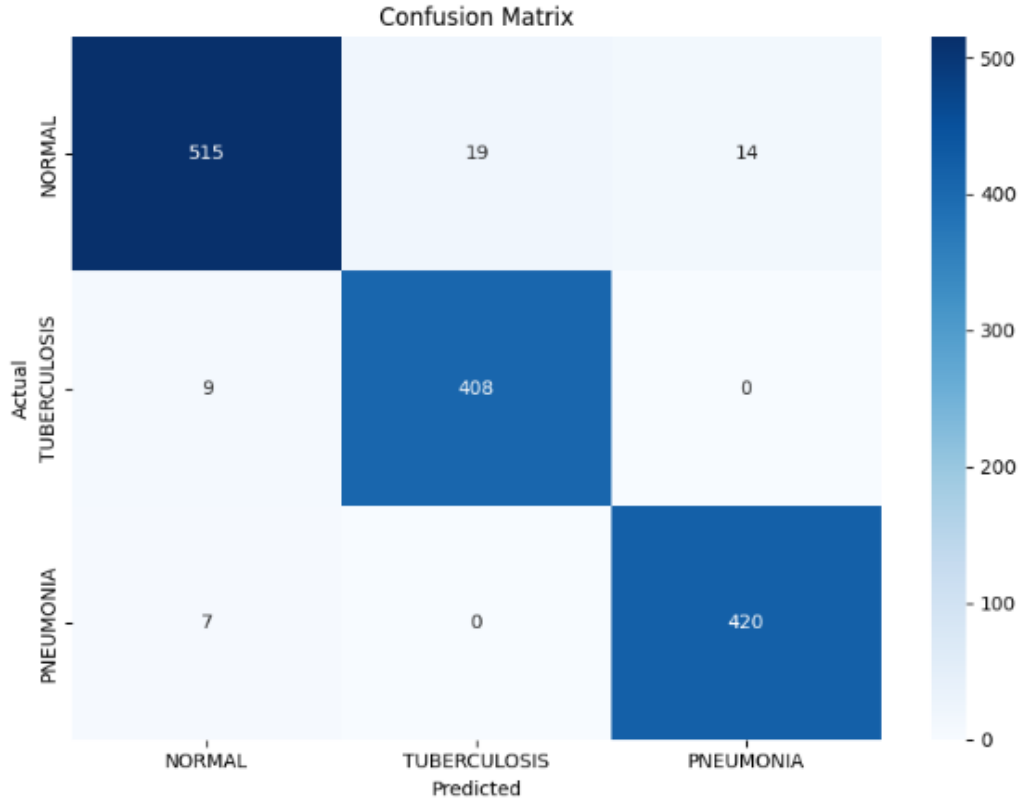


Figure 4.10: Confusion matrix of the stacking model.

The classification report and confusion matrix revealed high accuracy and balanced performance across all three classes. The following table summarizes the per-class precision, recall, and F1-score on the test set (1,392 images):

Table 4.9: Per-class classification metrics of the stacking ensemble.

Class	**Precision**	**Recall**	**F1-score**
Normal	0.97	0.94	0.95
Tuberculosis	0.96	0.98	0.97
Pneumonia	0.97	0.98	0.98

The model demonstrated strong class-wise performance across all categories. For **Normal** cases, a high precision of 0.97 indicates few false positives, while a recall of 0.94 suggests some normal images were misclassified, primarily as tuberculosis or pneumonia. **Tuberculosis** was identified with a recall of 0.98, meaning nearly all true TB cases were correctly detected; its precision of 0.96 reflects a small number of false positives. **Pneumonia** achieved the highest overall metrics, with a precision of 0.97, recall of 0.98, and an F1-score of 0.98, indicating the model's exceptional ability to accurately and consistently classify pneumonia images.

The confusion matrix (Figure 4.10) confirmed a strong separation between classes, with minimal confusion between tuberculosis and pneumonia, suggesting that the stacking ensemble effectively captured distinct pathological patterns. The primary source of misclassification occurred between *Normal* and the other two classes, likely due to subtle thoracic anomalies that visually resemble early-stage disease.

After all the evaluations and detailed analysis of classification metrics, training behavior, and confusion patterns, it becomes evident that the stacking ensemble consistently outperforms each individual model used in isolation. The stacking model achieved a training accuracy of 97.54% and a validation accuracy of 96.48%, with low training and validation losses of 0.0670 and 0.0848, respectively. In comparison, the individual CNNs—DenseNet201, VGG16, and ResNet101—achieved validation accuracies ranging from 95.33% to 97.61% on the Pneumonia-Tuberculosis subset. While these standalone models performed well, each showed slight variability depending on the dataset. The stacking strategy effectively integrates their feature representations, leveraging their complementary strengths to deliver a more robust and generalized classifier. This ensemble approach ensures balanced performance across all classes, especially in challenging multiclass scenarios. Hence, stacking emerges as a superior choice over individual models, offering both improved accuracy and stronger generalization.

- **Soft Voting Ensemble (with Genetic Algorithm Weights)**

  To improve classification performance while maintaining a lower computational cost compared to stacking, a soft voting ensemble was applied using the predictions of three CNN models: VGG16, ResNet101, and DenseNet201. Each model produced a probability distribution over the three target classes: Normal, Tuberculosis, and Pneumonia.

  Instead of using equal weights for soft voting, we employed a GA to optimize the contribution of each model. The GA searched for the most effective combination of weights that maximized classification accuracy on the validation set. The final optimized weights determined the influence of each model in the ensemble as follows:

  VGG16: 82.7%

  ResNet101: 12.4%

  DenseNet201: 4.9%

  The ensemble prediction was computed as a weighted average of the output probabilities from each model using these GA-derived weights. The final class label was assigned based on the class with the highest combined confidence score.
  This GA-optimized ensemble achieved an overall accuracy of 96.77% on the validation set, demonstrating improved robustness and better generalization compared to relying on any single model. An example of this weighted soft voting process is illustrated in table 4.10, showing predictions for five test samples where the true class is Normal. The predicted class probabilities reflect the contribution of each model weighted according to the GA-optimized weights (VGG16: 82.7%, ResNet101: 12.4%, DenseNet201: 4.9%).

Table 4.10: Sample predictions using GA-weighted soft voting ensemble.

Sample	**True Label**	**Predicted Probabilities (0 / 1 / 2)**
1	0	[0.626, 0.373, 0.0002]
2	0	[0.945, 0.053, 0.0018]
3	0	[0.973, 0.027, 0.00009]
4	0	[0.935, 0.064, 0.00036]
5	0	[0.580, 0.417, 0.0022]

The previous table 4.10 present some examples for the technique of softvoting using GA weights, The true class for all samples is *Normal* (class 0), and the probabilities correspond to the classes: Normal (0), Tuberculosis (1), and Pneumonia (2).
The predicted labels for all five samples matched the true labels, demonstrating the ensemble's confidence in aggregating model outputs effectively.

The soft voting model demonstrated strong and well-balanced performance across all classes. It accurately predicted **Normal** cases with high precision and recall, while also showing robust detection of both **Tuberculosis** and **Pneumonia** cases, similar to the stacking model. The ensemble's use of averaged probability distributions contributed to reduced prediction variance and improved confidence, especially in borderline cases.

The confusion matrix (Figure 4.11) confirmed these findings, showing minimal misclassifications and strong class separation, particularly between tuberculosis and pneumonia. The model correctly predicted 524 Normal, 398 Tuberculosis, and 414 Pneumonia cases, with only 24, 19, and 13 misclassifications respectively. While all classes were well predicted, Tuberculosis showed slightly more confusion, mainly being predicted as Normal. With a validation accuracy of 96.77%, the model exhibited excellent generalization.

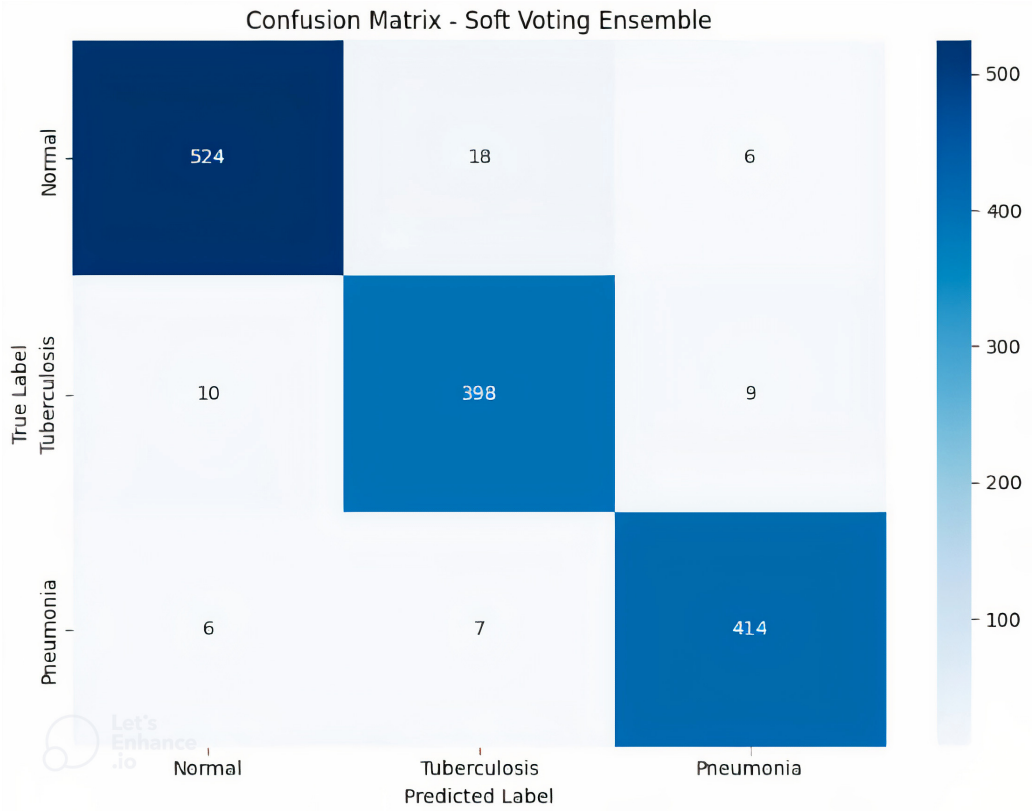


Figure 4.11: Confusion matrix of the soft voting ensemble.

These results confirm that the GA-weighted soft voting strategy not only improved accuracy but also enhanced the model's ability to confidently distinguish between challenging cases. The method's reliance on probabilistic averaging made it highly effective in reducing uncertainty and increasing prediction reliability, making it a valuable tool for real-world medical diagnostics.

4.3 Conclusion

This chapter explored the effectiveness of DL techniques for chest X-ray disease classification. We presented a complete workflow, starting with data preparation, followed by HPO using Random Search and GAs. The optimized models were then trained and evaluated individually, and finally combined using EL methods such as soft voting and stacking. By combining multiple finetuned CNN models, we demonstrated improved classification accuracy and robustness. The next chapter summarizes our key findings, discusses the limitations of our approach, and outlines possible directions for future research.

5

Conclusion and future perspectives

Contents

5.1 General conclusion

In this thesis, we investigated a DL-based framework for chest X-ray disease classification, aiming to enhance diagnostic accuracy and robustness in medical image analysis. We utilized a balanced and diverse multiclass dataset comprising Normal, Pneumonia, and Tuberculosis cases. We selected and fine-tuned three powerful pretrained CNN models(DenseNet201, ResNet101, and VGG16) training them end-to-end on the target data. Their predictions were then combined using EL techniques, including hard voting, soft voting, and stacking, to leverage their individual strengths.

To further improve performance, we applied hyperparameter optimization methods such as Random Search and Genetic Algorithms. Throughout the study, we evaluated multiclass classification tasks using key metrics like accuracy, precision, recall, F1-score, and confusion matrices. The results demonstrate that combining DL with ensemble methods and optimization techniques leads to a more accurate, stable, and reliable system for automated chest disease diagnosis. Among the ensemble approaches, the best results were achieved using soft voting with an accuracy of 96.77%, followed closely by stacking with 96.48%.

5.2 Limitations and Future Perspective

Although the proposed framework achieved high accuracy and demonstrated robustness in classifying chest X-ray images, some limitations remain. First, while a consistent preprocessing pipeline was applied, the use of more advanced techniques tailored to medical imaging such as contrast enhancement, noise reduction, or intensity normalization could further improve image quality and model robustness. Additionally, the study was limited to three CNN architectures (VGG16, ResNet101, and DenseNet201); exploring more recent or lightweight models such as

EfficientNet, MobileNet, or transformer-based networks could offer a better balance between accuracy and computational efficiency. Moreover, hyperparameter optimization (HPO) was performed only on the base models using Random Search and Genetic Algorithms. Applying HPO to the ensemble learning (EL) components—such as the weight assignment in soft voting or meta-learner configuration in stacking—could unlock additional performance gains. Future work may also consider using more advanced HPO techniques, such as Bayesian optimization or evolutionary strategies, to improve tuning efficiency. Beyond the ensemble techniques already explored, investigating alternatives like bagging, boosting, or hybrid ensembles may lead to more stable and accurate predictions. Finally, although ensemble methods improved performance, their higher computational cost and longer training time could be a constraint in real-time or resource-limited clinical environments, warranting further exploration into lightweight deployment strategies.

Bibliography

[1] upGrad. Basic cnn architecture: Explaining 5 layers of convolutional neural network. Blog post, June 2025. Accessed March 10, 2025.

[2] GeeksforGeeks. What is transfer learning?, 2021. Accessed: April 16, 2025.

[3] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. "densely connected convolutional networks". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269. IEEE, July 2017.

[4] Imdad Ali, Muhammad Muzammil, Ihsan Ul Haq, Amir A. Khaliq, and Suheel Abdullah. Deep feature selection and decision level fusion for lungs nodule classification. *IEEE Access*, 9:18962–18973, 2021.

[5] Rohit Modi. Resnet — understand and implement from scratch. Medium, 2020. Disponible à: https://medium.com/analytics-vidhya/resnet-understand-and-implement-from-scratch-d0eb9725e0db (Accessed June 15, 2025).

[6] GeeksforGeeks. What is fine-tuning in deep learning?, 2023. Accessed: June 16, 2025.

[7] Viso.ai contributors. Ensemble learning. Viso.ai. Available at https://viso.ai/deep-learning/ensemble-learning/ (Accessed April 15, 2025).

[8] Binary Terms. Genetic algorithm in data mining: Working, example, applications. https://binaryterms.com/genetic-algorithm-in-data-mining.html, 2025. Accessed June 13, 2025.

[9] Sethanan K. et al. "a three-stage ensemble boosted convolutional neural network for classification and analysis of covid-19 chest x-ray images". *Frontiers in Medicine*, 9:861680, 2022.

[10] Ruaa N. Sadoon and A. Chaid. "classification of pulmonary diseases using a deep learning stacking ensemble model". *Informatica*, 48(14), 2024.

[11] Nazik Alturki, Muhammad Umer, Abid Ishaq, Nihal Abuzinadah, Khaled Alnowaiser, and Abdullah Mohamed. "combining cnn features with voting classifiers for optimizing performance of brain tumor classification". *Cancers*, 15(6):1767, 2023.

[12] Joseph Bamidele Awotunde, Agbotiname Lucky Imoize, Oluwafisayo Babatope Ayoade, Moses Kazeem Abiodun, Dinh-Thuan Do, Adão Silva, and Samarendra Nath Sur. "an enhanced hyper-parameter optimization of a convolutional neural network model for leukemia cancer diagnosis in a smart healthcare system". *Sensors*, 22(24):9689, 2022.

[13] Abdullah A Asiri, Ahmad Shaf, Tariq Ali, Muhammad Aamir, Muhammad Irfan, and Saeed Alqahtani. "enhancing brain tumor diagnosis: an optimized cnn hyperparameter model for improved accuracy and reliability". *PeerJ Computer Science*, 10:e1878, 2024.

[14] Alejandro Morales-Hernández, Inneke Van Nieuwenhuyse, and Sebastian Rojas Gonzalez. A survey on multi-objective hyperparameter optimization algorithms for machine learning. *Artificial Intelligence Review*, 56:8043–8093, 2023.

[15] Daniyal Asif, Mairaj Bibi, Muhammad Shoaib Arif, and Aiman Mukheimer. "enhancing heart disease prediction through ensemble learning techniques with hyperparameter optimization". *Algorithms*, 16(6):308, 2023.

[16] Mahmoud Smaida and Serhii Yaroshchak. "bagging of convolutional neural networks for diagnostic of eye diseases". In *International Conference on Computational Linguistics and Intelligent Systems*, 2020. Creative Commons License Attribution 4.0 International (CC BY 4.0).

[17] Mamar Khaleda, Djamel Gaceb, Fayçal Touazia, Chakib Ammar Aouchiche, Youcef Bellouche, and Ayoub Titoun. "new cnn stacking model for classification of medical imaging modalities and anatomical organs on medical images". In *CEUR Workshop Proceedings*, volume 3609, 2023. Accessed June 13, 2025.

[18] Emrana Kabir Hashi and Md. Shahid Uz Zaman. "developing a hyperparameter tuning based machine learning approach of heart disease prediction". *Journal of Applied Science & Process Engineering*, 7(2):631–647, 2020.

[19] L. Fdez-Díaz, J. R. Quevedo, and E. Montañés. "regularized boosting with an increasing coefficient magnitude stop criterion as meta-learner in hyperparameter optimization stacking ensemble". *Neurocomputing*, 551:126516, 2023.

[20] Rifatul Islam Majumder. Pneumonia & tuberculosis with normal & non-x-ray [dataset]. medRxiv, 2025. Disponible à: https://www.kaggle.com/datasets/rifatulmajumder23/combined-unknown-pneumonia-and-tuberculosis/code.

[21] U.S. Food and Drug Administration. Medical imaging, 2024. Accessed: 2025-06-21.

[22] Imperial Imaging Technology. Medical imaging systems explained: Different types and purposes. https://www.imperialimaging.com/imaging-technology-explained-different-types-and-purposes/, Jul 2024. Accessed June 01, 2025.

[23] Health care: Definition & overview. https://study.com/academy/lesson/health-care-definition-overview.html.

[24] Pragati Patel and Shivani Gajjar. "a review: Comparative analysis of artificial intelligence, machine learning (ml), and data science". *International Journal of Research Publication and Reviews*, 4(10):2451–2456, 2023. ISSN 2582-7421.

[25] Koffka Khan and Wayne Goodridge. "artificial intelligence (ai) versus machine learning (ml) versus deep learning (dl)". *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, 5(3):117–123, 2022.

[26] IBM. Deep learning. https://www.ibm.com/think/topics/deep-learning, 2025. Accessed March 15,2025.

[27] Datacamp. Introduction to convolutional neural networks (cnns). https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns, 2022. Accessed: June 26, 2025.

[28] Saida Sarra Boudouh and Mustapha Bouakkaz. "breast cancer: Breast tumor detection using deep transfer learning techniques in mammogram images". 2022.

[29] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, 2017.

[30] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks – densenet. Papers With Code, Sep 2016. Available at https://paperswithcode.com/method/densenet (accessed 14 Jun 2025).

[31] GeeksforGeeks contributors. Densenet explained. GeeksforGeeks, 2024. Available at https://www.geeksforgeeks.org/computer-vision/densenet-explained/ (accessed 14 Jun 2025).

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[33] Siddhesh Bangar. Vgg-net architecture explained. Medium blog post, June 2022. Accessed June 14, 2025.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[35] Wenchao Gu, Shuang Bai, and Lingxing Kong. A review on 2d instance segmentation based on deep neural networks. *Image and Vision Computing*, 120:104401, 2022.

[36] Farshid Hajati and Mohammad Ali Moni. "ensemble learning for disease prediction: A review". *Healthcare*, 11(12):1808, 2023. Open Access under CC BY 4.0.

[37] Hela Elmannai, Hager Saleh, Abeer D. Algarni, Ibrahim Mashal, Kyung Sup Kwak, Shaker El-Sappagh, and Sherif Mostafa. "diagnosis myocardial infarction based on stacking ensemble of convolutional neural network". *Electronics*, 11(23):3976, 2022.

[38] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[39] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[40] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. CRC press, 2012.

[41] David H Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

[42] Petro Liashchynskyi. "grid search, random search, genetic algorithm: A big comparison for nas". 2019.

[43] Morales-Hernández Alejandro, Van Nieuwenhuyse Inneke, and Rojas Gonzalez Sebastian. "a survey on multi-objective hyperparameter optimization algorithms for machine learning". *arXiv preprint arXiv:2111.13755*, 2022.

[44] Mohaimenul Azam Khan Raiaan, Sadman Sakib, Nur Mohammad Fahad, Abdullah Al Mamun, Md. Anisur Rahman, Swakkhar Shatabda, and Md. Saddam Hossain Mukta. "a systematic review of hyperparameter optimization techniques in convolutional neural networks". *Decision Analytics Journal*, 11:100470, June 2024.

[45] TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from https://www.tensorflow.org/, 2015. Accessed: May 14, 2025.

[46] François Chollet et al. Keras. https://github.com/keras-team/keras, 2015. Accessed June 14, 2025.

[47] Usama Bajwa. Classification report. Kaggle Notebook, 2023. Retrieved from Kaggle.