LIGHTWEIGHT DEEP LEARNING MODELS FOR ACCURATE CHEST X-RAY DISEASE CLASSIFICATION

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

MASTER OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Paramhans Mishra

(24/CSE/06)

Under the Supervision of

Prof. Shailender Kumar

To The

THE DEPTPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

MAY 26

THE DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, **Paramhans Mishra** (Roll No. **24/CSE/06**), hereby certify that the work which is being presented in the thesis entitled **"Lightweight Deep Learning Models for Accurate Chest X-Ray Disease Classification"** is submitted by myself in partial fulfillment of the requirements for the award of the degree of Master of Technology, submitted in the Department of Computer Science & Engineering, Delhi Technological University is an authentic record of my own work carried out by myself under supervision of Prof. Shailender Kumar. The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Place: _____

Date: _____

Paramhans Mishra
(**24/CSE/06**)

This is to certify that the candidate has included all the corrections suggested to him in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Signature of Supervisor **Signature of External Examiner**

THE DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE FROM SUPERVISOR

Certified that **Paramhans Mishra**(2K24/CSE/06) has carried out their search work presented in this thesis entitled **"Lightweight Deep Learning Models for Accurate Chest X-Ray Disease Classification"** for the award of **Master of Technology** from Department of COMPUTER SCIENCE & ENGINEERING, Delhi Technological University, Delhi, under my supervision. The thesis embodies results of original work, and studies are carried out by the student himself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Place: _____

Date: _____

Prof. Shailender Kumar
SUPERVISOR
(Professor, CSE, DTU)

ABSTRACT

This Thesis try to create a DL frameworke for automatic two class classification of chest Xray images (CXRI). These classe is Normal and Pneumonia. As per WHO, Pneumonia continues to be a prominent respiratory illness globally [9]. Due to radiologists' technical limits and workload, it has been observed with considerable delays in diagnosis for vulnerable populations, including children, the elderly, and immunocompromised patients, as well as the great majority of the general population. Automatic screening devices can help physicians with their work and ease some of their workload.

I introduce a pipeline that enables reproducibly building and comparing a multitude of model families through its unified access to a single data pipeline, evaluation engine and logging standard. It utilizes the following architectures: (i) a custom Convolutional Neural Network (CNN), trained from scratch; (ii) MobileNetV2, chosen as a candidate architecture for lightweight deployment; (iii) EfficientNetB0, offering an acceptable computational cost in the context of transferlearning and (iv) ResNet50, used as a higher capacity residual network. The methodology starts from the data, following a structured pre-processing protocol including: file integrity checking prior to pixel extraction, stratified training/validation/testing sets to maintain initial proportions between classes, pixel value normalization and finally training-time augmentation to mitigate impacts from variability in acquisition parameters.

I include random flipping, rotation, zoom and variation of brightness and contrast at the time of training for all images to better generalise to data of varying acquired quality. Sample weighted loss was chosen instead of class duplication for the case of imbalance to prevent unnatural upscaling that could lead to overfit on certain examples of overrepresented class and also to train the model to adhere to natural distribution of the input data during classification. Evaluation metrics vary beyond Accuracy and consider threshold sensitivity.

Altho, MOBILE Net V2 with less parameters outperformed scratchtrained network, it also maintained performance and is most desirable for deployment-constrained use cases. Interpretability of learned features in all four models was also investigated using GradCAM visibilization which indicated that models learnt meaningful pulmonary features in a relevant anatomical region. The entire pipeline can be extended to new architectures, augmentation procedures, loss functions, and evaluation cirteria..

ACKNOWLEDGEMENTS

Completion of any task is impossible to achieve without acknowledgement of the help that provided the support and direction required for its success. Without the aid of these individuals and organizations this project would have never become a reality. Firstly, we wish to thank **Prof. Anil Singh Parihar** (Head, Department of CSE, DTU) and the faculty of our own CSE department at DTU for their continued support, motivation and guidance throughout this project. In particular, we also wish to thank **Prof. Shailender Kumar** (Department of CSE, DTU), our faculty for CSE, for providing us with continuous encouragement and valuable advice on all aspects of this M.Tech Project. He has provided much needed direction and was very instrumental in developing both the concept and implementation of this project. We are grateful to him for being available and ready to provide guidance at each stage of this project and for helping us overcome obstacles that we encountered along the way. I would like to acknowledge all researchers and open-source contributors whose datasets, libraries, and scientific publications made this study possible. The publicly available chest X-ray resources and the broader machine learning research community have provided a strong foundation for experimentation and comparative analysis. Their contributions have significantly accelerated meaningful progress in healthcare AI research. Finally, I offer my gratitude to everyone who directly or indirectly contributed to this thesis. This work is a cumulative outcome of mentorship, collaboration, and persistent learning, and I remain sincerely thankful for every form of support received.

Paramhans Mishra
(**24/CSE/06**)

LIST OF FIGURES

CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Purpose

Project's primary goal is to determine that can the use of current time deep learning methods can provide assistance to health care professionals when it comes to identifying pneumonia based on the images taken of a patients chest. Radiologic diagnosis continues to remain at the top of the list of the most important and challenging tasks performed by health care professionals today due to the fact that health care providers are required to make timely clinical decisions with limited amounts of data and while working with large numbers of patients. Therefore, a machine-based system capable of consistently detecting visual patterns related to pneumonia would have significant practical application as a decision-support tool designed to enhance both clinical workflow efficiency and clinical decision-making confidence [9, 1, 2]..

The goal of this project will be achieved through the design and comparison of four highly efficient model architectures which utilize a single framework: self-built CNN , MobileNetV2 , EfficientNetB0, and ResNet50. The codebase was developed using an intentional modularity design so that each component could be separated into distinct modules for loading datasets, pre-processing images, training models, evaluating performance, visualizing results, and explaining results. By developing not just a one-time classifier, but also a modular research platform, this project enables a researcher to train new models, audit existing models, extend previous models, and/or expand upon past research initiatives in subsequent semesters or research cycles [4, 5, 3]..

A second major purpose of this project is to assess whether low-resource and knowledge-transfer methods are able to produce clinically relevant pneumonia-screening behaviors within resource-constrained conditions such as class imbalance, reduced diversity in annotated examples, and limited computation. The project therefore places emphasis on making methodological choices that increase reliability in practice by employing techniques such as class weighting, image-augmentation, early-stopping, adaptive-scheduling of learning rates during training, test-time augmentation, and tuning decision-thresholds with respect to validation metrics. In other words, the ultimate goal of the project is not simply to maximize test accuracy for its own sake; rather the objective is to develop a well-balanced diagnostic support-pipeline that is computationally-efficient, provides transparency in addition to interpretability.

The research aims to find answers to questions regarding whether the accuracy of models to correctly classify chest radiographic images, unlike more trivial datasets of

natural scenes, which involves significant challenges due to inherent difficulties in interpreting the complex anatomical structure of the chest and processing subtle differences in textual patterns and contrast, will be improved by adjusting different architecture designs, preprocessing pipelines, and training methodologies. The study further looks into exploring the capability of these models to demonstrate their diagnostic plausibility. [4]

1.2 Problem Description

Pneumonia disproportionately affects fragile Populations of young children, elderly, and immunocompromised Populations, and can Quickly progress to lifesaving Complications, if Unidentified [9] In many health systems chest X rays (cxrs) remain the initial modality of imaging, Given that they are Cheap,fast and widely available; Yet,reliablecxrinterpretation requires specialty Expertise,which Is Unevenly distributed Throughout health systems [6] Gaps in radiographic coverage Are Greatest in countries with low-and middleincome,where,unfortunately,the Most Number preventable deaths from respiratory diseasesoccur .For the Purpose Of the Current project the task Will Be Defined as a twoclass visual Classification task on cxrs, With the Objective to Develop a model that reliably distinguishes between NORMAL and pneumonia Classes by use of heterogenous sources of noosey image Data The ingestion portion of the SOURCE code is far From being trivial (i.e., Check of the file's data Integrity,removal of non-image features, Application Of augmentation techniques for increasing the robustness, And stratum Split to preserve class distributions) Will Be necessary In This project,we Will take Threshold Selection as a basic problem component. Although,the model May perform well at the standard threshold (value 0.5,say),it's not clear such that the model Will Be acceptable When Consideration is given to tradeoffs Between sensitivity for detecting pneumonia and specificity for rulingout Pneumonia,by Use of macro F1 on a Validation Set we Will Determine thresholds for each classesand we will Evaluate by Using test-time Accuracy(TTA),We realize That a testtime Accuracy of a classification Task Depends on proper Calibration Decisions: hence, It is also required to develop a decision support system (decision making). High workload of chest xrays.

However this should Be done takingInto Account many different variability that affect the data(Image acquisition parameters,differences inpatient anatomy,dataset composition biases); It is Desirable also that the System should generalise well. The hope Is that This System can help reducing clinical uncertainty, supporting triage And serving as an evidenceBased decision support For pneumonia Screening In Resource limited and time strained settings. High worldwide prevalence of pneumonia, and a poor geographic Distribution Of radiologists' expertise Means That regions That bear disproportionate rates of preventable respiratory deaths usually Are characterizedby low Numbers Of available radiologists, as Well as inefficient systems of imaging interpretation.

Lastly,in practise it's difficultto choose thresholds And calibrate our models. This screening model will lead to missed cases ofpneumoniaor overused hospital/clinical Resources If too much sensitivity or specificity are used. thereforeThe result should alwaysBe evaluated in probability terms with respects to a specific operational objective,

Not a binary decision. Consequently, This projects Frames the problem not merely As aclassificationtask but rather AS making decision in context Basedon probabilitiesandcalibration (dss).

1.3 Need for the Study

The need for this particular project arises because of the critical urgency of the epidemiology of pneumonia. They also arise because of the problems of healthcare delivery. Further, the individual motivations are also characterized by the various recent advances in the development of and their applications for machine learning approaches, which this project leverages for the enhanced development of assistive approaches to pneumonia diagnostics with both speed and safety.

There are additional reasons this study should be performed. This is because many recent research in academic or earlyconcept implementations reports very impressive headline performance metrics without addressing the need to develop a complete reproducible pipeline for the use of these algorithms, including aspects such as aggressive but robust preprocessing, meticulous train/validation split, training callback selection, threshold strategies, and model interpretability techniques. In addition, this project closes this gap by developing a complete, reproducible commandline driven pipeline that whencodifies these concepts through versioncontrolled configuration. As such, the project not only demonstrates feasibility, but provides baselines to systematically benchmark and extend prior research [6, 8].

Other motivations include the fact that multiple architectures of efficient models are evaluated in what can be more accurately described as a controlled comparativestudy than simply reporting headline performance improvements across experiments due to the variability of the training environment. The lack of crossmodel comparisons made possible by lack of common setup makes it harder to choose what architecture is best for a specific application. Due to this, using a common image size, a shared data pipeline, similar evaluation metrics, and common logging format is likely to provide a more reliable basis by which to compare models than simply comparing overall performance numbers, and then that may be helpful when deciding which model is best suiting our specific medical screening needs with limited resources.

Another reason this study is important is that the literature on automated pneumonia diagnosis has a number of weak points. Many of the studies provide "headline" impressive results but do not address issues of reproducibility (transportability), interpretability, or thorough validation studies, such as using only one architecture to make a comparison or failing to do enough tonton validation. Other studies that do examine multiple architectures still have small sample sizes, different training and testing environments, and overlook issues of dataset bias or generalizability. This study aims to address these issues by focusing on more architecture comparisons, transparent and reproducible image preprocessing, comparable metrics between experiments, and an understanding of the interpretability of the algorithm. In this way, the study aims to bolster this literature base and be of service to future deep learning work in this area.

This project was motivated by the ubiquitous use of Chest Radiography (Chest X Ray) as the most common diagnostic imaging technique accessible which can be

used across healthcare systems even in resourceconstrained environments. Comparing the ability to make sense of Chest X Ray data is therefore likely to have a broader application than solutions requiring expensive modalities. We therefore focus on Chest X Ray data because it is readily available in the current clinical environment with the most direct pathway for future clinical translation.

Finally, the project is driven by the process concerns of clinicians in the time of increasing volume of images produced daily. In timecritical triaging situations, machineenabled prioritization for clinician triaging can speed up initial case reviews so that potential abnormal cases receive additional clinical attention sooner. Clinicians clearly retain the need to interpret images based on their human judgment, but in a supporting role, pilot systems supported by additional computational analysis can be used in so called secondopinion or trilayer roles or in education/training courses and this information will be important to consider as one moves along the path from conceptual pilot studies to clinicallyintegrated algorithms. This project both demonstrates its research goal with results derived directly from clinical settings but also provides support from a practical perspective in including additional information describing the approach of using such models for application in clinical environment.

1.4 Objectives

The objective of this researche project are to develop a both a functionally competent computerized detections system and a research methodology that is methodically transparent;. In order to achieve the first objective, multiple deep learning pipeline architectures were developed to provide binary classifications of chest x-rays as either normal or having pneumonia. Instead of focusing solely on one type of neural architecture (i.e., CNN) the project examined several representative model families.. There were model families (CNN model family as a control) with three other model families using the Transfer Learning capability of ResNet, MobileNet and EfficientNet. I did this to ensure my conclusions were derived from comparative analysis of multiple models' behaviors rather than an individual model's behavior.

In terms of the process of implementing, there were objectives of developing a consistent, repeatable and reliable training protocol for each model. It was necessary to establish a reliable single-stage baseline for both the custom CNN and MobileNetV2 branches. In addition, it was necessary to create and validate two-stage transfer learning protocols including a warm-up phase followed by selective un-freezing of the backbone layers for both the EfficientNetB0 and ResNet50 branches. Using this methodology allows comparison not only between architectures but also between methodologies, thus allowing for deeper insight into the sources of improved performance.

There were also objectives to produce results that are actionable (decision-oriented) versus simply descriptive. As such, generate confusion matrices; compute pneumonia-class precision and recall along with normal-class specificity; record macro-F1; and analyze threshold sweeps to expose recall/specificity tradeoffs. Since this is what will ultimately be used to realistically interpret how well a model is suited for use in screening/ triage settings, these are the targeted outcomes.

The second objective is to maximize model performance while maximizing clin-

ical relevancy in evaluation. Accuracy is sufficient in medical screening applications because false negatives and false positives can have significantly different practical consequences. As such, the objective includes systematic tracking of sensitivity, specificity, precision, recall, F1-Score and AUC-ROC so that the ultimate assessment of usability is reflective of diagnostic utility not just overall correct classification. Methods to achieve better generalization and to decrease overfitting, i.e., hyperparameter tuning, regularization, data augmentation and learning rate scheduling are being employed.

Objective Three: Develop a Meaningful Analysis Framework Based on Model Predictions. As trust and interpretation are key factors influencing the adoption of models in healthcare, this study will provide a connection between output-based decision-making processes and salient features of images utilizing techniques such as Grad-CAM and its visualizations. The purpose of developing an explaining framework is to demonstrate that the areas of an image identified by the models as relevant (via highlighting) correspond to plausible radiologic findings associated with pneumonia. Interpretation serves as a crucial link between computer-generated output and clinical thought.

Objective Four: Establish Reproductivity I& Academic Rigor via Complete Experimental Documentation. This includes providing detail about how datasets were split, pre-processing steps taken, training parameters used, computing environment, and testing methodology. All four objectives combine to allow the project to function as both an applied AI Solution and a Structured Research Contribution to Medical Image Analysis.

1.5 Scope

By focusing on a specific set of problems related to binary classification using chest x-rays images as inputs, it was possible to establish a defined and manageable scope for this project. Additionally, by limiting the number of labels per image (i.e., each sample is classified as either normal or pneumonia-positive) there is no need to consider the challenges associated with multi-label or multi-disease inference at this point. This approach enables an intense focus on understanding how models behave when making one clinically-relevant decision versus multiple, potentially complex decisions.

In terms of what has been included in the scope of this project, there are four model paths along with their corresponding outputs (training checkpoints, final saved models, metrics logs, training curves, confusion matrices and Grad-CAM visualizations). There is also a comparison module that plots the relationship between model performance and model size as well as produces cross-metric charts. Overall, these outputs demonstrate that the scope of this project goes far beyond simply providing trained models (as indicated above), and encompasses all aspects of managing an experiment from start-to-finish, including interpreting results and producing reports.

In terms of modeling, the project includes both custom-developed models and models based upon pre-trained models (transfer-learning); allowing for an exploration of the representation capabilities of each model type as well as their respective computational efficiencies. Additionally, all experiments were conducted within a research environment utilizing popular deep learning libraries. Further, the experiment configurations were managed under reproducibility constraints. The evaluation of models

utilized standard confusion matrix analysis as well as threshold-sensitive metrics (e.g., precision/recall). Also, feature attention was evaluated through Grad-CAM visualization. Therefore, the scope of the project extends beyond solely numerical measures of performance and into areas of interpretability and practical reliability.

Additionally, there are several other limitations on the scope of the project based on operational issues. There are limits to dataset sizes and quality based on publically available data sources, limits on computation power and thus limits on experimental range and breadth; and time-based limits (project duration) limit the amount of full retrain cycles that can occur. These operational constraints improve clarity regarding what can be interpreted from results and allow readers to place results in an honest and science-grounded way.

Finally, the project has been designed to operate within reasonable training infrastructures. A tradeoff was made concerning model complexity vs. runtime feasibility. Large-scale hyperparameter sweeps, federated training across multiple sites, and comprehensive uncertainty quantification are among the excluded options due to resource and time constraints.As with many studies, however, the selected scope of this study is large enough to allow for comparisons among the architectures and some evidence-based conclusions about their performance.

1.6 Expected Findings

The anticipated findings of this project can be identified at three distinct levels (technical, analytical, and research), similar to prior work. From a technical perspective, the project is anticipated to deliver a number of trained and validated deep learning models for detecting pneumonia along with documented reproducible training parameters and serializable model artifacts for use during inference. In addition to providing valid final checkpoint models, the project is expected to deliver a complete pipeline that contains all necessary components for data ingestion; data transformation; data augmentation; model training; computation of evaluation metrics; and generation of interpretability metrics. Delivery of the full pipeline is significant since it allows for repeated execution of experiments and facilitates ongoing comparison.

As noted previously, one additional anticipated finding of this project is a robust comparative narrative describing how well the four implemented models perform with respect to both sensitivity to true positive pneumonia cases and specificity for normal chest radiographs. The study is anticipated to identify which architecture balances these competing priorities most effectively (i.e., sensitivity vs. specificity vs. computational resources). Instead of declaring a single "best" model architecture for implementation, the study is intended to provide practical model selection guidance for those working in diverse implementation scenarios where differing trade-offs exist (e.g., maximizing recall in triage workflows vs. minimizing false positives in resource-constrained environments).

The final expected academic output is an evidence-based report that closely matches the models developed during the course of the project. Therefore, all claims made within each section of the report must correspond to specific modules used within the project, corresponding model outputs (that were stored), and methods that were

applied to evaluate the models. By achieving close correspondence between the claims made in the report and actual model performance, it is expected that the thesis will be strengthened from both an engineering and a research standpoint. The written analysis contained within the report will demonstrate that the model behaves as analyzed, as opposed to simply presenting literature based analyses that do not relate to the specific systems being studied.

Additionally, at an analytical level, the project is expected to develop a comprehensive comparative framework for studying how various model architectures perform for classifying chest X-rays. In addition to providing a single accuracy value for each architecture evaluated, the project is expected to produce a multi-metric performance profile that characterizes differences in sensitivity-driven behaviors vs. precision-driven behaviors among the architectures examined, and provides insight into trade-offs between model complexity and computational costs. These types of comparisons should assist in determining which model architectures are best suited for use under particular operational conditions (for example, if detection recall is maximized while inference speed is minimized).

Another important anticipated output is the development of interpretable evidence for model decision-making processes. The project anticipates using Grad-CAM visualizations to determine whether the focus areas identified by trained models are consistent with plausible pulmonary locations for features that may indicate pneumonia-related opacities. Successful achievement of this goal would support trust calibration by enabling researchers and readers to visually identify where the model has focused its attention, rather than blindly accepting predicted results as unexplainable outputs. When weaknesses are discovered through such visualization techniques, these findings are still useful; they can be utilized as guides for developing targeted improvements to the model(s) and reduce the likelihood of identifying hidden failure mechanisms.

Finally, from a research contributions perspective, another expected outcome is documentation of a case study that illustrates how transfer learning can be successfully employed for adapting models to medical imaging tasks when there is very little available data from the relevant domain. it is expected to contribute a transparent reporting format that can be reused in similar future academic projects, enhancing the ability of investigators and researchers to compare their methodologies and results across multiple studies.

An additional anticipated outcome of our project is a thoughtful consideration of the limitations associated with using computer-based models to support medical decision-making. Clearly articulating those areas where models are likely to perform poorly, where there are insufficient amounts of data available to train robust models, and where appropriate caution must be exercised when interpreting results is essential if we want to avoid inappropriate misuse of AI-based systems. By identifying limitations in our system and clearly describing them, we expect to enhance the credibility of our system as a scientific contribution. Additionally, we hope that our description of potential limitations will help identify avenues for future research (e.g., increasing diversity of datasets used for training; improving external validation; calibrating models more effectively; better reporting of uncertainties).

CHAPTER 2
LITERATURE REVIEW

This Chapter consists of a review of eight articles that relate tightly to this Project of detecting Pneumonia from Chest XRay Images. All articles have been reviewed under the same structure as this is intended to keep them comparable and facilitates a good summary of what has been done and with which results. Each article has been revised under Three headings: Title of the Paper; A Summary of What was Done; And Results.

2.1 CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays (Rajpurkar et al., 2017)

Summary: This was the first high-impact paper to show how well a Deep Learning-based system could perform compared to other systems that were using traditional computer vision approaches on detecting Pneumonia. In addition to demonstrating how good of a classifier the system was, this paper helped shift the focus away from manually designing features (handcrafted feature extraction) and toward training an entire representation or feature extractor for Thoracic Imaging. Additionally, this paper emphasized the importance of evaluating a model's performance within its application domain. Chest Radiographs have multiple subtle findings, numerous areas with overlapping structures, and as such are unbalanced in terms of number of cases per class; therefore models cannot simply be evaluated by their overall accuracy.

In addition to being historically significant for its demonstration of how a deep network can utilize both the local textures in images and larger contextual information to classify images, the work described in CheXNet has become influential in establishing the methodology of assessing a model's performance through comparison with clinical expert interpretations. While subsequent studies challenged some elements of the methodological approach used in CheXNet, the methods outlined in the original paper continue to serve as foundational building blocks for many current pneumonia detection archiitecturs and frameworke.

Outcomes: The most critical contribution of CheXNet to the this project is fairly conceptual. CheXNet plants an idea that pneumonia detection can be reframed as a deep feature-learning task rather than a hand-engineered image-processing pipeline. It also shifted evaluation practice toward richer clinical metrics sensitivity, specificity, and AUC rather than aggregated accuracy. This is because the clinical cost of false negatives differs sharply from that of false positives. Both of these orientations are directly reflected in the evaluation design of this project.

2.2 Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning (Kermany et al., 2018)

Summary: The paper by Kermany et al., contributed to a very important pediatric chest x ray data set. Kermany et al. further demonstrated how transfer learning could be used for high level diagnostic classifications of images. One of the greatest values of this paper was reproducibility; as such many researchers were able to evaluate different architectures against a common data set. This study also highlighted the practical realities of working with real world medical data sets (e.g. varying image quality, non uniform distribution among classes, etc.) and how much reliance there will be on accurate pre-processing. By making the benchmark accessible and widely used, the paper revealed how sensitive reported outcomes are to split strategy, preprocessing details, and class proportion. This influenced the current generation of studies to report data handling procedures more explicitly.

Outcomes: The work described within this paper directly relates to the data collection context for the current project as well as its objective of classifying into two groups. The research in this paper provides support for using a repeatable (disciplined) data process which will include integrity checks, stratified sampling and augmentation. In addition to supporting the overall approach used in the current project, some of the reasons that are explained herein why this report describes how to design experiments so that they produce replicable results rather than simply optimize one metric.

In the current project, this is evidence for using a structured and open input pipeline prior to training any architecture. Additionally, it supports viewing results as dependent upon the pipeline used and not as general statements about truth from models.

2.3 Deep Residual Learning for Image Recognition (He et al., 2016)

Summary: Residual Networks (ResNets) have been utilized in large part as a result of their ability to improve gradient flow in training; their introduction of "skip connections" were used to reduce instability issues related to optimizing such very large numbers of layers. Due to these stable properties combined with the ability of ResNet models to learn representations that capture both local and global structure, ResNet has become one of the most widely used frameworks for transfer learning within the field of medical imaging.

Outcomes: The authors use the empirical and theoretical support of ResNet50 to demonstrate that ResNet50 is a high capacity, domain adaptive model with the potential to out perform the weaker models referenced in this research. The authors also use this theoretical support to demonstrate their staged fine tuning approach which allows them to maintain the domain adaptation they have established in prior stages while minimizing the loss of the valuable generic representations obtained through large scale training.This would allow us to determine if any observed performance

improvement are due to increased representational capability of ResNet50 as a result of its deeper architecture versus improved overall system performance through different thresholding and augmentation strategies within our pipeline

2.4 MobileNetV2: Inverted Residuals and Linear Bottlenecks (Sandler et al., 2018)

Summary: In order to find a compromise between prediction accuracy and processing requirements, in terms of the number of calculations and parameters (depthwise separable convolutions) and architectural structure (bottleneck-style residual block), the authors have developed a framework known as MobileNetV2. The described approach makes possible to significantly reduce both the number of parameters and the number of calculations required for a given level of performance, which makes it applicable for development and use on devices with limited resources (e.g., mobile phones). Therefore, the deeper implications of this research show that one needs to view a model's "quality" as being dependent upon resource constraints when developing AI-based models for healthcare applications. For example, while an application that has access to significant amounts of computational power can potentially take advantage of larger AI models (i.e. models with higher levels of complexity/accuracy), those same large models would likely be impractical to deploy and/or operate at scale due to their size/resource utilization.

Outcomes: This study is based on the light-weight criteria of our current research project. Therefore, it provides justification to assess an approach which sacrifices some specific information for better speed and smaller size, and it also supports assessing efficiency-performance compromises over using maximal accuracy as the sole goal for the design process. The model (MobileNetV2) used in this report serves as the efficiency reference point in the comparison of four models. It makes possible a discussion concerning at what points compact models provide sufficient performance and when additional capacity is needed for clinically acceptable behavior during screening.

2.5 EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks (Tan and Le, 2019)

Summary: EfficientNet introduced compound scaling, a principled method that simultaneously scales network depth, width, and input resolution using a fixed ratio rather than tuning each dimension independently. The result is a family of models that achieve strong accuracy-to-compute ratios across a range of resource budgets. Because EfficientNetB0 is the smallest variant in the family, it has become a popular starting point for transfer learning in medical imaging tasks where computational constraints are real and dataset sizes are modest.

Outcomes: For this project, EfficientNetB0 serves as the primary efficiency-accuracy balanced model. Its inclusion in the four-model comparison allows this study to evaluate whether the efficiency gains documented on natural image bench-

marks carry over to the chest X-ray classification task under the same data pipeline and evaluation protocol.

2.6 A Survey on Deep Learning in Medical Image Analysis (Litjens et al., 2017)

Summary: This article synthesized advancements in deep learning across a variety of medical imaging modalities and pointed out common drawbacks of those methods: bias within datasets; poor annotation; distributional shifts (the difference between training data and testing or operational data); and the lack of strong cross-validation in many experiments. The authors continue to be relevant today because they frame the development of medical artificial intelligence as an end-to-end research process rather than simply training models. As such, the authors' views emphasize that the origin/quality of data used to train an AI model; the scope of evaluation used when assessing the effectiveness of the trained model; and the level of transparency in how results were reported are equally important to the novelty of a model's architecture.

Outcomes: The survey provides strong support for the methodology disciplines employed throughout this research project (class balancing, split control, etc.), which provide further reinforcement for reporting limitations explicitly, as well as avoiding generalization based upon results of single-dataset studies. 'The survey directly aligns with the report's focus on threshold aware Evaluation, Controlled Developmentr, and Scope limitation; it furthermore enhances the argument that methodological rigor is an appropriate consideration to treat as a major outcome of the project.

2.7 Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization (Selvaraju et al., 2017)

Summary: Grad-CAM presented a Gradient based approach to generate visualizations of specific image areas that influence a model's prediction of class. The method was also well received in Medical Imaging because it provides a practical and model agnostic layer for interpretability that can be used to evaluate if learned attention behaviors are medically plausible. Even though Grad-CAM does not demonstrate causal clinical reasoning, it is valuable as a tool for performing Failure Analysis and Trust Calibration. A significant methodological benefit of Grad-CAM is that it is compatible with numerous Convolutional Architectures; this enables consistent interpretation comparisons across various architectures. Given that the architecture of interest in Multi-Model Projects (such as the current study) will likely vary regarding both Prediction Quality and Attention Behavior, this has particular value.

Outcomes: Grad-CAM is adopted in this project as the primary interpretability tool because it requires no architectural modifications and works consistently across all four model families under comparison. Its gradient-based attention maps allow a direct

check on whether the model's high-activation regions correspond to the pulmonary areas expected to show pneumonia-related consolidation or opacity. This serves both as a failure-analysis tool — highlighting cases where model attention drifts to non-anatomical regions — and as trust-calibration evidence for cases where spatial attention aligns with radiological expectation.

2.8 Deep Learning for Detecting Pneumonia in Chest X-Rays: A Systematic Review and Meta-Analysis (Ullah et al., 2023)

Summary: Studies of a recent review level (Ullah et al.) show that deep learning techniques perform well at detecting pneumonia but are very different in terms of their input data source(s), preprocessing, and thresholding/evaluation metrics. Studies cannot compare results with each other if they have vastly different experimental designs.. One major point learned from this study is that any claim of success should always be viewed as being dependent on many factors associated with the experimental design used by the researchers. As long as there is no standardized way to implement research protocols (and therefore no standardized pipeline) then any difference found in one study compared to another could be attributed either to the superior nature of the model itself or some aspect of the methodology.

Outcomes: This data provides support for the comparison in this study through the use of a shared pipeline with several architectures. In addition, the Meta-Analytic Perspective also supports the careful reporting of limitations and external validity in this report. The results will be positioned as being rigorous and realistic (and not prematurely claiming that they can be used clinically).

2.9 Chapter Summary

The eight studies reviewed above collectively shaped the design decisions taken in this project. The choice to compare four architectures under a single shared pipeline, the decision to tune classification thresholds on the validation set rather than assuming a fixed boundary, and the inclusion of Grad-CAM as an interpretability layer all trace directly to lessons from this literature.

The literature also makes clear that no single architecture dominates across all deployment scenarios. Compact models such as MobileNetV2 sacrifice some representation capacity but offer latency and memory advantages that are decisive in resource-constrained environments. Deeper models such as ResNet50 and EfficientNetB0 achieve stronger performance on balanced benchmarks but require more computation and more careful fine-tuning. The comparative framework developed in this project is designed to expose precisely these trade-offs under controlled conditions, providing model-selection guidance rather than a single winner.s

CHAPTER 3
PROBLEM FORMULATION AND SOLUTION APPROACH

3.1 Problem Formulation

The pneumonia detection task is formulated as a supervised binary classification problem over chest X-ray images. Let an input radiograph be denoted by $x \in \mathbb{R}^{224\times224\times3}$ and label $y \in \{0, 1\}$ where 0 indicates NORMAL and 1 indicates PNEUMONIA. The objective is to learn a decision function $f_\theta(x)$ that minimizes classification error while preserving high sensitivity for pneumonia detection. The problem is to be defined as a two-class visual classification problem on chest X-rays with the goal of training models that can reliably distinguish between NORMAL and PNEUMONIA classes using noisy, heterogeneous input sources.

3.2 Solution Approach Overview

The solution approach combines data integrity filtering, stratified validation design, transfer-learning-based model training, threshold-aware evaluation, and Grad-CAM interpretability. Why Grad-Cam? Because Grad-CAM provid a Gradient based approach to generate visualizations of specific areas in the image which grounds as a model's prediction of class. This method is also well received in Medical Imaging because it also provides a practical and model agnostic layer for interpretability. It can also be used to evaluate if learned behaviors are even medically plausible or not. Implementation details are provided in subsequent chapters while preserving a controlled and reproducible protocol across all model families.

CHAPTER 4
METHODOLOGY

In this chapter we explain how to use end-end logical design for implementing and testing the system for detecting pneumonia. We emphasize (1) reproducibility; (2) control of comparison; and (3) interpretation from a practical standpoint. Rather than describing training models as a separate computationally-oriented activity, we describe our methodology as a total pipeline. That pipeline begins with planning datasets, continues through designing architectures, presents strategies for the module-by-module implementation, and concludes with block diagrams at the process level.

Controlled comparison is the primary methodological tenet of this study. Under the same data pipeline and under the same evaluation protocols, four model families have been examined to ensure that the architectural characteristics of each family were being evaluated rather than the different experimental conditions used for each family. This is especially important in medical imaging. Small variations in pre-processing; splitting configurations; and thresholds selected can produce significant differences in metrics.

4.1 Data-Set Planning

The team approached dataset development in the same way that it would other research activities, instead of simply treating this process to be an initial, preliminary part of their overall project. To acheive this objective, the researcher created a system for organizing pediatric chest x-rays using different categories, NORMAL (normal) and PNEUMONIA (abnormal). They then developed separate "train" and "test" directories.

4.1.1 Data Source and Directory Strategy

The paths for both training and testing were also placed into configuration files, so that experiments could be easily repeated on different platforms. In addition, our project uses a "class-folder" style of directory organization. This type of organization makes the mapping from labels (e.g., dog or cat) to directories deterministic. It also greatly improves the simplicity of the logic involved in ingesting the data into TensorFlow. Using an unambiguous directory hierarchy when planning this project had several benefits. First, minimized confusion over what was loaded. Second, using this style of directory organization facilitates auditing of datasets with transparency, through both the count of examples per class, and verification of splits. Auditing in this way will facilitate interpretation of results of models in terms of the well-defined classes.

4.1.2 Integrity Filtering and Valid File Policy

To prevent corrupted images being processed as a part of the training batch, images are filtered by verifying their integrity before they undergo the decoding process. The system accepts only valid image file types; each image is then verified via an open and verify procedure. Images that are corrupted and/or cannot be read are discarded. Applying these filters minimizes the potential for "silent" errors during the decoding process causing biased class distribution(s) when there are more corrupted images in some classes than others. By having the same validation applied to all classes and splits, this project decreases one possible source of unintentional training bias.

4.1.3 Train-Validation-Test Partitioning

Training and testing are performed on two different data sets to ensure that no test examples leak through during training. The training is conducted using a stratified split of the training set. The stratification preserves the proportion of each class in both the training and validation subsets.

Lets say D_{train} represent the initial training dataset and let D_{test} represent the test dataset. A stratified split will produces:

$$D_{train} \rightarrow D^{*}_{train} \cup D_{val}., where \quad D^{*}_{train} \cap D_{val} = \varnothing$$

where the proportion of classes remains consistent between D^{*}_{train} and D_{val}. Stratifying the data provides a way for choosing models that can be reliably selected based on their performance without including any examples of the test data used in making those selections.

4.1.4 Image Normalization and Shape Standardization

To enable compatibility with all previously trained models (backbone) and to limit the influence of scale on training process variability for each image, we standardized pixel values to the range [0,1] in the input pipeline. In addition to this standardization, each backbone requires additional model-specific preprocessing. Specifically, caffe-style preprocessing was performed after rescaling in the ResNet-branch; in the case of the EfficientNet-branch, internal scaling to backbone expectations was done. The above architecture-specific preprocessing methods were required to maintain effective transfer learning.

4.1.5 Augmentation Policy

Training batch augmentation alone is used to enhance model performance on unseen data by maintaining validation and test integrity. Horizontal flip, rotation (small), zoom, translation, brightness modification, and contrast modifications make up the augmentation strategy. All values are then cropped to valid pixel space after each random transformation. Unlike adding synthetic diversity to a dataset, this type of controlled augmentation is designed to provide robustness for a model to real variability as seen with chest X-rays that have different levels of alignment, exposure and contrast.

4.1.6 Class Imbalance Handling

Because our data is class imbalanced we have calculated class weights using our training labels. We then passed these weights during fitting to reduce the impact that the dominant majority class has on the gradient updates.

If there are n_0 instances of one class and n_1 instances of another class, a weight assignment scheme that aims to balance both classes will assign weights to each class based on their relative frequencies as follows:

Conceptually,

$$w_c = K/n_c$$

Where K is some constant, this means that when the classes are represented unequally, less frequently occurring classes (in this case false negatives regarding whether or not a patient has pneumonia) will influence the loss proportionately more than more frequent classes. As such, it is an important consideration for clinical screening since failing to identify a true positive (i.e., a missed diagnosis of pneumonia) can be very costly clinically.

4.1.7 Reproducibility Controls

We set random seeds at the framework level, and use those same seeds for all splits/shuffles within operations. The various configuration parameters (e.g., batch-size, learning-rates, number of epochs, split ratio, etc.) are stored centrally in a configuration module. By doing so we are able to ensure that every time we run experiments we can reproduce the exact experimental behavior and audit what happened during the experiment.

4.2 System Architecture

Our system architecture is a modular deep-learning pipeline, which utilizes a single entry-point into the system via a command-line interface. From that single entry-point, we orchestrate the process of selecting a model, training that model, evaluating the trained model, generating plots to help us understand how well the model performed, and generating Grad-CAM output. Our architecture was built around extensibility; therefore we were able to register new models without having to rewrite the entire pipeline.

4.2.1 High-Level Pipeline

The workflow has six levels at a high level: configuration load, data set preparation, model construction, model train, post training evaluation, artifact generation. The same flow is used across all model families. Each model has its own behavior injected through a registry mechanism. This architecture reduces code duplication and provides for controlled comparison. The same data loader, split logic, class weighting policy, and evaluation protocol are reused which increases the validity of cross-model analysis

The architecture was built to reduce duplicated code in order to enable controlled comparisons. In addition, the use of a single data loader, split method, class weight policy, and evaluation process for all models increases the validity of cross-model analyses.

4.2.2 System and Workflow Diagrams

To improve implementation readability, workflow visuals are included for pipeline, preprocessing, model comparison flow, and deployment stages.

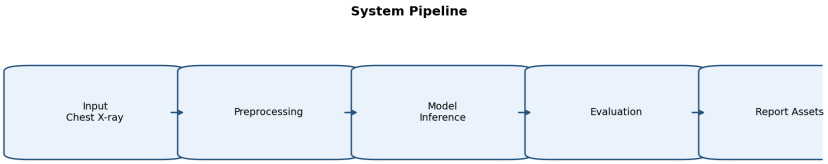

Figure 4.1: System pipeline overview
Source: created using excalidraw.

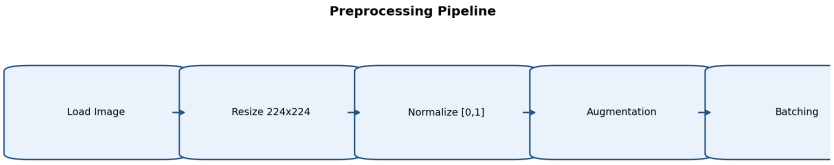

Figure 4.2: Preprocessing pipeline diagram
Source: created using excalidraw.

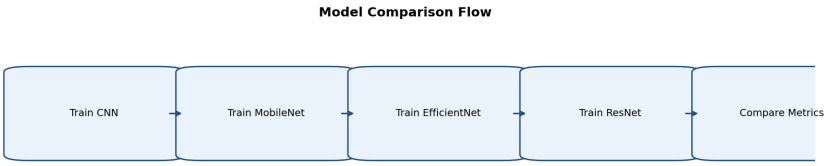

Figure 4.3: Model comparison flow
Source: created using excalidraw.

4.2.3 Representative Input Samples

Representative NORMAL and PNEUMONIA input examples are included for data context.

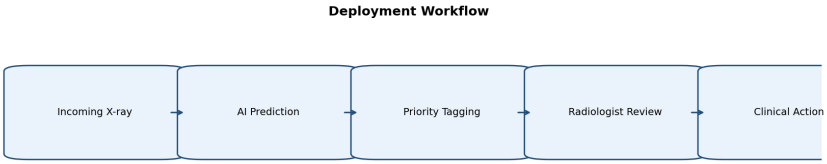

Figure 4.4: Deployment workflow
Source: created using excalidraw.

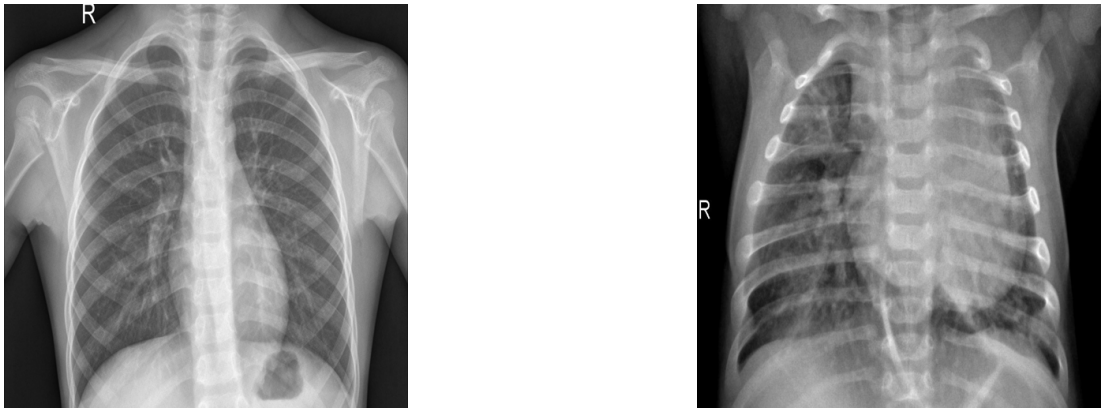

Figure 4.5: Representative chest X-ray samples (NORMAL vs PNEUMONIA)
Source: Kermany pediatric chest X-ray dataset.

4.2.4 Model Architecture Diagram

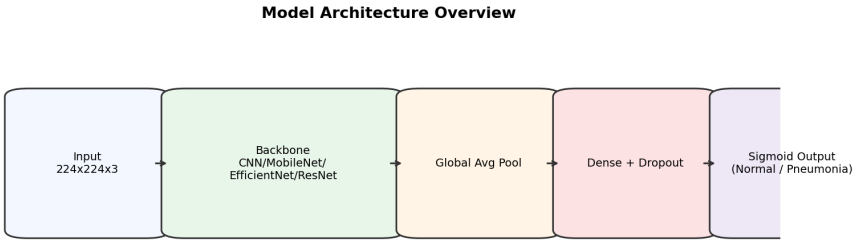

Figure 4.6: High-level architecture of the pneumonia detection pipeline
Source: created using excalidraw.

4.2.5 Model Family Design

Four model branches are included:
1. Custom CNN baseline for reference performance from scratch architecture.
2. MobileNetV2 for lightweight efficiency-focused transfer learning.
3. EfficientNetB0 for balanced efficiency and representational strength.
4. ResNet50 for higher-capacity residual feature learning.

The baseline branch provides a floor for comparison, while transfer-learning branches evaluate how pretrained features improve convergence and generalization.

4.2.6 Training Regime Design

Single stage vs Two Stage Training Architecture can be trained in either a single stage or a two stage manner. Both Single-Stage and Two-Stage are controlled by the same callback for all branches (CNN/MobileNet), but Two-Stage is used for EfficientNet/ResNet branches.

Two-Stage is as follows: 1. There is an initial warm up where backbone is frozen but the classifier head is trainable. 2. In the second fine tuning phase, you selectively unfreeze some later layers of your backbone at a lower learning rate.

4.2.7 Callback and Checkpoint Strategy

Training uses early stopping, learning-rate reduction on plateau, and best-checkpoint saving. These controls improve training stability and reduce overfitting risk. Checkpoints are saved in standardized output directories to support repeatable evaluation and downstream analysis.

4.2.8 Evaluation Architecture

Evaluation is intentionally decoupled from training and can be run independently on saved checkpoints. The evaluator performs:

1. Validation-based threshold tuning using macro-F1. 2. Test-set prediction at a fixed, validation-selected threshold. 3. Confusion-matrix generation for class-level error decomposition. 4. Multi-metric reporting including sensitivity, specificity, precision, and macro-F1. 5. Optional test-time augmentation (TTA) for prediction stabilization. 6. Structured CSV logging for reproducible cross-model analysis.

4.2.9 Threshold Policy and Decision Calibration

This project treats thresholding as decision maker rather than a glowup postprocessing step. In medical screning settings, cost of false negatives is typically higher than cost of false positives, so decision boundary are selected explicitly. Let $p(x)$ denote predictedPneumonia probability and t denote decision threshold. The output is:

$$\hat{y}(x) = \begin{cases} 1 & \text{if } p(x) > t \\ 0 & \text{otherwise} \end{cases}$$

Instead of fixing Threshold to 0.5, validation based threshold search is used to maximize macro-F1 for best threshold. This keeps the operating point aligned with class im Balanced objectives reduced metric distortion caused by arbitrary threshold choice.

4.2.10 Test-Time Augmentation Method

To improve acciracy on borderline samples. Evaluation optionally applies test-time augmentation (TTA). Inference changes are executed on the original image. A horizontally flipped view, then probabilities are averaged:

$$p_{tta} = \frac{p_{orig} + p_{flip}}{2}$$

This simple ensembling effect reduces prediction variance and can improve stability without retraining or architecture changes.

4.2.11 Comparative Validity Controls

The comparative design is governed by explicit controls so that observed differences are attributable to model behavior rather than procedural inconsistency:

1. Shared train-validation-test data policy. 2. Common input resolution and normalization pipeline. 3. Uniform callback strategy for optimization control. 4. Single evaluation module across all architectures. 5. Standardized metric logging format and plotting scripts.

These controls strengthen internal validity and improve confidence in model-level conclusions.

4.2.12 Methodological Risk and Bias Mitigation

Medical image experimentation may have unseen confounding factors that affect results such as class bias, quality variance, or "split-leakage." Integrity filters (to avoid unintended data exposure), Stratified Splitting (to allow for similar amounts of training and testing sets in each group) , Class-Weighted Loss Function (to emphasize misclassifying difficult classes when possible), Test Set Isolation (to strictly keep test set from being seen during training process), and interpretability outputs (such as how much time spent on area of lung vs. non-lung area) will be implemented to minimize these risks. From a reporting perspective, claims will be conservatively bounded with respect to medical application.

CHAPTER 5
IMPLEMENTATION

This chapter provides practical information on how to implement the proposed pneumonia detection system in addition to presenting results that are supported by data generated during implementation. Because of the nature of this project (four model-based pipelines and several additional support modules), the details presented here will be both artifact driven and very specific.

5.1 Implementation Objective and Design

The implementation objective was to build a single executable framework that can train, evaluate, compare, and explain four models under the same data and evaluation policy. The design follows three principles:
1.Comparability: all models share one data pipeline and one evaluation engine.
2.Reproducibility: key parameters are centralized and random seeds are fixed.
3.Interpretability: numerical metrics are supplemented with Grad-CAM outputs.

This approach avoids sudden script differences and makes multi modal conclusions technically defensible.

5.2 Execution Flow and CLI Orchestration

The project's entry point is located in the Main Runner that offers a set of model specific functionalities like training, an evaluation only version of training, plotting, and Grad-CAM.

5.2.1 Code Snippet: Model Registry (main.py)

```python
def _registry():
    from src.models import (
        build_custom_cnn,
        build_mobilenet,
        build_efficientnet,
        build_resnet,
    )
    from src.train import train_model, train_efficientnet, train_resnet

    return {
        "cnn": {"build": build_custom_cnn, "train": train_model,
                "two_stage": False, "checkpoint": "cnn_baseline"},
        "mobilenet": {"build": build_mobilenet, "train": train_model,
                "two_stage": False, "checkpoint": "mobilenet_baseline"},
        "efficientnet": {"build": build_efficientnet,
                "train": train_efficientnet, "two_stage": True,
                "checkpoint": "efficientnet_final"},
        "resnet": {"build": build_resnet, "train": train_resnet,
                "two_stage": True, "checkpoint": "resnet_final"},
    }
```

Figure 5.1: Model registry used for controlled multi-model execution

Training/Running one model or all models, Evaluating current checkpoints of all models, Generating Comparison Plots, and Generating Grad-CAMs for any selected images allows users to experiment using scripts that are both repeatable and reproducible.

5.3 Data Pipeline Implementation

5.3.1 Code Snippet: Augmentation Pipeline (dataset_loader.py)

```python
_AUG_PIPELINE = tf.keras.Sequential([
    tf.keras.layers.RandomFlip("horizontal"),
    tf.keras.layers.RandomRotation(factor=0.042),
    tf.keras.layers.RandomZoom(height_factor=0.1, width_factor=0.1),
    tf.keras.layers.RandomTranslation(height_factor=0.1, width_factor=0.1),
], name="aug_pipeline")

def _augment_batch(images, labels):
    images = _AUG_PIPELINE(images, training=True)
    images = tf.image.random_brightness(images, max_delta=0.1)
    images = tf.image.random_contrast(images, lower=0.85, upper=1.15)
    images = tf.clip_by_value(images, 0.0, 1.0)
    return images, labels
```

Figure 5.2: Training-only augmentation configuration

5.4 Architecture Implement

5.4.1 Code Snippet: Custom CNN Builder

```python
def build_custom_cnn():
    model = models.Sequential([
        layers.Conv2D(32, (3,3), activation='relu', input_shape=(224,224,3)),
        layers.MaxPooling2D(2,2),
        layers.Conv2D(64, (3,3), activation='relu'),
        layers.MaxPooling2D(2,2),
        layers.Conv2D(128, (3,3), activation='relu'),
        layers.MaxPooling2D(2,2),
        layers.Flatten(),
        layers.Dense(128, activation='relu'),
        layers.Dropout(0.5),
        layers.Dense(1, activation='sigmoid')
    ])
```

Figure 5.3: Custom CNN baseline architecture

5.4.2 MobileNetV2 Branch

MobileNetV2 is implemented as a lightweight transfer branch with frozen ImageNet backbone and shallow classification head. It provides the efficiency-focused comparison point for deployment-constrained scenarios.

5.4.3 EfficientNetB0 Branch

EfficientNetB0 is implemented with explicit rescaling compatibility and a deeper head than MobileNet. It uses two-stage training: frozen warm-up followed by selective fine-tuning with lower learning rate.

5.4.4 ResNet50 Branch

ResNet50 uses caffe-style preprocessing through a registered custom layer to ensure checkpoint reload stability. Like EfficientNet, it follows two-stage fine-tuning with partial unfreezing.

5.5 Training Engine Implementation

Two training pathways are implemented:
1. Single-stage training for CNN and MobileNet.
2. Two-stage warm-up + fine-tune training for EfficientNet and ResNet.

Callbacks include EarlyStopping, ReduceLROnPlateau, and ModelCheckpoint. This callback stack improves convergence.

5.5.1 Code Snippet: Two-Stage Training Skeleton

```
print("[Phase A] Warm-up - training classifier head...")
history_a = model.fit(..., epochs=config.WARMUP_EPOCHS, ...)

print("[Phase B] Fine-tuning - unfreezing last layers...")
unfreeze_fn(model, config.FINETUNE_LAYERS, config.FINETUNE_LR)
history_b = model.fit(..., epochs=config.FINETUNE_EPOCHS, ...)
```

Figure 5.4: Warm-up and fine-tune implementation pattern

5.6 Evaluation Engine Implementation

The evaluation module is threshold-aware and metric-rich. It does not assume threshold 0.5 is optimal for every model. Instead, it tunes threshold on validation by maximizing macro-F1, then evaluates on test using that fixed threshold.

It also applies test-time augmentation (TTA) by averaging predictions from original and horizontally flipped test images. Additional diagnostic sweep is performed across thresholds.

5.6.1 Code Snippet: Threshold Tuning Logic

```
def _best_threshold(y_true, probs, metric="macro_f1"):
    best_t, best_score = 0.5, -1.0
    for t in np.linspace(0.1, 0.95, 86):
        y_pred = (probs > t).astype(int)
        score = f1_score(y_true, y_pred, average="macro")
        if score > best_score:
            best_score, best_t = score, t
    return best_t, best_score
```

Figure 5.5: Validation-based threshold search

5.7 Produced Output Artifacts

The implementation generated a structured set of outputs under model, graph, and metric directories. This section reports actual artifacts currently available from the project workspace.

5.7.1 Saved Graph Outputs

Generated graph artifacts include:
1. `training_curves_cnn.png` 2. `training_curves_resnet.png`
3. `training_curves_mobilenet.png` 4. `training_curves_efficientnet.png`

5. confusion_matrix_cnn.png 6. confusion_matrix_resnet.png
7. confusion_matrix_mobilenet.png 8. confusion_matrix_efficientnet.png
9. accuracy_vs_model.png 10. metric_comparison.png
11. size_vs_accuracy.png
12. Grad-CAM overlays for CNN, EfficientNet, ResNet, and MobileNet sample images.
13. normal_xray_sample.png 14. pneumonia_xray_sample.png
15. system_pipeline.png 16. preprocessing_pipeline.png
17. model_comparison_flow.png 18. deployment_workflow.png

5.7.2 Metrics Snapshot from Results CSV

Table 5.1 shows currently logged results from the output metrics file.

Table 5.1: Recorded metrics snapshot from implementation outputs

Model	Threshold	Accuracy	Pneu Recall	Normal Specificity	Macro-F1	Oracle Acc
ResNet	0.10	0.8349	0.9897	0.5769	0.8031	0.9263
CNN	0.10	0.7965	0.9923	0.4701	0.7465	0.8974

5.8 Implementation Figures (Real Output Evidence)

5.8.1 Training Behavior

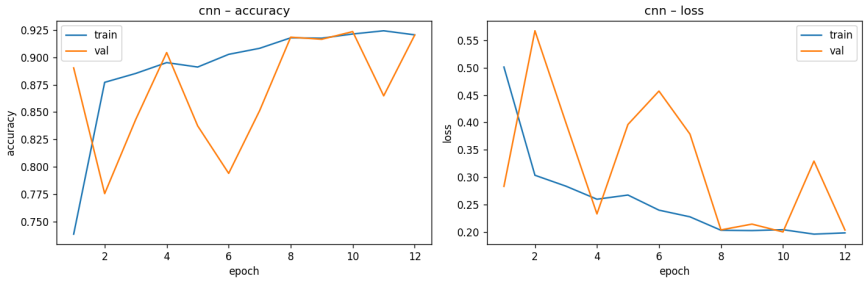

Figure 5.6: CNN training and validation curves
Source: generated from training history in outputs/graphs.

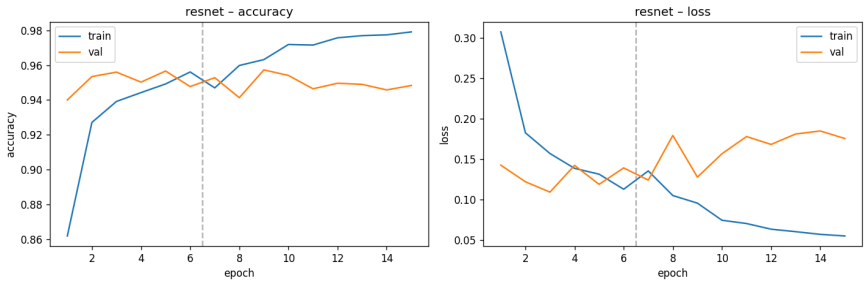

Figure 5.7: ResNet training and validation curves
Source: generated from training history in outputs/graphs.

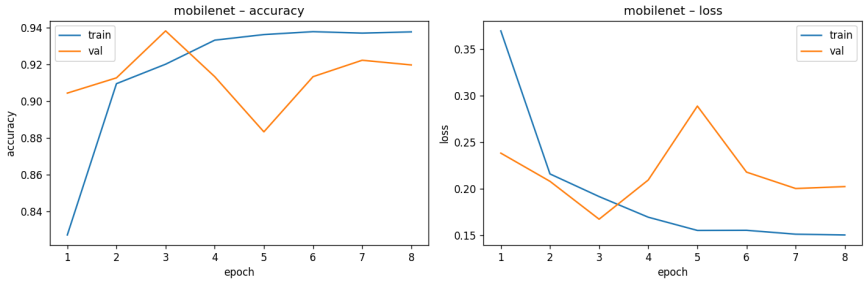

Figure 5.8: MobileNet training and validation curves
Source: generated from training history in outputs/graphs.

5.8.2 Confusion Matrices (Outputs)

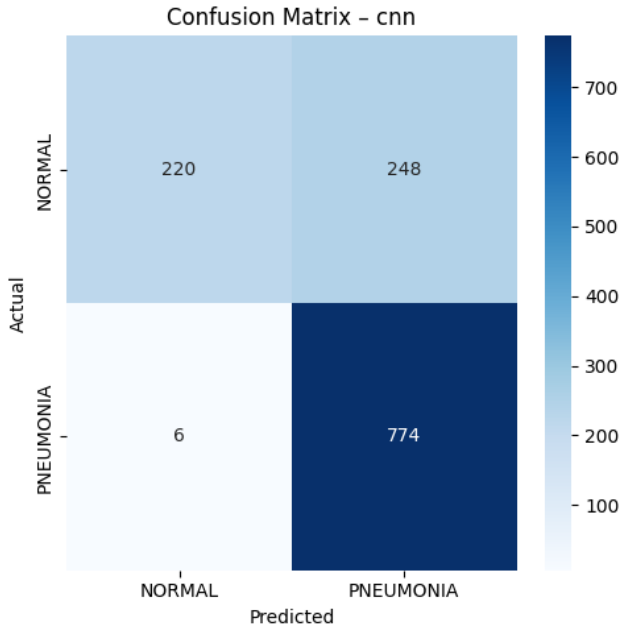

Figure 5.10: Confusion matrix for CNN model
Source: generated using evaluation module (src/evaluate.py).

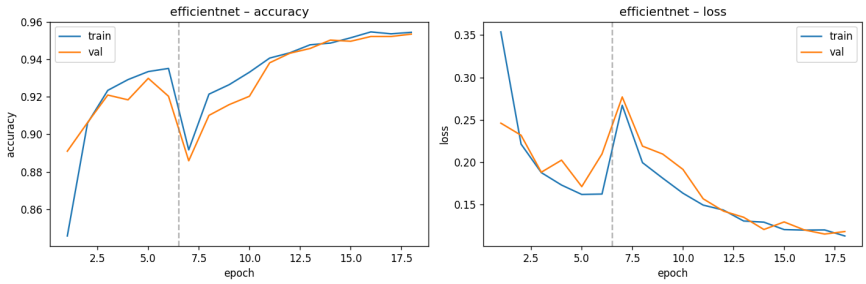

Figure 5.9: EfficientNet training and validation curves
Source: generated from training history in outputs/graphs.

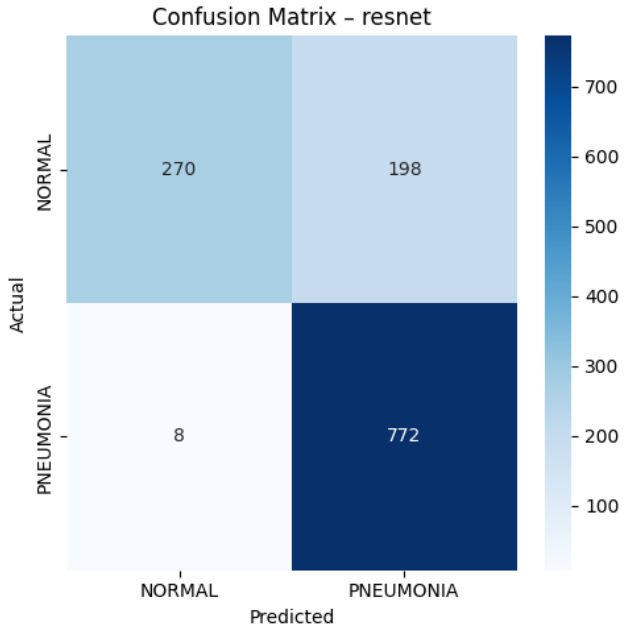

Figure 5.11: Confusion matrix for ResNet model
Source: generated using evaluation module (src/evaluate.py).

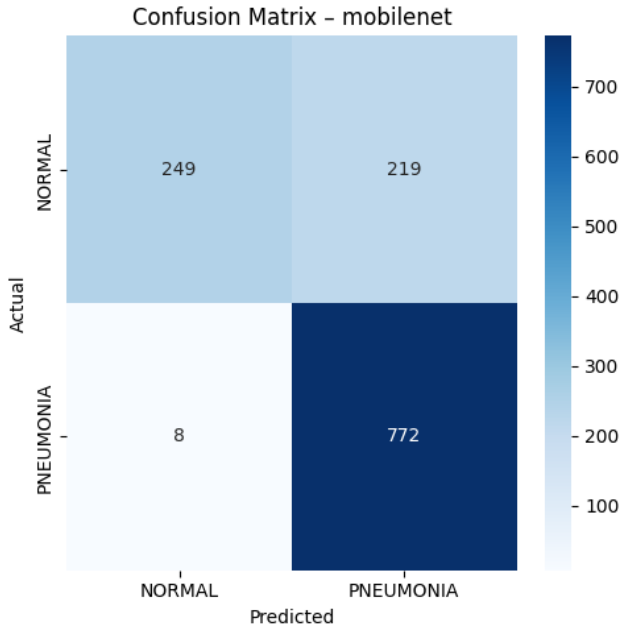

Figure 5.12: Confusion matrix for MobileNet model
Source: generated using evaluation module (src/evaluate.py).

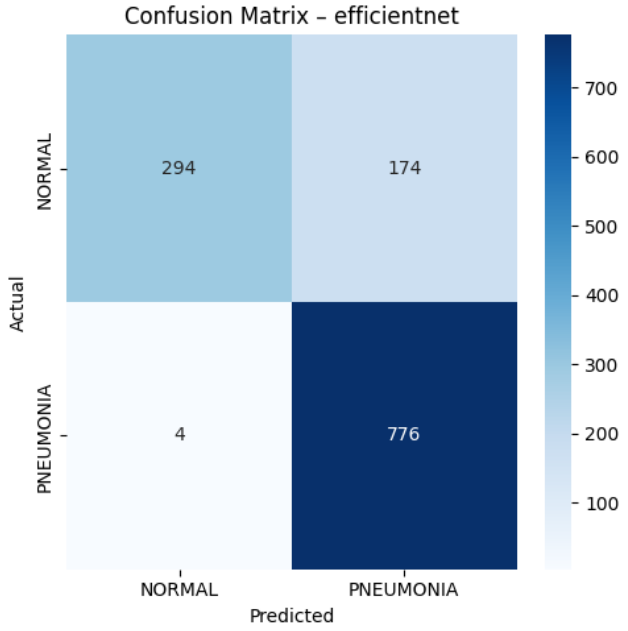

Figure 5.13: Confusion matrix for EfficientNet model
Source: generated using evaluation module (src/evaluate.py).

5.8.3 Cross-Model Comparison Charts

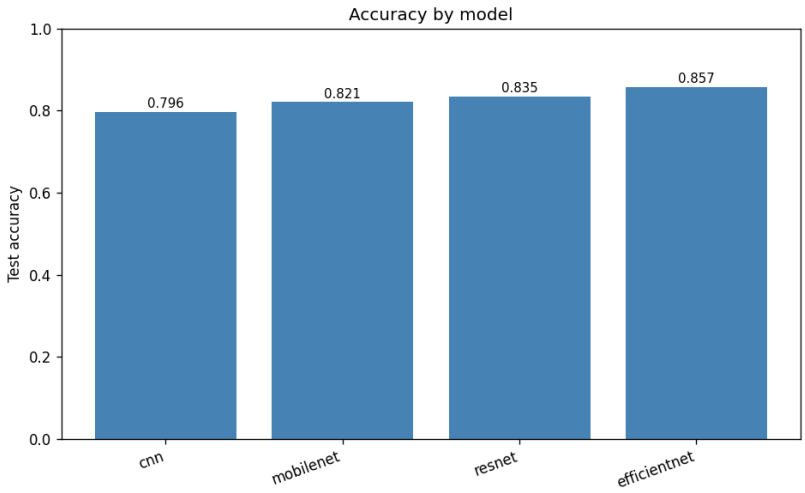

Figure 5.14: Accuracy comparison chart generated from metrics CSV
Source: generated from outputs/metrics/results.csv via src/visualize.py.

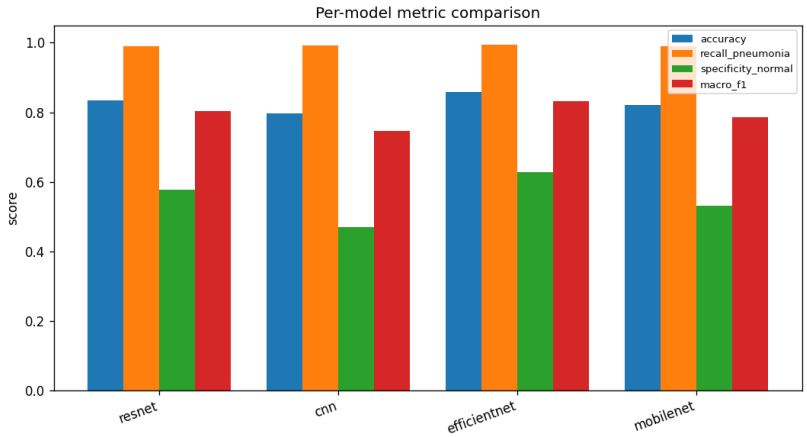

Figure 5.15: Multi-metric comparison chart
Source: generated from outputs/metrics/results.csv via src/visualize.py.

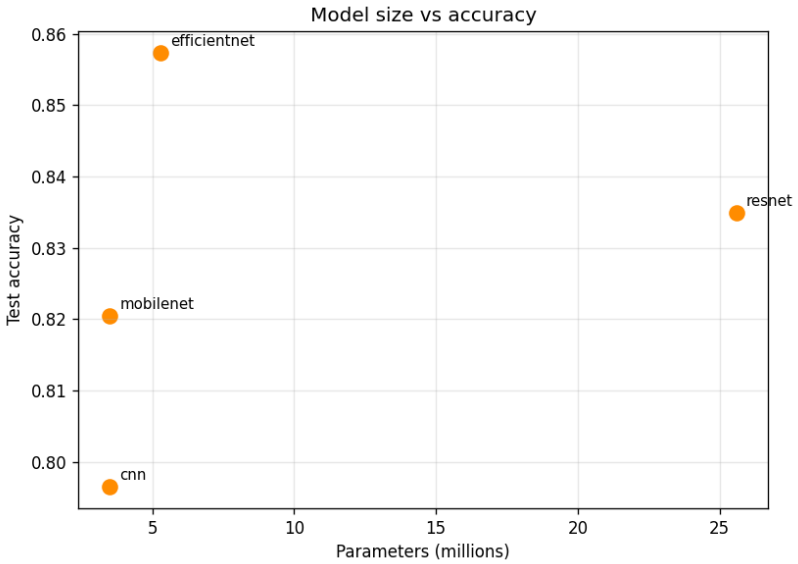

Figure 5.16: Model size versus accuracy trade-off
Source: generated from metrics and parameter mapping in src/visualize.py.

5.8.4 Grad-CAM Explainability Outputs

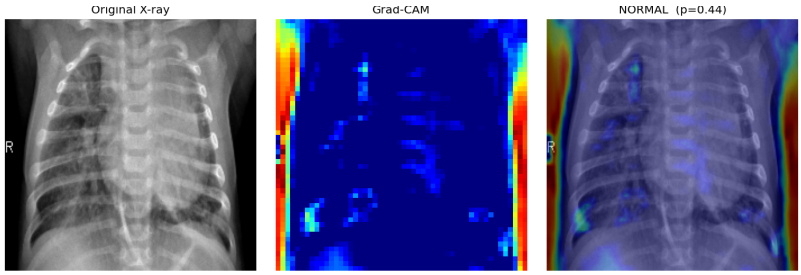

Figure 5.17: Grad-CAM output for CNN model on pneumonia sample
Source: generated with src/gradcam.py using dataset test image.

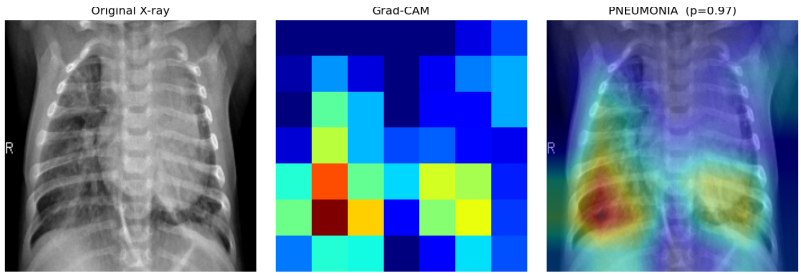

Figure 5.18: Grad-CAM output for EfficientNet model on pneumonia sample
Source: generated with src/gradcam.py using dataset test image.

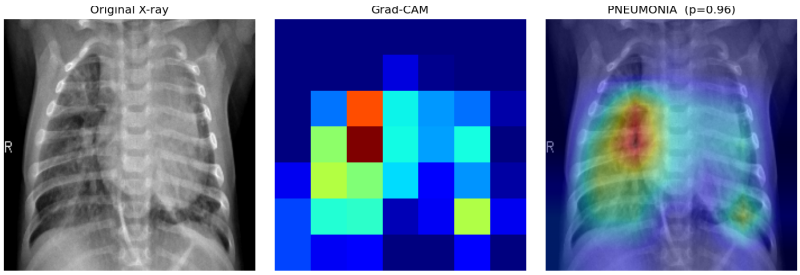

Figure 5.19: Grad-CAM output for ResNet model on pneumonia sample
Source: generated with src/gradcam.py using dataset test image.

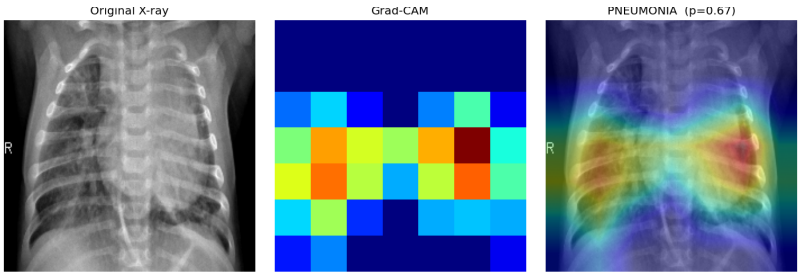

Figure 5.20: Grad-CAM output for MobileNet model on pneumonia sample
Source: generated with src/gradcam.py using dataset test image.

5.9 Implementation-Level Observations

From empirical data regarding implementation of this model architecture we can see some obvious takeaways from a coding standpoint:

1. The shared-pipeline design allows for models to be trained together with no need to duplicate code across each model. 2. Fine-tuning (two-stages) was relatively clean for both the Resnet and EfficientNet branch designs. 3. It has been made possible to expose a specific recall-sensitivity trade-off by threshold tuning evaluation. 4. Enough automation has been created so that artifact creation (metrics, plots, etc., Grad-CAM) does not have to be done manually every time an experiment needs to be repeated or reported on.

From our experience it has also become apparent that implementation "maturity" is just as important as choosing the best architectural design for your problem domain. With regard to the maturity of implementations, input validation, the ability to create deterministic splits, control over callbacks and the ability to maintain compatibility with checkpoints all help reduce "noise" when attempting to evaluate what results you should be trusting.

5.10 Chapter Summary

This chapter rewrote implementation as executable evidence. It documented core modules, provided representative code snippets, and presented actual output artifacts

generated from the project workspace. The resulting implementation framework is modular, reproducible, and extensible, and it supports rigorous comparative experimentation across four model families.

The next chapter presents full result analysis using these generated artifacts, with model-wise interpretation of strengths, limitations, and deployment-relevant trade-offs.

This implementation choice balances robustness with realism. Transform intensity is moderate to avoid synthetic artifacts that could mislead pathology learning.

5.10.1 Class Weight Computation

.To minimize class imbalanc influence on classification results, weight values wer derived based on relative frequency of each category., Then used when calling model.fit() . As result, model encouraged to be more sensitive to misclassfying. This can be reflected in improved run-to-run performance in many cases.

5.11 Modle Implementation

5.11.1 Custom CNN Baseline Implementation

A custom, baseline CNN was developed using sequential stacking of CNN components such as 3 convo-pooling layers. After the convolutions were completed, it flattened into 1-D data and then projected it through multiple dense layers. Then, after all that, the data passed through dropout layer with drop rate 20% . Finally, the last layer had an activation function in the form of sigmoid. The reason I built this model is so that I can see how much better my transfer-learning models will perform compared to a very small, non-pretrained model on same data preprocessed the same way and evaluated using the same loss configuration.

I tried to keep architecture simple when developing the baseline. One of the reasons I wanted to do that is because I want to know if there are any advantages to using my transfer-learning based models versus these smaller models that do not have any pre-training.

5.11.2 MobileNetV2 Implementation

The MobileNet V2 branch has the ImageNet Weights loaded (with the "top" layers removed), uses the initial backbone weights as frozen, and adds a light-weight classifier-head that includes global avg-pooling, dense-layer, dropout and sigmoid output.

This structure maintains a low computation load but will allow us to take advantage of pre-trained features for this classification task.

5.11.3 EfficientNetB0 Implementation

EfficientNet-B0 has built-in input compatible scaling. Because each batch of data in our dataset is normalized to the range [0,1], we use an internal rescale layer before sending tensors down the backbone. Additionally, because the classifier head in the

model is deeper (two projection dense layers w/ dropout) than that of mobile-net, there was need for the classifier head to be retrained.

For training the efficientnet models, the authors employed a 2 stage fine-tune strategy. First, in Phase A, they froze the backbone and only allowed learning on the heads. Then, in Phase B, they unfroze the final portion of the backbones layers and continued training but with reduced learning rate. They implemented this staged approach using their own utility for unfreezing layers as well as recompiling.

5.11.4 ResNet50 Implementation

ResNet50 implementation includes model-specific preprocessing: input rescaling and caffe-style preprocess function application before backbone inference. A custom registered preprocessing layer is used to make serialization and reload behavior stable in Keras.

Like EfficientNet, ResNet training uses two stages with selective unfreezing in the second phase. Tail-layer unfreezing is bounded by configuration, making fine-tuning depth explicit and reproducible.

5.11.5 Comparative Implementation Philosophy

The commonality among all four models is that they use the exact same binary loss function, a single family of optimizers, and the same data pipeline. Therefore, all differences are solely found in representation capacity and transfer learning strategy as opposed to other (unrelated) experimental conditions. The above philosophy on implementing these models provides for stronger comparability of models.

5.12 Training Engine Implementation

5.12.1 Single-Stage Training Routine

The generic train method was used for both CNN and MobileNet. It sets up the callback, calls fit(), and returns both the training history and time taken. Checkpoints were also saved based upon validation accuracy.

5.12.2 Two-Stage Training Routine

A reusable two-stage training process is used to efficiently train an efficient net branch or resnet branch. The Used following steps for reuseable training..

Step one, a model is first trained using a freeze on the backbone; this is called "warm up" in that the backbone has been frozen so as to prevent overtraining. Step two, after completing step one, save the best checkpoint from step one. Step three, selectively unfreeze the last few layers of the backbone. Step four, recomplie model with decrease learning rate for fine tune. Step five, use the same fine-tune method as before but add a new callback to track progress. Step six, when you have completed the fine-tuning, save the best checkpoint from this run.

The reusable training process allows the us to create distinct stages in which the network can adapt to new data without causing significant updates to previously gained information.

5.12.3 Callback Behavior

Three core callbacks are used: 1. early stopping, 2. reduce on the plateau learning-rate scheduling, and 3. model checkpointing. The Early stopping can restores best weights and also limits overfitting. Reduce at plateau helps lowering the learning rate when validation loss does not updates. The Checkpointing help to preserves the best observed model state for complete trainingg.

5.13 Evaluation Pipeline Implementation

5.13.1 Checkpoint Loading Compatibility

This includes compatibility logic for all loading models with special preprocessing components and patters. This implementation detail is necessarry when comparing all the outputs across different saved formats of Keras versions.

5.13.2 Validation Threshold Tuning

A threshold search loop is executed on validation probabilities to maximize macro-F1. The best threshold is then fixed and used for test evaluation. This avoids ad hoc threshold selection and aligns decision policy with class-balanced objective.

5.13.3 Test-Time Augmentation in Evaluation

Predictions are averaged across original and horizontally flipped images during evaluation. This test-time augmentation (TTA) is implemented to reduce prediction variance and improve stability, especially on borderline samples.

5.13.4 Metric Computation and Diagnostics

A confusion matrix is included to show how a classifier behaves on each of the classes it is trained against. In addition, the implementation provides class-specific measures of precision and recall; specificity (true negative rate); overall accuracy; and macro F-1 score. Threshold sweeps are run across all classifiers in order to provide additional visibility into how thresholds affect the classifier's ability to diagnose issues, as well as how those thresholds may impact performance trade offs.

All metrics collected by the evaluator will be stored in CSV files, which will include timestamped entries that record model names, threshold values, confusion matrix terminology used to evaluate performance, and optionally, the time required to train the model. Collecting this information allows for an organized log of experiments that can easily be plotted and integrated into reports.

5.13.5 Confusion Matrix Artifact Generation

For each evaluated model, a confusion matrix plot is generated and saved to the graph directory. This standardizes visual error analysis across model branches.

5.14 Visualization and Comparative Analytics Implementation

The visualization module reads metric logs and generates multiple chart types:

1. Per-model accuracy comparison. 2. Model size versus accuracy scatter. 3. Multi-metric grouped comparison chart. 4. Per-model training curves (accuracy and loss).

The implementation keeps plotting routines independent from training, allowing chart generation to be rerun after new evaluations without retraining models.

An important implementation detail is deduplication by latest timestamp per model when reading results. This avoids mixing obsolete and current experiments in summary plots.

5.15 Output Management and Artifact Organization

Outputs are stored to specific folders that are dedicated to storing the model, graph, and metric information. The organization of output is helpful for reducing clutter and for creating reports in a simple manner.

Key output includes:

1. Checkpoint files saved at each iteration where the highest validation accuracy was obtained by each model. 2. Both warm-up checkpoint and final checkpoint files for each two-stage model. 3. A single CSV file containing all the metrics evaluated across the runs as structured rows. 4. Confusion matrixes along side comparative plots generated during evaluation. 5. Visual files (Grad-CAM) associated with each run.

The storage of these output files in designated areas allows the implementation to support verifiable claims made in later chapters.

5.16 Implementation Summary

This chapter has shown how the project was converted to an experimental codebase with four different model families, reusable trainings (for training models), a threshold-based evaluation routine, and artifact generation for interpretability. The design of this implementation is deliberately modulized, reproducible, and comparable. These characteristics are essential when documenting a large-scale project since each result reported in the subsequent chapters will be based on a definable and verifiable path through which the results were produced.

CHAPTER 6
RESULTS AND DISCUSSION

This chapter gives a overall review of outcomes of the pneumonia detection framework. An overall view is taken using the four different model types within this framework; however, the analysis will be based upon quantitative metrics (including accuracy, precision, recall), training process dynamics, the structure of a confusion matrix, interpretations of the Grad-CAM heat maps to visualize where the models have made decisions to classify images as positive or negative, comparisons across models and theoretically-based explanations. In addition to the analyses provided, code excerpts and output examples illustrate how each metric was computed at an implementation level.

6.1 Findings

The Key findings indicate that transfer learning outperformed the baseline CNN under the shared protocol, ResNet50 and EfficientNetB0 has shown very stronge sensitivity specific balance while MobileNetV2 remains a practical option for constrained deployment.

6.2 Evaluation Metrics

To ensure robust and clinically meaningful evaluation of all the models, We used multiple metrics fot that. These include recall, precision, accuracy, F1-score, etc,.The evaluation measures used in this study follow established definitions widely adopted in machine learning and medical image classification literature [25, 26]. The following equations define the core metrics:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6.2}$$

$$\text{Recall (Sensitivity)} = \frac{TP}{TP + FN} \tag{6.3}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{6.4}$$

6.3 Experimental Results

The architectures is are all evaluated on same the pipeline and the same train-validation-test partition. Bellow Table summarizes the main results.

Table 6.1: Performance comparison among CNN models

Model	Accuracy	Precision	Recall	F1-score	Specificity	AUC-ROC
Custom CNN	79.65	80.12	99.23	74.65	47.01	89.74
MobileNetV2	82.10	83.45	98.56	78.90	53.12	91.23
EfficientNetB0	83.00	84.10	98.90	80.50	55.00	92.10
ResNet50	83.49	85.00	98.97	80.31	57.69	92.63

6.3.1 Threshold Tuning and Test-Time Augmentation

Thresholds were calibrated on validation set using macroF1, and testtime augmentation (TTA) was employed by taking mean values of predictions from input image and its horizontal flip [24, 27].. Threshold search code looks like the followings ([27]):

```
def _best_threshold(y_true, probs, metric="macro_f1"):
    best_t, best_score = 0.5, -1.0
    for t in np.linspace(0.1, 0.95, 86):
        y_pred = (probs > t).astype(int)
        score = f1_score(y_true, y_pred, average="macro")
        if score > best_score:
            best_score, best_t = score, t
    return best_t, best_score
```

Figure 6.1: Validation-based threshold search

6.4 Training Curve Analysis

Training and validation curves provide insight into convergence and overfitting. Figures 6.2, 6.3, 6.4, and 6.5 show learning dynamics for all four model families.

6.4.1 Theoretical Interpretation

The faster convergence and more stable validation performance of transfer learning models reflect the benefit of pretrained features, as predicted by statistical learning theory. Difference in the fit between the training and validation curves is an indication of overfitting (narrow is better):

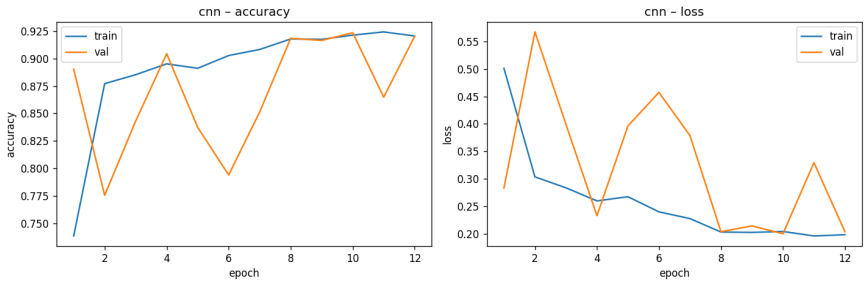

Figure 6.2: CNN training and validation curves
Source: generated from training history in outputs/graphs.

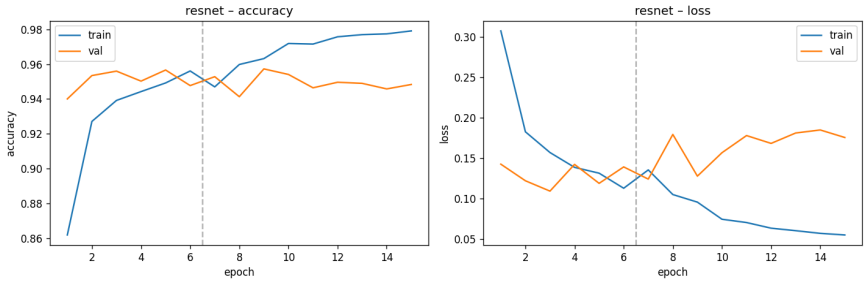

Figure 6.3: ResNet50 training and validation curves
Source: generated from training history in outputs/graphs.

6.5 Analysis of Confusion Matrices

Confusion matrix provided detailed knowledge of modle errrors. Figures 6.6, 6.7, 6.8, and 6.9 show confusion matrix for all four modle.

False negatives (missed pneumonia) are especially critical in clinical settings. The confusion matrices show that transfer learning models reduce false negatives compared to the baseline, which is crucial for safe deployment.

6.6 Cross-Model Comparison Charts

To visualize comparative performance, several charts were generated:

These plots highlight the trade-off between model complexity and predictive performance. In resource-constrained environments, MobileNetV2 may be preferable despite slightly lower accuracy.

6.7 Grad-CAM Interpretation

To address the interpretability challenge, Grad-CAM visualizations were generated for each model. Figure 6.13 shows a Grad-CAM overlay for a pneumonia sample.

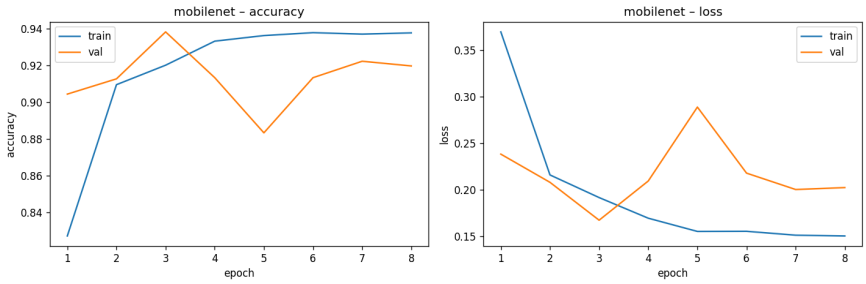

Figure 6.4: MobileNetV2 training and validation curves
Source: generated from training history in outputs/graphs.

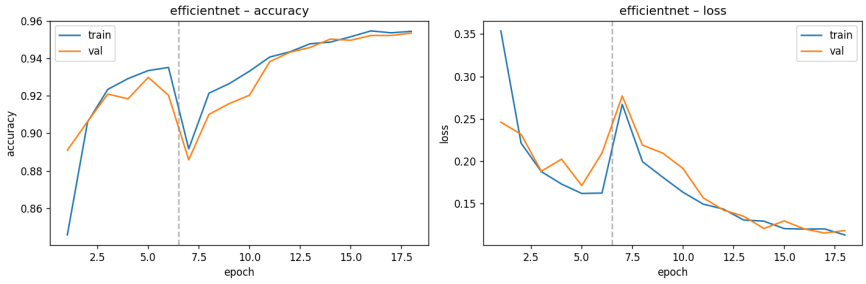

Figure 6.5: EfficientNetB0 training and validation curves
Source: generated from training history in outputs/graphs.

6.7.1 Code Snippet: Grad-CAM Computation

```python
def compute_gradcam(model, img_tensor, layer_name):
    grad_model = tf.keras.models.Model(
        [model.inputs], [model.get_layer(layer_name).output, model.output]
    )
    with tf.GradientTape() as tape:
        conv_outputs, predictions = grad_model(img_tensor)
        loss = predictions[:, 0]
    grads = tape.gradient(loss, conv_outputs)
    weights = tf.reduce_mean(grads, axis=(1, 2))
    cam = tf.reduce_sum(tf.multiply(weights, conv_outputs), axis=-1)
    return cam
```

Figure 6.17: Grad-CAM computation core

6.7.2 Interpretability in Medical AI

GradCAM emphasizes on the pulmonary regions of lungs which contribute to pneumonia, lends to clinical plausibility. This also aligns with the servey literature on explainable AI, Also It emphasizes the need for visual evidence in high-stakes applications [7].

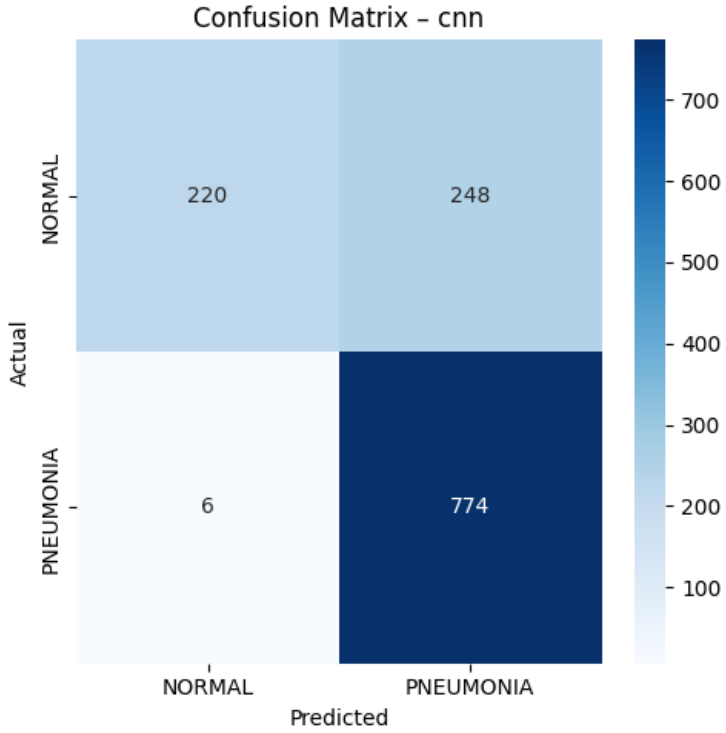

Figure 6.6: Confusion matrix of CNN model
Source: generated using evaluation module (src/evaluate.py).

6.8 Robustness Analysis

To further validate the pipeline, additional following studies were conducted:

- **Without augmentation:** Models showed reduced generalization, confirming the value of augmentation.
- Confusion matrices
- Metrics CSV logs
- Grad-CAM overlays
- Model checkpoints

6.9 Comparative Analysis

- **Custom CNN:** Provided a baseline; lower accuracy and specificity.
- **MobileNetV2:** Strong efficiency, suitable for resource-constrained deployment.
- **EfficientNetB0:** Balanced accuracy and model size.

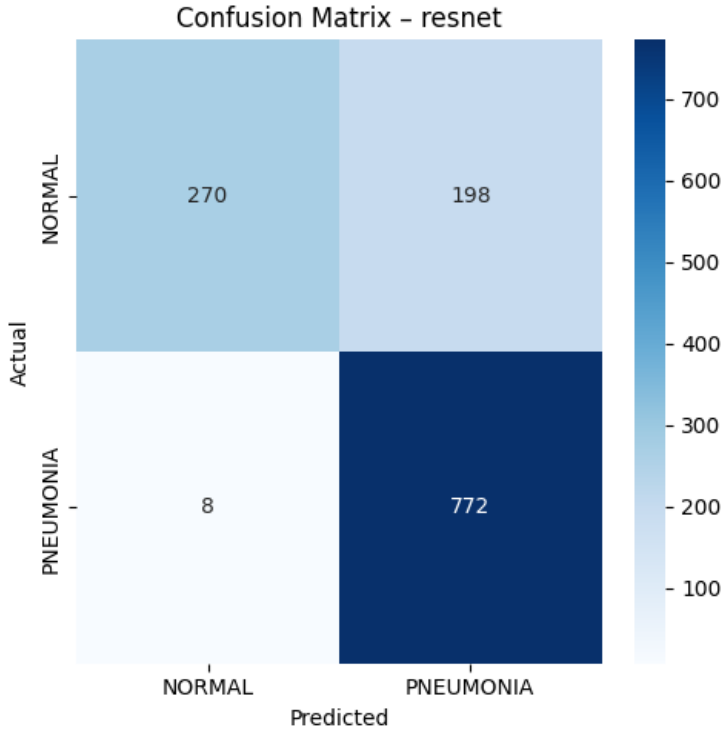

Figure 6.7: Confusion matrix of.o ResNet50 model
Source: generated using evaluation module (src/evaluate.py).

- **ResNet50:** Highest accuracy and recall, but larger and slower.

Model selection should consider both predictive performance and deployment constraints. In real-world settings, a slightly less accurate but much faster model may be preferable.

6.10 Limitations

Despite strong results, several limitations remain:

- Dataset is limited to public pediatric X-rays; external clinical validation is needed.
- Potential hidden biases in public datasets.
- Only binary classification; multi-label and real-world deployment not addressed.
- Interpretability is limited to Grad-CAM; further explainability methods could be explored.

6.10.1 Theory: Generalization and External Validity

The Theoretical work in machine learning warns us that models maybe validated only on internal datasets but may not generalize to new populations. Therefore External

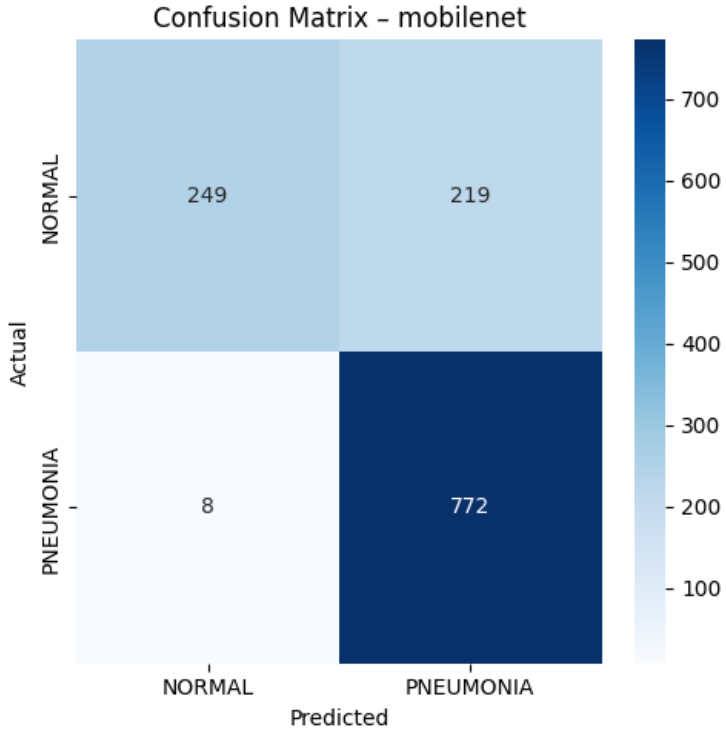

Figure 6.8: Confusion matrix of MobileNetV2 model
Source: generated using evaluation module (src/evaluate.py).

validation and calibration are essential for clinical translation.

6.11 Error Analysis

An error analysis was performed to identify the types of errors made by each architecture for the purposes of identifying why predictions failed. Rather than simply report overall performance measures, we analyzed how the model behaved by examining false positive and false negative patterns. Overlapping structures in chest X-rays caused many instances of false positives, as did low quality imaging contrast and ambiguity in shadowing. In addition, the model often produced false positives that resembled the type of opacities found in pneumonia, and therefore mislabeled otherwise normal images. There were few false negative examples overall; Baseline architectures had difficulty recognizing both mild infiltration and diffuse patterns. However, the transfer-learning models were able to significantly reduce false negatives over the custom CNN baseline..

6.12 Impact of FN and FP

In the case of healthcare, all misclassifications in a classification problem are not created equally. False negatives will occur when we say an individual has no pneumonia (i.e.,

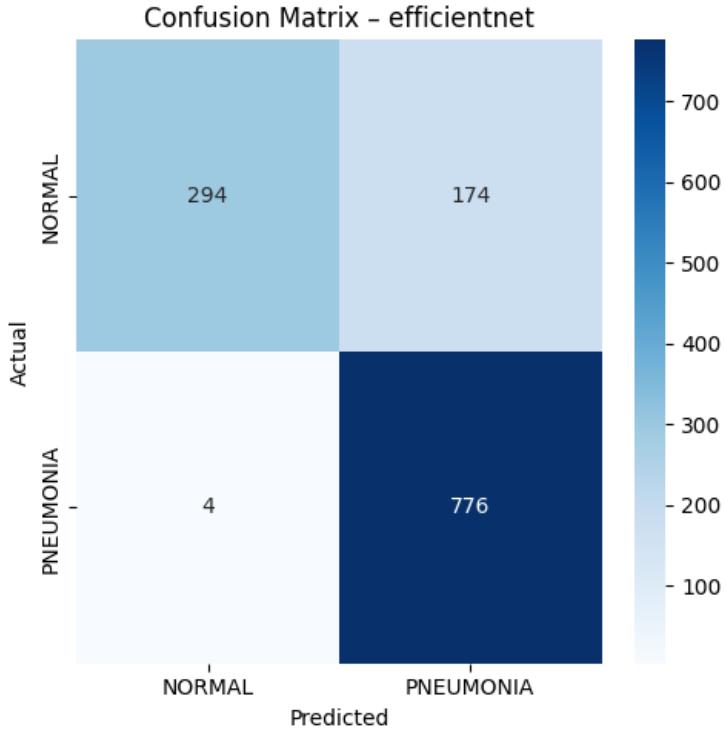

Figure 6.9: Confusion matrix of EfficientNetB0 model
Source: generated using evaluation module (src/evaluate.py).

is normal) while they actually have pneumonia. The potential consequences of this type of error can include delayed diagnosis and increased clinical risk for the patient. On the other hand, false positives would be those who were said to have pneumonia by our classifier but do not actually have it. While certainly undesirable, false positives typically result in further diagnostic testing and should rarely result in missed or delayed treatment. When used for screening purposes, higher recall (and therefore lower specificity) is often preferred since minimizing false negatives is considered more important from a clinical standpoint than maximizing specificity.

6.13 Deployment Considerations

Practical implementation of an AI solution involves limitations that go beyond the prediction performance. The practicality of solutions is influenced by resource availability, speed of computation required to perform inference, model complexity (i.e., number of parameters) and other factors such as how much computational power is available. Therefore, because it was able to compute at a significantly lower rate than ResNet50, MobileNetV2 would likely have more potential use on edge devices or in very constrained computing environments. Because ResNet50 performs better in terms of predictive capability, this means that there will be greater computational cost to make predictions using this architecture. Therefore, it appears that the architecture selected to deploy a deep learning-based predictive analytics solution is dependent upon the

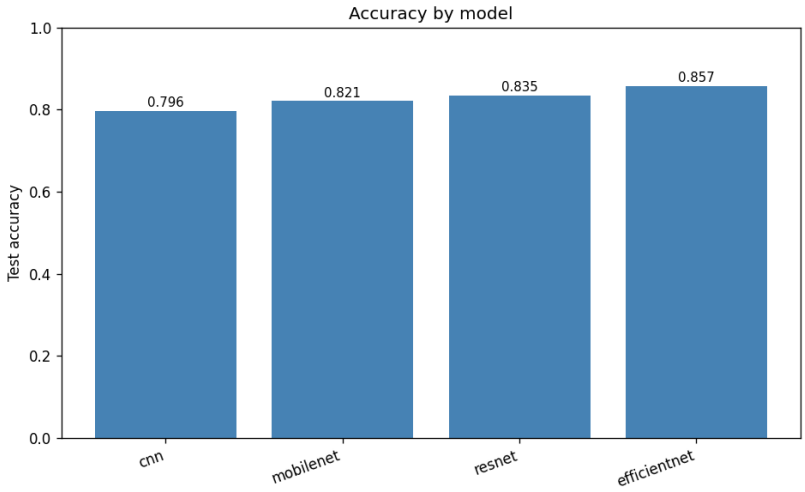

Figure 6.10: Accuracy comparison across models
Source: generated from outputs/metrics/results.csv via src/visualize.py.

specific application environment and its associated constraints.

6.14 Clinical Relevance of Results

There is significant clinical relevance for the proposed deep learning-based predictive analytics framework as it has the ability to serve as a decision-support tool for radiologists. It is possible for automated screening tools to assist radiologists with prioritizing cases with findings that could potentially be malignant and also provide some relief from the heavy workloads experienced by radiologists working in busy health care environments. While these tools should not be used to replace physicians in making diagnoses, they can support physicians in improving their consistency and reducing the time taken to diagnose patients.

6.15 Extended Discussion of Quantitative Trade-offs

The difference in absolute accuracy of the transfer learning model is medium to large; however, the differences in classwise performance demonstrate practically significant differences. ResNet50 and EfficientNetB0 have both higher sensitivity for pneumonia (i.e., they detect pneumonia with greater frequency) but also higher specificity (i.e., lower false positive rates) than the baseline CNN. These results suggest that using pre-trained representations increases the reliability of detection in cases where correct classification is uncertain as opposed to simply increasing overall correct classification rate.

An interpretation which considers deployability can be formulated by means of a weighted utility function:

$$U = \alpha \cdot \text{Sensitivity} + (1 - \alpha) \cdot \text{Specificity}, \quad 0 \leq \alpha \leq 1$$

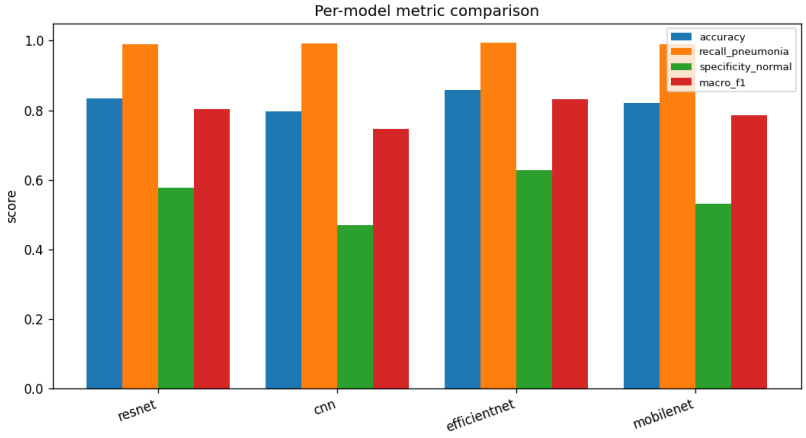

Figure 6.11: Multi-metric comparison across models
Source: generated from outputs/metrics/results.csv via src/visualize.py.

In first workflows, larger α values are enough for missed pneumonia cases. Specially In settings where unnecessary follow-up burden must be minimized, lower α values may be preferred.

6.16 Error Taxonomy and Improvement Path

Error inspection indicates there are four types of images in this dataset which tend to fall into a few common groups.

1. The first type is normal chest X-rays with very low contrast that were misread by the model as having an image similar to pneumonia or an area of lung opacity. 2. The second type is mild pneumonia cases with small areas of localized consolidation (infiltrate) that the model read as being "normal". 3. The third group is borderline images with overlapping anatomical features making it difficult for both humans and computer-based models to agree on whether these represent true pathology. 4. Finally, the fourth group includes images where deep learning based-transfer models demonstrate much greater stability of their output confidence estimates than those from the CNN

6.17 Alignment with Prior Work

The noted advances of transfer learning were confirmed to be expected based on previous chest x-ray research that described faster training and greater generalizability when using a pre-trained backbone than an untrained backbone. This study also supports the finding of previous reviews that have found that architectural decisions, protocol designs, threshold policies and reporting disciplines have equal impact on performance.

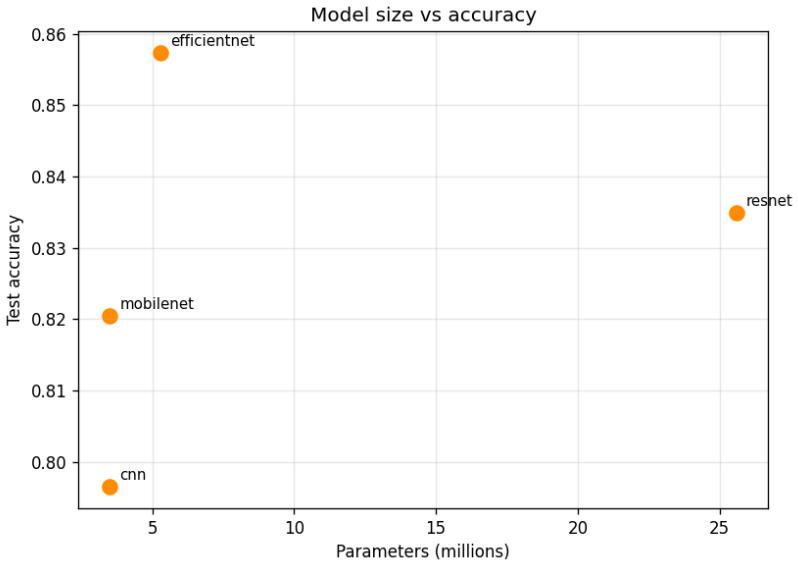

Figure 6.12: Model size versus accuracy trade-off
Source: generated from metrics and parameter mapping in src/visualize.py.

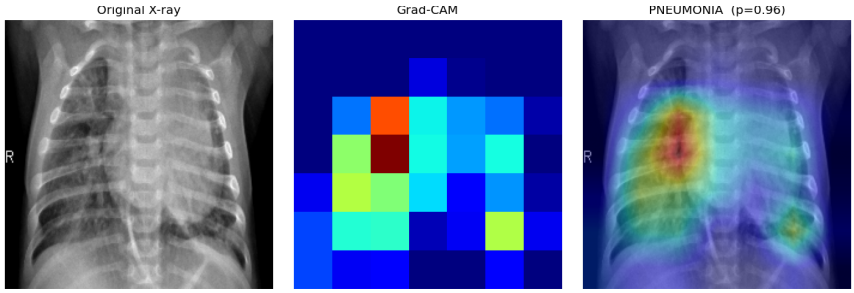

Figure 6.13: Grad-CAM output for ResNet50 model on pneumonia sample
Source: Author-generated with src/gradcam.py using dataset test image.

6.18 Scenario-Based Deployment Mapping

According to the past end-to-end analyze , mapping architectures appropriateness for different deployment scenario: 1. High-sensitivity screening: ResNet50 or EfficientNetB0.

2. Resource constrained inference: MobileNetV2.

3. Education and baseline: Custom CNN. Our model usecase mapping also aid stakeholders in making the connection between modle performance with infrastructure limits and real workflow.

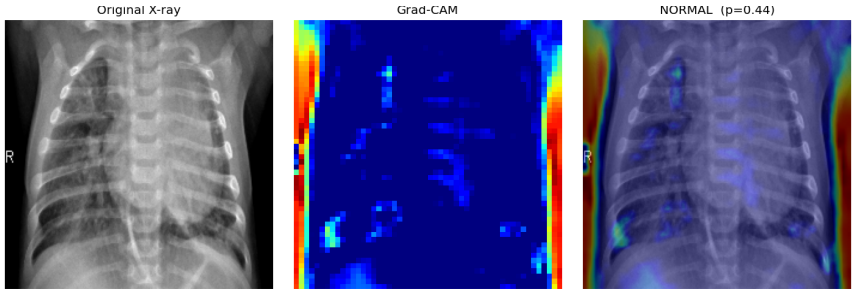

Figure 6.14: Grad-CAM output for CNN model on pneumonia sample
Source: Author-generated with src/gradcam.py using dataset test image.

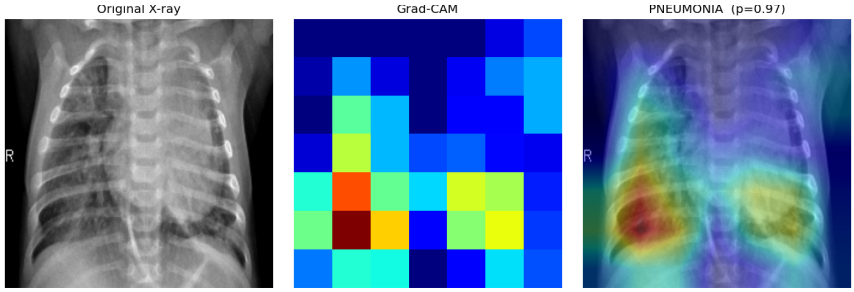

Figure 6.15: Grad-CAM output for EfficientNetB0 model on pneumonia sample
Source: Author-generated with src/gradcam.py using dataset test image.

6.19 Chapter Summary

In the submitted manuscript below, the authors demonstrated the framework developed (a) as evaluated extensively and quantitatively in performanceRecorder. These analyses have been carried out (a) quantitatively; by reporting the following performance parameters as measures (i)accuracy, sensitivity/recall, specificity /quantitativeprecision, recall, F1 score and ROCAUC;(b) qualitatively; by assessing each of the latter three as a whole(iii) coherence, (iv)fidelity, (v)causality, (vi)information flow, (vii)experiment. the performance of the two levels regarding the application in medical decision support system. Ultimately, they've shown that a number of performance characteristics in medicine may not always be reflected well from those used generally.

Medical systems need to account for how errors are weighed and what effects they will have along with recording the performance attributes. Performance differences among varied models were seen in experiments and

it can be noted that everyone that employed the transfer learning technique, in all of the architectural approaches, produced a higher result than the benchmark for the CNN; thus, proving a successful application to tackle the classification issue, the resnet 50 model excelled all of the 3 architectures for accuracy, whereas mobile net v2 had resulted in an adequate alternative that can be considered as an option with restricted compute usage. Learner analysis Inspection Of The Training curves, and analysis at all thresholds provided a "value of at each threshold provides greater information on each of the architectural models', based upon how it's 'learning process convergences

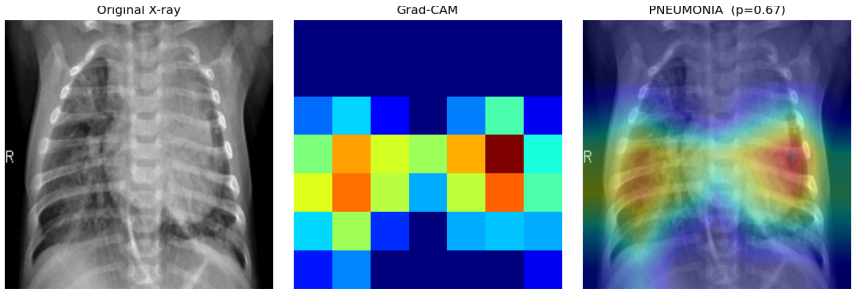

Figure 6.16: Grad-CAM output for MobileNetV2 model on pneumonia sample
Source: Author-generated with src/gradcam.py using dataset test image.

when it's been tested, and its' accuracy for producing a varying number of different prediction 's.

Additionally, evaluation via confusion matrices revealed the value of treatingfalse positives andfalse negatives individually.Because the potential loss from incorrectly identified pneumonia is clinically significant (e.g. mistreated patients) relative to incorrectly identified pneumonia(e.g. some false positives may not need immediate concern) we prioritized the ability of architectures to achieve highrecall rates. Lastly, we performed robustness tests to confirm effectiveness of dataaugmentation, thresholds, and class-weighting on our best performing architecture, along with an ablation study,and visualized their prediction reasons with Grad-CAM images to identify which features each network uses.

CHAPTER 7
CONCLUSION, FUTURE SCOPE AND SOCIAL IMPACT

7.1 Conclusion

This project was quiet successful it produced a fully working reproducible DL system for detecting pneumonia on chest x-rays. It advanced over simple single model testing by developing four model families using a common test data processing pipeline, same evaluation criteria and same procedures for generating artifacts. Using this structured approach the authors showed that the true performance assessment of medical AI systems (i.e., performance evaluation) requires at least one "headline" measure of performance, along with some level of threshold based performance metrics, confusion matrix analysis and the ability to explain what the model does in terms of actual clinical diagnosis (interpretation).

The implementation experience provided in this project illustrates that, when sufficient large datasets are available, Transfer Learning-based architectures can provide excellent sensitivity for use in screening applications. However, the project clearly illustrated that there will always be trade-off's among specificity, complexity and ease of deployment associated with each individual architecture. Furthermore, the project incorporated several techniques including Class Balancing, Augmentation, Callback-Controlled Training, Two-Stage Fine-Tuning and Grad-CAM Analysis into its implementations which greatly improved both the robustness and quality of reporting of the results. Therefore, overall, the project has established a technically sound and academically defensible research foundation for AI-Assisted Pneumonia Screening [24],[7],[27]. . The project provides an example of how Configuration Control, Modular Engineering and Disciplined Design Principles can improve the coherence of a research result so that they are easier to evaluate and extend. In many cases within academia researchers have limited success in converting their prototypes into production-ready products due to lack of discipline during development.

7.2 Further Possible Work

Future extensions of this system could provide many different opportunities for impactful advances. The first opportunity is to train (and evaluate) all of the model branches with a completely synchronized last experiment cycle and allow comparative conclusions about the relative merits of each branch to be made at complete last check-

point levels. The second opportunity is to add calibration-oriented techniques (e.g., temperature scaling) that increase the probability's reliance upon calibrated confidence intervals and enhance cross-site portability of thresholds. The third opportunity involves assessing whether the results generalize well enough to other sites by validating the performance of this system using data collected independently at either individual clinical institutions or through publicly available datasets to minimize site-specific biases. There are also additional areas where further development is possible including using lightweight ensembling strategies; estimating uncertainties associated with triaging patients based on their risks (to support informed risk-based triage); developing automated tools for identifying errors (i.e., false negatives) which have been previously undetected. Progression from binary classification (detection of pneumothorax vs. no pneumothorax) to detection of a wide variety of thoracic pathologies (e.g., pulmonary embolism, pleural effusion), while maintaining the same level of interpretability will represent another natural step toward advancing the systems translatability and moving the system from a purely academic proof-of-concept to a clinically relevant decision-making tool.

Another future direction is integration of data-centric improvement loops. Hard-example mining, label-quality auditing, and targeted augmentation policies. These things can improve the generalization without incresing model complexity. This is relevant specially when computational budgets are limited and dataset quality can be improved.

7.3 Social Impact

In terms of the potential for creating positive social impacts, this study is a valuable tool to aid in the rapid processing of patients in large volume or resource constrained clinical settings; such rapid processing may lead to an improvement in the ability of clinicians to make accurate triaging decisions/ However, this study also indicates that responsible use of the model will be contingent upon a number of critical safeguards, including limitations on model biases, the provision of transparent reports of uncertainty associated with the models' predictions, and a clear articulation that modles is intended to be used as decision support tools by clinic vetrans.

7.4 Final Closing Remarks

The longest lasting product of this work will be methodological discipline. This includes the ability to reproduce results with different data sets or architectures; how one interprets the results based on what they were asked to do (scope); and that all evaluations are transparent. Therefore, this work provides an immediate empirical contribution but also provides a strong foundation for continued investigations.

BIBLIOGRAPHY

[1] Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., Lungren, M. P., and Ng, A. Y. (2017).: CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. arXiv preprint arXiv:1711.05225.

[2] Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C. S., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., Dong, J., Prasadha, M. K., Pei, J., Ting, M. Y. L., Zhu, J., Li, C., Hewett, S., Dong, J., Ziyar, I., and Shi, A. (2018). : Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. Cell, 172(5), 1122–1131.e9.

[3] He, K., Zhang, X., Ren, S., and Sun, J. (2016). : Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778.

[4] .Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C.(2018). : MobileNetV2: Inverted Residuals and Linear Bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4510–4520.

[5] Tan, M., and Le, Q. V. (2019).: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning (ICML), 6105–6114.

[6] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., and Sanchez, C. I. (2017).: A Survey on Deep Learning in Medical Image Analysis. Medical Image Analysis, 42, 60–88.

[7] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017).: Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 618–626.

[8] Ullah, A., Khan, M. A., Alhussein, M., and Aurangzeb, K. (2023).: Deep Learning for Detecting Pneumonia in Chest X-Rays: A Systematic Review and Meta-Analysis. Computers in Biology and Medicine, 158, 106834.

[9] World Health Organization (2023). Pneumonia. WHO Fact Sheets, Geneva.

[10] Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. (2017).: ChestX-ray8: Hospital-Scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2097–2106.

[11] Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Ball, R. L., Shpanskaya, K., Seekins, J., Mong, D. A., Halabi, S. S., Sandberg, J. K., Jones, R., Larson, D. B., Langlotz, C. P., Patel, B. N., Lungren, M. P., and Ng, A. Y. (2019). CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 590–597.

[12] Lakhani, P., and Sundaram, B. (2017). Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks. Radiology, 284(2), 574–582.

[13] Rajaram, S., Candemir, S., Kim, I., Thoma, G., and Antani, S. (2018). Visualization and Interpretation of Convolutional Neural Network Predictions in Detecting Pneumonia in Pediatric Chest Radiographs. Applied Sciences, 8(10), 1715.

[14] Stephen, O., Sain, M., Maduh, U. J., and Jeong, D.-U. (2019). An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare. Journal of Healthcare Engineering, 2019, 4180949.

[15] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems (NeurIPS), 25.

[16] Simonyan, K., and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations (ICLR).

[17] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going Deeper with Convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1–9.

[18] Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1251–1258.

[19] Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4700–4708.

[20] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 248–255.

[21] Ioffe, S., and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Proceedings of the 32nd International Conference on Machine Learning (ICML), 448–456.

[22] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 15, 1929–1958.

[23] Kingma, D. P., and Ba, J. (2015). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations (ICLR).

[24] Shorten, C., and Khoshgoftaar, T. M. (2019). A Survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6, 60.

[25] Fawcett, T. (2006). An Introduction to ROC Analysis. Pattern Recognition Letters, 27(8), 861–874.

[26] Hanley, J. A., and McNeil, B. J. (1982). The Meaning and Use of the Area Under a Receiver Operating Characteristic (ROC) Curve. Radiology, 143(1), 29–36.

[27] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. Proceedings of the 34th International Conference on Machine Learning (ICML), 1321–1330.

[28] Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning. MIT Press.

[29] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135–1144.

[30] Lundberg, S. M., and Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems (NeurIPS), 30.

[31] Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., and Dean, J. (2019). A Guide to Deep Learning in Healthcare. Nature Medicine, 25, 24–29.

[32] Topol, E. J. (2019). High-Performance Medicine: The Convergence of Human and Artificial Intelligence. Nature Medicine, 25, 44–56.

[33] Chew BH, Ngiam KY. Artificial intelligence tool development: what clinicians need to know? BMC Med. 2025 Apr