الجمـــهوريـــة الجـــزائريـــة الديـمقـراطيـــــــة الشـعبيـــــة
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
وزارة التعـلـيـم الـعـالـــي و الـبـحـث الـعـلـمي
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
جـامـعـة عمّـــار ثـلـيـــجـــي بالأغـــواط
AMAR TELIDJI UNIVERSITY OF LAGHOUAT

كـلـيـــــة الـعـلـــــوم
FACULTY OF SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

Master's Thesis

Field: **Mathematics and Computer Science**
Specialty: **Computer Science**
Option: **Data Science and Artificial Intelligence**

By:
Derouiche Sarra **Merrad Asmaa**

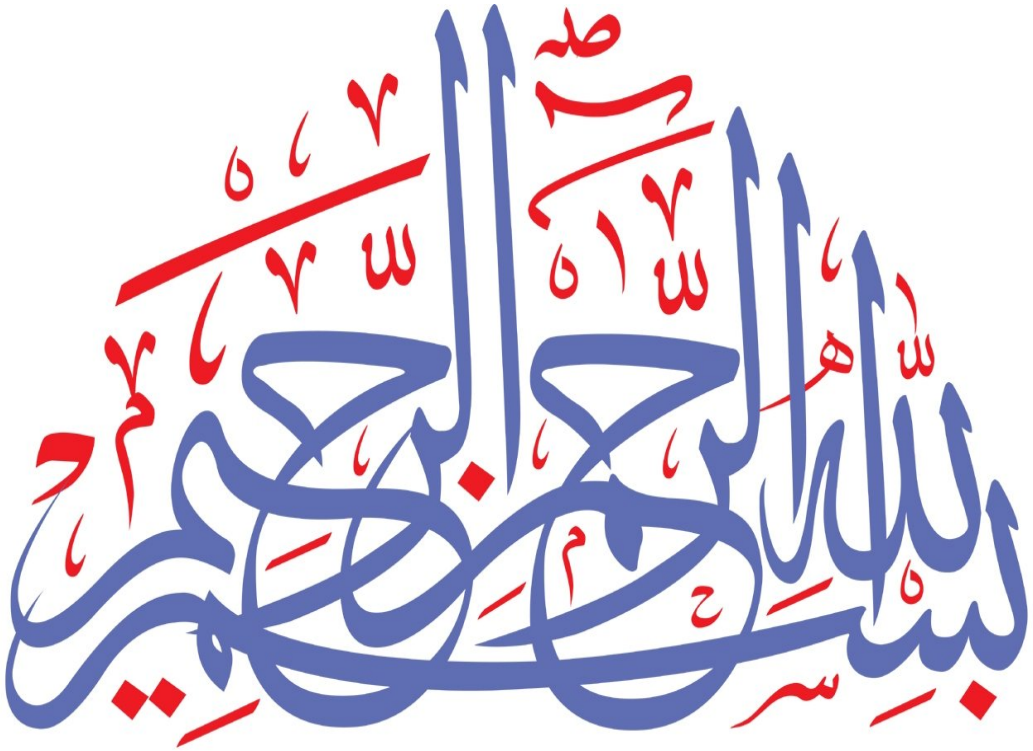
TOPIC

Evaluating the Impact of Region of Interest Detection Methods on Medical Image Classification

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

Pr. Nasreddine Lagraa	Prof	President
Dr. Saida Sarra Boudouh	M.A.B	Examiner
Dr. Mohamed El Habib Maicha	M.C.B	Examiner
Dr. Younes Guellouma	M.C.A	Supervisor

Thesis No. …… – Academic Year 2024/2025


بسم الله الرحمن الرحيم

Acknowledgments

First and foremost, we would like to express our deepest gratitude to Allah (Elhamdouli'Allah) for granting us the strength, perseverance, and capability to successfully complete this thesis. We extend our sincere thanks to our advisor, Dr. Younes Guellouma, for his continuous guidance, encouragement, and for providing us with a supportive research environment.

We are also truly grateful to Prof. Chaker Abdelaziz Kerrache for his invaluable assistance in facilitating our work and providing access to the necessary equipment and resources.

Special thanks to Dr. Saida Sarra Boudouh, our Deep Learning instructor, for delivering a valuable and inspiring course. Her guidance, encouragement, and commitment to excellence greatly enriched our learning experience and motivated us to delve deeper into the field.

We also extend our sincere thanks to Cheknane Maroua for her insightful advice and helpful feedback throughout this work.

Our deepest gratitude goes to our parents, who have been our first and most constant supporters. Their endless love, prayers, sacrifices, and encouragement have been the foundation that allowed us to reach this point.

We also wish to express our appreciation to all our professors and colleagues at the LIM Laboratory and the Department of Computer Science at Laghouat University, whose support and collaboration have been instrumental to this work.

We would like to sincerely thank the jury members for taking the time to review and evaluate our work. Their valuable comments, constructive feedback, and thoughtful suggestions have greatly contributed to the improvement and refinement of this thesis.

Dedications

I dedicate this work to:
My dear grandparents — your prayers, warmth, and love have always guided me.
To my wonderful parents — thank you for your unconditional support, your sacrifices, and your constant belief in me. You are my greatest source of strength.
To my beloved brother, Moustapha — you have always been my support and protector. I'm endlessly grateful for you.
To my precious cousins, Noussaiba, Soumia, and Aicha Djoual — you are more than family; you are a part of my soul. Your love means the world to me.
To my lovely Binom, who has been with me in every step, supporting me, helping me grow into a better person. She has been not just a university colleague, but the sister I never had. Thank you from the bottom of my heart. May Allah gather us together in Jannah (Heaven).
And to my amazing friends from university — you were one of the best parts of my journey. Your friendship gave me strength, joy, and unforgettable memories. Sarra , Souad , Intissar , Abir , Maroua ,Fatiha - thank you for being there, for believing in me, and for walking this path with me.
thank you all.

Asma Merrad

Dedications

I dedicate this work to:

To the ***heroic engineers of the 7th of October*** *and the brave people of* ***Gaza*** *— your strength inspires us. I pray that this work, even if it is small, serves our Ummah. I hope our knowledge strengthens us against all enemies. May we witness the day when* ***Palestine is free****, and stand in prayer at* ***Al-Aqsa Mosque****.*

To my dear ***parents*** *— thank you for your love, your sacrifices, and for teaching me to work hard and stay true to my values.*

To my siblings, ***Fatima, Safaa, and Youcef*** *— your presence means the world to me.*

To my ***family****, both those still with us and those who have passed — your love and prayers are always with me.*

To the people I love most — ***Raihana, Fatima, Sarra, and Khadidja*** *— your presence in my life is a blessing I will always be grateful for.*

To the ones who Allah placed in my life to make this university journey brighter — ***Souad, Asmaa, Intissar, and Abeer*** *— words will never be enough to thank you.*

To my dear ***binôme*** *— thank you for your support and for standing by me through every step of this journey. You became more than a teammate — you became family. You lifted me up when I needed it most, and I'm truly thankful that Allah placed you in my life.*

To all my friends, I am so thankful to have you.

To all my ***teachers*** *along the way — thank you for shaping my path and helping me grow.*

To my dearest ***professor*** *— I am honored to have learned from you. May Allah reward your guidance and kindness.*

To the one who stood by me during my sleepless nights, who helped me stay strong, and reminded me to keep going even when I felt tired, my gratitude could never be enough.

I pray that Allah reunites us in Jannah.

Sarra Derouiche

ملخص :

لا تزال مهمة تصنيف الصور الطبية تمثل تحدياً كبيراً بسبب الطبيعة الدقيقة والمتنوعة لأنماط الأمراض عبر أنماط التصوير المختلفة. تقدّم نماذج التعلم العميق حلولاً واعدة؛ إلا أنّ دمج عملية اكتشاف مناطق الاهتمام (ROI) ضمن عملية التدريب لا يزال غير مفهوم بشكل كامل. تستكشف هذه الرسالة فعالية تقنية Grad-CAM كطريقة غير خاضعة للإشراف لاكتشاف مناطق الاهتمام ضمن إطار عمل ذي مرحلتين. في المرحلة الأولى، يُستخدم Grad-CAM لتوليد صور مركّزة على مناطق الاهتمام من صور الأشعة السينية للصدر وصور الرنين المغناطيسي للدماغ، دون الحاجة إلى ترميزات على مستوى البكسل. أما في المرحلة الثانية، فنقوم بتدريب ومقارنة نماذج تصنيف عميقة باستخدام كلّ من هذه الصور القائمة على مناطق الاهتمام والصور الأصلية الكاملة. تتكوّن البنية المعمارية للنموذج من شبكة عصبية التفافية مسبقة التدريب EfficientNetB4) أو DenseNet201) ، مع رأس تصنيف مخصص، واستراتيجيتين لضبط الأوزان: تجميد الطبقات بالكامل أو تجميد جزئي (25% من الطبقات العلوية قابلة للتدريب). أظهرت النتائج أنّ استخدام الصور الكاملة يتفوق باستمرار على النسخ المحوّلة القائمة على مناطق الاهتمام، حيث حقق نموذج DenseNet201 مع الضبط الجزئي أعلى دقة (98.00% في صور أشعة الصدر، و99.00% في صور الرنين المغناطيسي للدماغ). تشير هذه النتائج إلى أنّه على الرغم من قيمة Grad-CAM في التفسير البصري، إلا أنه قد لا يكون أداة فعالة لاكتشاف مناطق الاهتمام بطريقة غير خاضعة للإشراف أثناء التدريب، إذ قد يستبعد إشارات سياقية ضرورية للتعلم القوي.

الكلمات المفتاحية: تصنيف الصور الطبية، منطقة الاهتمام(ROI) ، تقنية Grad-CAM، التعلم العميق، أشعة الصدر السينية، الرنين المغناطيسي للدماغ

Abstract

Medical image classification remains a challenging task due to the subtle and varied nature of disease patterns across imaging modalities. Deep learning models offer promising solutions; however, the integration of region-of-interest (ROI) detection into the training process is still not well understood.This thesis explores the effectiveness of Grad-CAM as an unsupervised ROI detection method within a two-phase framework. In Phase 1, Grad-CAM is used to generate ROI-focused images from chest X-rays and brain MRIs without requiring pixel-level annotations. In Phase 2, we train and compare deep classification models using both these ROI-based inputs and the original full images. The architecture consists of a pretrained convolutional backbone (EfficientNetB4 or DenseNet201), a custom classification head, and two fine-tuning strategies: frozen and partially unfrozen (top 25 % trainable layers). Results show that full-image inputs consistently outperform ROI-transformed versions, with DenseNet201 and partial unfreezing achieving the highest accuracy (98.00 % on chest X-rays, 99.00 % on brain MRIs). These findings indicate that while Grad-CAM is valuable for visual interpretation, it may not serve as an effective unsupervised ROI detector during training, as it may exclude contextual cues critical for robust learning.

Keywords : **Medical Image Classification, Region of Interest (ROI), Grad-CAM, Deep Learning, Chest X-ray, Brain MRI**

Contents

List of Figures

List of Tables

List of Acronyms

AI Artificial Intelligence. 4, 5, 9, 18, 20, 21, 67

AUC Area Under Curve. 27

CAM Class Activation Mapping. 15, 16, 17, 30

CNN Convolutional Neural Network. 4, 7, 8, 9, 15, 16, 17, 18, 28, 30, 33, 39, 44

CT Computed Tomography. 31, 32, 67

DL Deep Learning. 4, 6, 7, 10, 18

DWT Discrete Wavelet Transform. 22

FFT Fast Fourier Transform. 22

FN False Negative. 11, 13

FP False Positive. 11, 13

FPN Feature Pyramid Network. 20, 24

GAP Global Average Pooling. 15, 16

GLCM Gray-Level Co-occurrence Matrix. 22

GLDM Gray-Level Difference Matrix. 22

Grad-CAM Gradient-weighted Class Activation Mapping. 4, 15, 16, 17, 18, 19, 26, 27, 30, 31, 32, 33, 34, 35, 36, 39, 43, 50, 51, 52, 53, 56, 57, 62, 63, 64, 65, 66, 67

KNN k-Nearest Neighbour. 7, 26

LCAM Lung Class Activation Map. 27, 28

MAE Mean Absolute Error. 32

ML Machine Learning. 4, 5, 6, 7, 9, 18

MRI Magnetic Resonance Imaging. 10, 31, 32, 36, 37, 43, 50, 52, 53, 56, 57, 60, 62, 63, 64, 65, 66

NAS Neural Architecture Search. 40

NN Neural Network. 7, 9

PPV Positive Predictive Value. 13, 47

R-CNN Region-based CNN. 19

RFMiD Retinal Fundus Multi-Disease Image Dataset. 11, 31

RIS RadiologyInformation System. 20

ROI Region of Interest. 9, 19, 20, 27, 28, 30, 34, 35, 36, 38, 39, 43, 44, 47, 50, 51, 52, 53, 54, 56, 57, 59, 60, 62, 63, 64, 65, 66, 67

RPN Region Proposal Network. 20, 24

SSIM Structural Similarity Index. 32

SVM Support Vector Machine. 7, 28

TDR True Discovery Rate. 13

TL Transfer learning. 10

TN True Negative. 11, 13

TNR True Negative Rate. 13

TP True Positive. 11, 13

TPR True Positive Rate. 13, 47

VIT Vision Transformer. 9, 67

1 Introduction

Contents

1.1 Context

Healthcare systems around the world face increasing challenges in diagnosing and managing complex diseases such as cancer, respiratory illnesses, and neurological disorders. Early and accurate detection plays a vital role in improving patient outcomes, reducing treatment costs, and saving lives. Conditions like brain tumors and chest infections can progress rapidly, making timely diagnosis essential. Identifying these diseases early often relies on thorough clinical evaluations supported by imaging technologies.

Medical imaging has become a core part of modern clinical practice, enabling non-invasive visualization of internal body structures. Techniques such as X-rays and magnetic resonance imaging (MRI) help detect abnormalities like tumors, lesions, or unusual tissue growth. For example, brain MRI scans are commonly used to identify tumors such as gliomas or meningiomas, while chest X-rays assist in diagnosing lung infections and other thoracic conditions. As the volume of medical imaging continues to grow, there is increasing interest in using computer-based tools to assist in faster, more consistent, and more accurate image analysis.

While the diagnostic utility of these techniques is well established, analyzing medical images remains complex due to their high dimensionality, modality-specific features, and variability introduced by different acquisition protocols. Among the most pressing challenges in medical image analysis is the accurate and efficient identification of Regions of Interest (ROIs) localized areas containing diagnostically significant information such as tumors, lesions, or structural abnormalities. Traditionally, this task has depended on the expert judgment of radiologists, who manually identify ROI regions during visual inspection. However, this manual process is time-consuming, prone to inter-observer variability, and increasingly unsustainable in the face of rapidly growing imaging volumes.

In recent years, many *supervised* ROI detection methods—such as segmentation networks and object detection models—have shown strong performance. These methods typically rely on

annotated data to learn where diagnostically relevant regions are located, and they have been widely validated across various medical tasks. In contrast, *unsupervised* ROI detection techniques remain poorly explored and under-evaluated, especially in terms of how they influence classification performance.

This thesis is motivated by the hypothesis that explicitly detecting and utilizing ROIs during preprocessing may impact the performance of medical image classification systems. Specifically, we aim to evaluate whether ROI-guided classification using Grad-CAM as an unsupervised method for ROI detection can offer meaningful improvements. Grad-CAM has primarily been employed as a post-hoc visualization tool to explain model predictions, but its effectiveness as a standalone mechanism for unsupervised ROI extraction—and its impact on downstream classification accuracy—has not been sufficiently studied.

In this work, Grad-CAM is used not for interpretability, but rather as an *unsupervised* tool to extract candidate ROIs, which are then used to train classification models. This allows us to evaluate whether isolating the regions that the model already finds important can lead to performance improvements, even without any labeled supervision during ROI detection.

1.1.1 Research Problem and Research Question

Although deep learning models perform well in medical imaging tasks, there has been limited research on how tools like Grad-CAM can be used not just to interpret model predictions but to actively influence and potentially improve model training. Grad-CAM highlights the regions in an image that most influence the model's decision, helping to identify important areas such as tumors or lesions.

While supervised ROI methods have been thoroughly studied, unsupervised ROI extraction especially through methods like Grad-CAM—has not been rigorously evaluated in terms of its practical utility for improving classification performance.

This research explores whether using ROI regions generated by Grad-CAM can improve the training and testing of classification models. In other words, we aim to find out if focusing only on the most relevant parts of an image—those highlighted by Grad-CAM—can lead to better model performance, despite being extracted without supervision.

Research Question

> *Does using Grad-CAM to detect Region of Interest (ROI) areas improve the performance of deep learning models in medical image classification tasks?*

To answer this question, the work is divided into two phases:

- **Phase 1: ROI Detection** — Use Grad-CAM to generate ROI heatmaps without any supervision or labels.
- **Phase 2: Evaluation** — Train and test classification models using both original and ROI-transformed images, with and without fine-tuning.

1.2 Organization of the Thesis

The structure of this thesis is outlined as follows : **Chapter Two** presents background information and explains key concepts, including machine learning, deep learning, Region of

Interest (ROI) detection, and the Grad-CAM method. **Chapter Three** reviews related works by discussing previous researches and techniques used for ROI detection and image visualization in healthcare. **Chapter Four** describes the main contributions of this study, including data collection, experimental design, the methods used, and the results from both phases of the research. Finally, **Chapter Five** summarizes the findings, discusses the limitations of the study, and offers recommendations for future work.

2

Background

Contents

2.1 Introduction

In recent years, the integration of Artificial Intelligence (AI) into medical image analysis has grown significantly, driven by the advancement of Machine Learning (ML) and Deep Learning (DL) techniques. These computational approaches enable automated systems to learn from data, identify complex patterns, and support clinical decision-making with improved accuracy and efficiency.

This chapter provides a foundational overview of ML, its learning paradigms, and the evolution toward DL architectures, including Convolutional Neural Network (CNN)s that are particularly effective in processing visual data. Additionally, we explore key interpretability tools such as Gradient-weighted Class Activation Mapping (Grad-CAM), which help reveal the decision-making processes of deep networks, and discuss common classification performance metrics that are crucial for evaluating model effectiveness. By establishing this theoretical background, the chapter sets the stage for understanding the methods and technologies employed in the development of intelligent diagnostic systems.

2.2 Machine Learning

ML is a subfield of AI that focuses on developing algorithms and models that enable computers to learn from data rather than being explicitly programmed. For example, an ML model can be trained to identify spam emails by learning from a labeled dataset of examples. These datasets, known as training data, help the model identify patterns and relationships that it can later use to make predictions on new, unseen data.

Unlike traditional programming, where rules are predefined,ML systems adapt and improve their performance over time by refining their internal parameters based on experience. This capability is especially valuable in applications involving large, complex, or high-dimensional datasets. While all ML is considered part of AI, not all AI approaches rely on ML . Traditional symbolic AI, for instance, involves rule-based systems rather than data-driven learning.

ML models rely on mathematical and statistical techniques to perform tasks such as classification, regression, and clustering. By identifying patterns in past data, these models aim to make informed decisions or predictions that resemble human-like reasoning and decision-making [19, 20].

2.2.1 Paradigms of Machine Learning

ML approaches are commonly categorized into four main paradigms based on the availability and nature of labeled data. Each paradigm addresses different types of problems and leverages different algorithmic strategies. An overview of these paradigms is presented in Figure 2.1 [1].

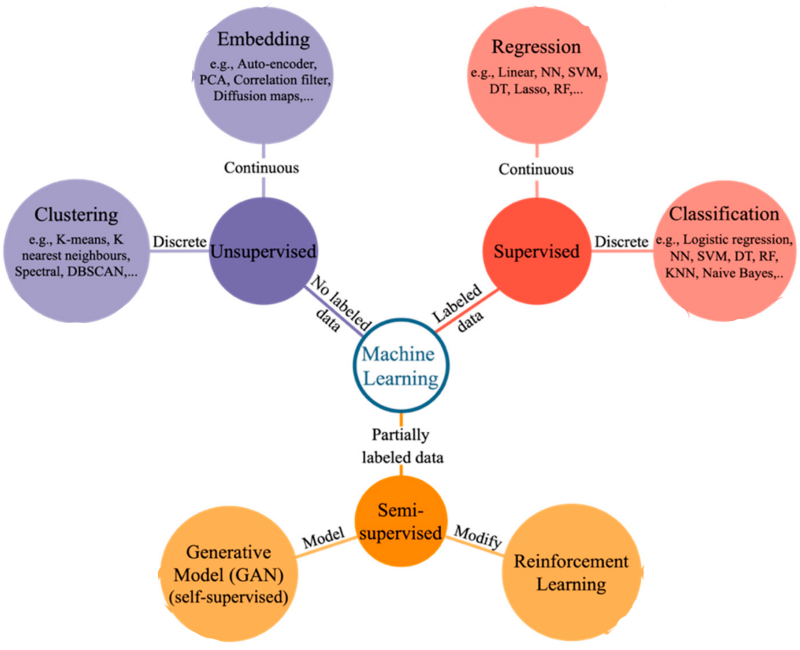


Figure 2.1: Paradigms of Machine Learning.[1]

Supervised Learning

Supervised learning involves training a model using labeled data, where both the input features and corresponding output labels are provided. The model learns to map inputs X to outputs Y, and once trained, it can generalize to predict outcomes for new inputs. This approach is analogous to learning with a teacher, where the correct answers are known during training [21].

Common applications include:

- **Classification:** Predicting discrete categories (e.g., spam vs. non-spam).
- **Regression:** Predicting continuous values (e.g., house prices).

The dataset is typically split into a training set and a test set. The training set is used to teach the model, while the test set is used to evaluate its performance.

Unsupervised Learning

Unsupervised learning deals with data that has no associated labels. The goal is to identify hidden patterns, groupings, or structures within the data. Unlike supervised learning, it does not require labeled examples and is useful for exploring unknown datasets.[22]

Common tasks include:

- **Clustering:** Grouping similar data points into clusters.
- **Dimensionality Reduction:** Reducing the number of input variables while preserving important information.
- **Association Mining:** Discovering relationships between variables (e.g., market basket analysis).

Semi-Supervised Learning

Semi-supervised learning falls between supervised and unsupervised learning. It uses a small amount of labeled data along with a large amount of unlabeled data to improve learning accuracy. This approach is particularly useful when labeled data is expensive or difficult to obtain, but unlabeled data is abundant [23].

Two common settings include:

- **Semi-supervised classification:** Uses both labeled and unlabeled data to improve classification accuracy.
- **Semi-supervised clustering:** Incorporates prior information such as labels or constraints into clustering tasks.

2.3 Deep Learning

DL is a specialized subfield of ML that focuses on the use of deep neural networks, which consist of multiple layers, to learn complex patterns directly from raw data. Unlike traditional

ML methods that rely heavily on manual feature engineering, deep learning models automatically extract relevant features during the training process. This ability makes DL especially effective in handling unstructured data, such as images, audio, and natural language [24].

By learning hierarchical representations, where higher-level features are built upon lower-level ones, DL achieves high performance in various domains such as image recognition, speech processing, and natural language understanding.

2.3.1 Differences Between ML and DL

While both ML and DL aim to develop models that learn from data to make predictions or decisions, they differ significantly in architecture and approach.

- **Feature Engineering:** Traditional ML models often require domain experts to manually design features, whereas DL models automatically learn these features during training.
- **Model Architecture:** ML includes a wide range of algorithms (e.g., decision trees, Support Vector Machine (SVM), k-Nearest Neighbour (KNN)) that are generally shallow, while DL relies on deep neural networks with multiple processing layers.
- **Data Requirements:** ML algorithms typically perform well on structured data and require less data, while DL models excel with large-scale unstructured data and need substantial datasets to perform effectively.
- **Performance:** DL generally outperforms ML in tasks such as image classification, speech recognition, and text generation, but may require more computational resources [25].

2.3.2 Convolutional Neural Networks

CNN are Neural Network (NN) created specifically to process images and videos.
A typical CNN architecture (Figure 2.2) consists of several key components:

- **Convolutional Layers:** These layers apply learnable filters (kernels) to extract local spatial features. The convolution operation for a 2D input image I with a filter K is defined as:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) \cdot K(m,n)$$

  where $S(i,j)$ is the resulting output called feature map, and the summation is performed over the dimensions of the kernel.[25]

- **Pooling Layers:** These layers reduce the spatial dimensions of feature maps, which helps lower computational cost and mitigate overfitting. A common pooling method is *max pooling*, which selects the maximum value within a defined window. The operation is defined as:

$$P(i,j) = \max_{(m,n)\in \text{window}} A(i+m, j+n)$$

  where:

- A denotes the input activation map,
- $P(i, j)$ is the pooled output at location (i, j),
- W is the set of integer index pairs (m, n) defining the pooling window, typically with $m \in \{0, \ldots, H-1\}$, $n \in \{0, \ldots, W-1\}$, where H and W are the height and width of the pooling window, respectively.

where A is the activation map input to the pooling layer, and $P(i, j)$ is the pooled output.

Max-Pooling: A max-pooling operator can be applied to down-sample the convolutional output maps by passing forward the maximum value within a group of R activations. The m-th max-pooled feature map is composed of J filters, represented as $\mathbf{p}_m = [p_{1,m}, p_{2,m}, \ldots, p_{J,m}] \in \mathbb{R}^J$, where each element is defined by:

$$p_{j,m} = \max\left(h_{j,(m-1)N+r}\right)$$

Here, $h_{j,(m-1)N+r}$ represents the activations within the pooling region indexed by j, and N and r denote the stride and position offset respectively.[26]

- **Fully connected layers:**These layers take the flattened feature maps or ponderated vectors in some cases and perform high-level reasoning. For a dense layer:

$$y = f(Wx + b)$$

where x is the input vector, W is the weight matrix, b is the bias vector, and f is a non-linear activation function.

- **Output layer:** To produce the final prediction in classification tasks, the softmax function is typically used to convert raw output scores into probabilities. However, it is important to note that softmax is not always used — for example, in binary classification tasks, a sigmoid function might be applied instead.The softmax function is defined as:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

Each layer transforms the input volume to an output volume of activations, which is then passed to the next layer. CNN have demonstrated exceptional performance in applications such as object detection, facial recognition, autonomous driving, and medical image analysis [27].

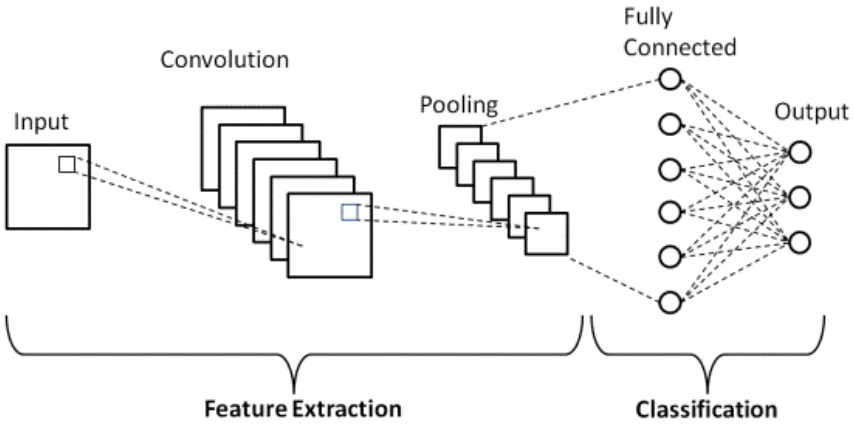


Figure 2.2: The CNN layers. [2]

2.3.3 Computer Vision

Computer vision is a subfield of AI that focuses on enabling machines to perceive, interpret, and understand visual information from the world. By utilizing ML techniques and NNs, computer vision systems are capable of analyzing digital images and video streams to extract meaningful patterns and insights. This capability allows automated systems to detect anomalies, recognize objects, make decisions, and perform tasks that traditionally required human visual perception.

Core Tasks in Computer Vision

The modern field of computer vision encompasses a diverse range of tasks, enabled by powerful AI models such as CNNs and Vision Transformer (VIT)s. These tasks include image classification, object detection, semantic segmentation, and emerging paradigms such as generative modeling and real-time inference. Below, we outline and discuss several key computer vision tasks, the models commonly used to address them, and their real-world applications.[28]

Image Classification

Image classification assigns a category label to an entire image and serves as a foundational task in computer vision. Models such as ResNet [29], VGG [30], and BLIP [31] have shown strong performance using deep architectures or vision-language integration. Applications span across healthcare (e.g., disease detection in medical scans), logistics (e.g., item categorization), and manufacturing (e.g., defect detection).

Object Detection and Localization

Object detection identifies and localizes multiple objects within an image using bounding boxes or masks. Key models include Faster R-CNN [32], YOLOv7 [33], and SSD [34]. These models are widely applied in real-time systems such as autonomous vehicles, surveillance, and industrial automation.

Semantic Segmentation

Semantic segmentation classifies each pixel of an image into a predefined category, offering detailed understanding useful in fields like medical imaging and autonomous navigation. Common models include FastFCN [35], DeepLab [36], and U-Net [37].

Instance Segmentation

Instance segmentation extends semantic segmentation by distinguishing between individual objects of the same class. Models like SAM [38] and Mask R-CNN [39] can separate multiple instances within a single image, which is critical for tasks requiring object-level granularity, such as medical diagnostics or traffic analysis.

Pose Estimation

Pose estimation involves detecting keypoints (e.g., joints) on objects or human figures, commonly used in gesture recognition, sports analytics, and human-computer interaction. Well-known models include OpenPose [40], MoveNet [41], and PoseNet [42], each optimized for different speed and accuracy needs.

2.3.4 Transfer Learning

Transfer learning (TL), is a method where a model trained on one task is reused or adapted for a different but related task. This approach is especially helpful in DL, where training a model from scratch often needs a large amount of labeled data and powerful computing resources. Instead of starting from zero, a model trained on a large dataset (such as ImageNet) can be adapted to a new task with fewer data and shorter training time [43]. TL generally follows two main strategies:

- **Feature extraction:** The pretrained model is used to extract useful features from the new data, and only the final classifier layer is trained.
- **Fine-tuning:** Fine-tuning works by using the weights from a pretrained model as a starting point. The model is then trained further on a smaller dataset related to the new task. This process can involve different types of learning, including supervised, self-supervised, semi-supervised, or reinforcement learning, depending on the available data and task requirements. For example, a model trained to classify general objects in images can be fine-tuned to recognize medical conditions in X-rays using a labeled medical dataset.

  Fine-tuning can be done in different ways like:

  - **Full fine-tuning:** Updates all the weights in the model. This method is powerful but computationally expensive and can risk losing previously learned general knowledge. To reduce this risk, hyperparameters such as the learning rate are often carefully adjusted.
  - **Partial fine-tuning:** Also known as selective fine-tuning, it updates only a part of the model, such as the final layers. Earlier layers, which usually capture general features like edges or textures, remain unchanged. This approach reduces training time and helps avoid overfitting.

– **Parameter-efficient fine-tuning (PEFT):** Reduces the number of parameters that need to be updated. Techniques include updating only specific weights or biases, which lowers memory and computational costs while maintaining performance.

In practice, the choice of fine-tuning method depends on the task, available data, and computing resources. Researchers often test various setups, datasets, and hyperparameters (e.g., learning rate, batch size) to find the best configuration for their application [43].

2.4 Classification Performance Metrics

This section discusses some metrics used to evaluate classification models. A summary of these metrics is provided in Table 2.2 with further details explained in the following subsections.

Confusion Matrix

The confusion matrix is an essential method for evaluating the performance of a binary classifier. It captures the frequency of each possible pairing between predicted and actual class labels. In binary classification—where the possible true labels are Positive, Negative—the confusion matrix is structured as a 2×2 table, as shown in Table 2.1.

Table 2.1: Confusion Matrix for Binary Classification [17]

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

Definitions

- **True Positive (TP)**: The model correctly classifies a positive instance as positive.
- **True Negative (TN)**: The model correctly classifies a negative instance as negative.
- **False Positive (FP)**: Also known as *Type I error*, the model incorrectly classifies a negative instance as positive (a "false alarm").
- **False Negative (FN)**: Also known as *Type II error*, the model incorrectly classifies a positive instance as negative (a "miss").[17]

Confusion Matrix in Multi-Class Classification

In classification problems involving N distinct classes, the confusion matrix is structured as an $N \times N$ table. In this matrix, each row i corresponds to the **actual class** i, and each column j represents the **predicted class** j produced by the model.

Let $M[i, j]$ represent the number of instances whose true label is class i, but which the model has classified as class j. Each entry in this matrix satisfies the following conditions:

- **Row Sum:**

$$\sum_{j=1}^{N} M[i,j] = \text{Total number of instances of class } i$$

- **Column Sum:**

$$\sum_{i=1}^{N} M[i,j] = \text{Total number of instances predicted as class } j$$

- **Diagonal Entries:**

$$M[i,i] = \text{Number of correctly classified instances of class } i$$

- **Off-Diagonal Entries:**

$$M[i,j]\ (i \neq j) = \text{Number of instances of class } i \text{ misclassified as class } j$$

Interpretation in Multi-Class Problems

- **Diagonal Elements:**
  Large values along the diagonal entries $M[i,i]$ reflect high classification accuracy for class i.

- **Row Totals (Actual Class Viewpoint):**
  The sum $\sum_{j=1}^{N} M[i,j]$ represents the total count of instances truly belonging to class i. Evaluating this against $M[i,i]$ reveals the ratio of correctly classified versus misclassified samples for that class.

- **Column Totals (Predicted Class Viewpoint):**
  The sum $\sum_{i=1}^{N} M[i,j]$ reflects the total number of instances the model predicted as class j, regardless of their actual label. A high number of incorrect predictions from one class to another suggests **confusion** between those specific classes.[17]

The confusion matrix offers valuable insights into the classifier's behavior, highlighting not only overall performance but also specific misclassification patterns between class pairs or groups.

Accuracy

Accuracy quantifies the overall fraction of correct predictions made by the model [17]. It is defined as:

$$\text{Accuracy} = \frac{\sum_{i=1}^{C} \text{Correct}_i}{\text{Total Samples}} \tag{2.1}$$

where C is the number of classes, and Correct_i is the number of correctly predicted samples in class i. Accuracy is straightforward to compute and interpret, making it a commonly used metric. However, it may be misleading when class distributions are imbalanced, as it can mask the types of errors made by the model.

Precision

Also referred to as *Positive Predictive Value (PPV)* or *True Discovery Rate (TDR)*, precision quantifies the proportion of predicted positive samples that are truly positive [17]. In binary classification, it is defined as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.2}$$

where TP represents the number of true positives and FP the number of false positives. Precision is particularly useful in scenarios where the cost of false positives is high (e.g., medical diagnosis or alert systems). However, it does not account for false negatives and therefore should be considered alongside recall for a comprehensive evaluation.

Recall

Also known as *Sensitivity* or *True Positive Rate (TPR)*, recall measures the proportion of actual positive samples that are correctly identified by the model [44]. In binary classification:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.3}$$

where FN is the number of false negatives. Recall is especially critical in tasks like disease detection or fraud identification, where missing a positive case can have severe consequences. Nevertheless, optimizing recall alone may lead to an increase in false positives, so it is important to balance it with precision.

F1-Score

The F1-score is a statistical measure that combines *Precision* and *Recall* into a single metric of classification performance [17]. It is defined as the harmonic mean of Precision and Recall:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.4}$$

The F1-score is especially beneficial when dealing with imbalanced datasets, as it balances the trade-off between false positives and false negatives. However, it assumes equal cost for both types of errors, which may not always be appropriate depending on the application.

Specificity

Specificity, also known as the True Negative Rate (TNR), measures the proportion of actual negative samples that are correctly identified as negative [17]. It is defined as:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{2.5}$$

where TN is the number of true negatives and FP the number of false positives. Specificity is commonly used in binary classification settings such as screening or medical diagnostics, where minimizing false alarms is important. While it complements recall, it offers limited insight into how the model handles actual positives and may be less intuitive in balanced datasets.[17]

Table 2.2: Metrics used in classification task [17]

Common Name	**Other Names**	**Definition**	**Interpretation**
True Positive	Hit	True sample labeled as true	Correctly labeled true sample
True Negative	Rejection	False sample labeled as false	Correctly labeled false sample
False Positive	False alarm, Type I Error	False sample labeled as true	Incorrectly labeled false sample
False Negative	Miss, Type II Error	True sample labeled as false	Incorrectly labeled true sample
Recall	True Positive Rate	$\frac{TP}{TP+FN}$	Percentage of true samples correctly labeled
Specificity	True Negative Rate	$\frac{TN}{TN+FP}$	Percentage of false samples correctly labeled
Precision	Positive Predictive Value	$\frac{TP}{TP+FP}$	Percentage of samples labeled true that are actually true
Negative Predictive Value	–	$\frac{TN}{TN+FN}$	Percentage of samples labeled false that are actually false
False Negative Rate	–	$\frac{FN}{TP+FN} = 1 - \text{Recall}$	Percentage of true samples incorrectly labeled
False Positive Rate	Fall-out	$\frac{FP}{FP+TN} = 1 - \text{Specificity}$	Percentage of false samples incorrectly labeled
False Discovery Rate	–	$\frac{FP}{TP+FP} = 1 - \text{Precision}$	Percentage of predicted true samples that are actually false
True Discovery Rate	–	$\frac{FN}{TN+FN} = 1 - \text{NPV}$	Percentage of predicted false samples that are actually true
Accuracy	–	$\frac{TP+TN}{TP+TN+FP+FN}$	Percentage of all samples correctly labeled
F1 Score	–	$\frac{2 \cdot TP}{2 \cdot TP+FP+FN}$	Harmonic mean of Precision and Recall; approaches 1 as errors decline

2.5 Grad-CAM

Grad-CAM is a widely adopted method for visualizing and interpreting the predictions of convolutional neural networks, especially in visual recognition tasks. It produces class-discriminative localization maps by leveraging the gradient information flowing into the final convolutional layers of a network, highlighting regions in the input image that are important for the model's decision [45].

2.5.1 Motivation and Intuition

In CNNs, the deeper convolutional layers capture high-level semantic features while preserving spatial information, making them well-suited for determining -where- the model is attending in the input image. Grad-CAM utilizes the gradient of the target class score with respect to these feature maps to identify which regions contributed most to the decision. Unlike earlier methods such as Class Activation Mapping (CAM) [3], which require architectural modifications (e.g., Global Average Pooling (GAP) and a single linear layer), Grad-CAM can be applied post hoc to a wide range of pretrained models without altering their structure or requiring retraining.

2.5.2 Class Activation Mapping (CAM)

CAM is a visualization technique that identifies image regions contributing significantly to a CNNs prediction. It relies on a specific architecture where a GAP layer is followed by a fully connected layer (see Figure 2.3).

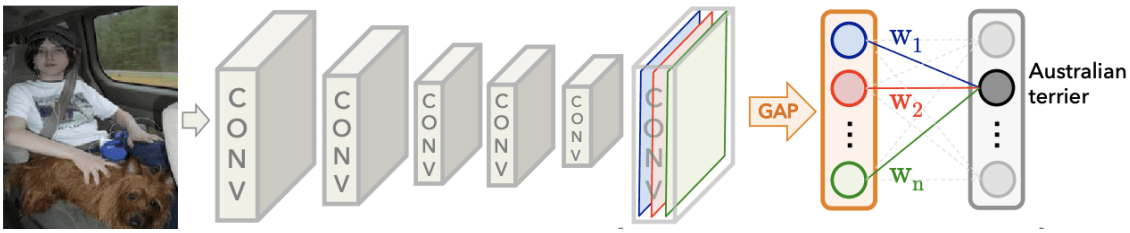


Figure 2.3: CAM architecture [3].

Given k feature maps $A^1, A^2, \ldots, A^k$, the class score y^c for class c is computed as:

$$y^c = \sum_k w_k^c \cdot GAP(A^k) \tag{2.6}$$

Here, w_k^c is the weight connecting the k-th feature map to class c. The corresponding class activation map is given by:

$$CAM^c = \sum_k w_k^c \cdot A^k \tag{2.7}$$

This weighted sum, upsampled to the input image size, highlights the most influential regions (Figure 2.4).

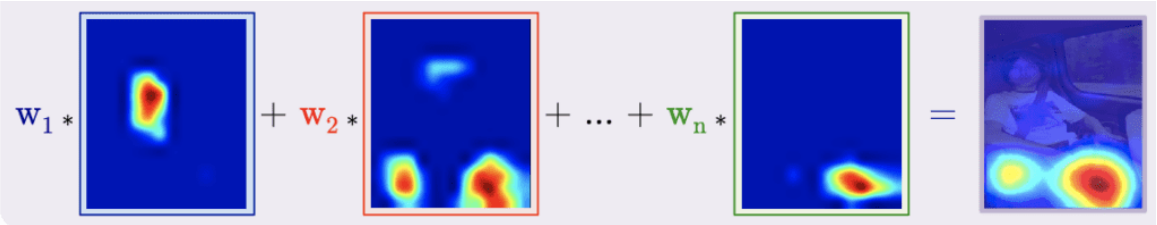


Figure 2.4: Computation of CAM and its overlay on the input image. [3]

2.5.3 Gradient-weighted Class Activation Mapping (Grad-CAM)

To overcome CAM's architectural constraints, Grad-CAM generalizes the idea by using the gradients of the output score for a specific class with respect to the feature maps of a convolutional layer. This enables its application to a wide variety of CNN architectures (Figure 2.5).

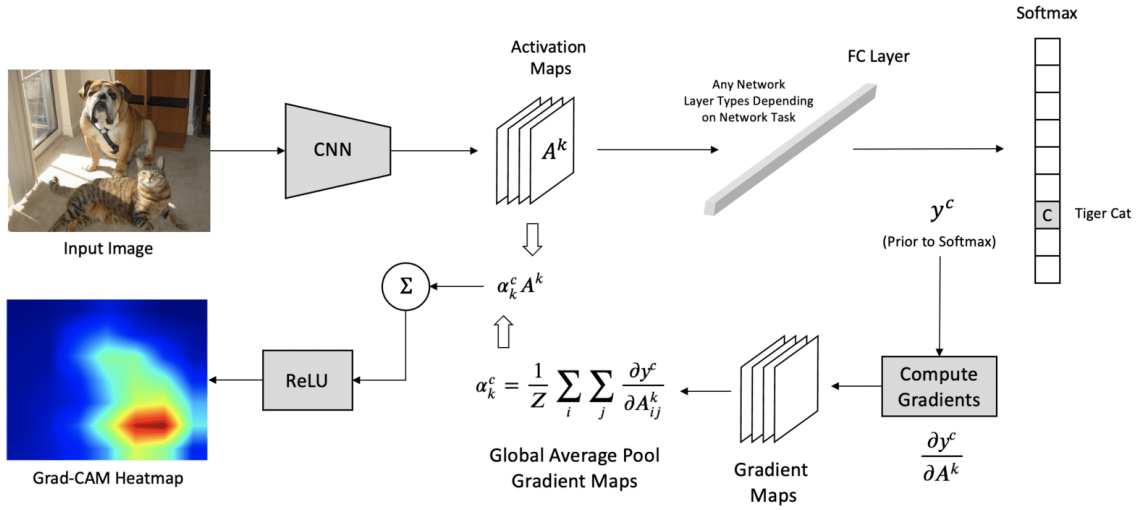


Figure 2.5: Grad-CAM architecture [4].

The method involves the following steps:

1. **Gradient Computation:** For a target class c, compute the gradient of the class score y^c with respect to the feature maps A^k:

$$\frac{\partial y^c}{\partial A^k} \tag{2.8}$$

2. **Weight Calculation:** Compute the importance weights α_k^c by performing GAP over the gradients:

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{2.9}$$

where Z is the number of spatial locations.

3. **Heatmap Generation:** Generate the class-discriminative heatmap:

$$L^c_{Grad-CAM} = ReLU\left(\sum_k \alpha^c_k A^k\right) \tag{2.10}$$

This heatmap, when upsampled and overlaid on the input image, reveals the areas that most influenced the model's prediction. In models like EfficientNetV2, the SiLU activation function may be used instead of ReLU to enhance expressiveness [4].

2.5.4 Comparison with CAM and Other Methods

Grad-CAM is a strict generalization of CAM. The weight expression used in CAM:

$$w^c_k = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}}$$

is mathematically equivalent to the Grad-CAM weights α^c_k, up to a normalization factor. While CAM requires specific architectural components, Grad-CAM can be applied post hoc to diverse CNN, including those used in image captioning and visual question answering [45].

However, Grad-CAM's heatmaps are relatively coarse. To improve detail and sharpness, it is often combined with high-resolution gradient-based methods such as Guided Backpropagation or Deconvolution. The resulting visualization, known as Guided Grad-CAM, is produced by element-wise multiplication of the Grad-CAM heatmap and the Guided Backpropagation map, yielding semantically rich and visually detailed explanations.

2.5.5 Applications in Medical Imaging

In medical imaging, Grad-CAM enhances model transparency by highlighting regions associated with diagnostic relevance, such as tumors, lesions, or anatomical boundaries. This visual interpretability supports clinical decision-making, model validation, and regulatory approval. Its compatibility with various architectures makes it an effective tool for post hoc analysis in computer-aided diagnosis systems and other sensitive domains.

2.6 Conclusion

This chapter has presented a comprehensive introduction to the fundamental concepts underlying ML and DL, with a focus on their application in medical image classification. We examined the core paradigms of ML, the unique capabilities of deep neural networks, and the role of CNN in visual recognition tasks. The Grad-CAM technique was introduced as a powerful tool for model interpretability, enabling visual assessment of the regions contributing to predictions—an essential feature in sensitive domains such as healthcare. Finally, we discussed key performance metrics for evaluating classification models. Together, these concepts form the conceptual backbone for developing, evaluating, and interpreting AI systems in medical imaging, providing a solid foundation for the more specialized methodologies discussed in the following chapters.

3

Related Work

Contents

3.1 Introduction

The integration of deep learning into medical imaging has led to significant advances in automated disease detection and localization. Within this context, (ROI) detection plays a pivotal role in enhancing both diagnostic accuracy and the interpretability of machine learning models. By focusing computational attention on clinically significant regions—such as lesions or pathological features ROI-based methods help address challenges associated with high-resolution, noisy medical images and the limited availability of annotated data.

This chapter presents a comprehensive review of current ROI detection techniques used in chest X-ray classification and other medical imaging tasks. The discussion is structured around two main categories: (1) established ROI detection methods, which typically involve object detection or segmentation models that explicitly identify areas of interest, and (2) Grad-CAM-based approaches, which use feature attribution to retrospectively highlight important image regions. These methodologies form the basis for the proposed framework in this thesis, which aims to enhance diagnostic performance through anatomically informed ROI segmentation and lightweight attention mechanisms.

3.2 Established ROI Detection Methods

In recent years, numerous established approaches have been developed to accurately detect regions of interest in medical images. These methods typically employ object detection or segmentation algorithms that explicitly localize pathological regions. The following section represent some studies and techniques :

1. **Guo et al. (2024) [5] :** In this study, the authors proposed a deep learning-based diagnostic system for chest X-ray analysis using a Faster Region-based CNN (R-CNN)

framework. The model architecture integrates a ResNet-50 backbone pretrained on ImageNet, along with a Feature Pyramid Network (FPN) to detect lesions across multiple scales, ranging from small nodules to larger consolidations. The detection process involves two stages: a Region Proposal Network (RPN) generates approximately 2000 candidate lesion regions, followed by ROI pooling and dual-headed fully connected layers for both lesion classification and bounding box regression. Non-maximum suppression is then applied to eliminate redundant detections and produce final lesion predictions [5]. As shown in Figure 3.1, the study also evaluated the system through a competition between radiologists with and without AI assistance.

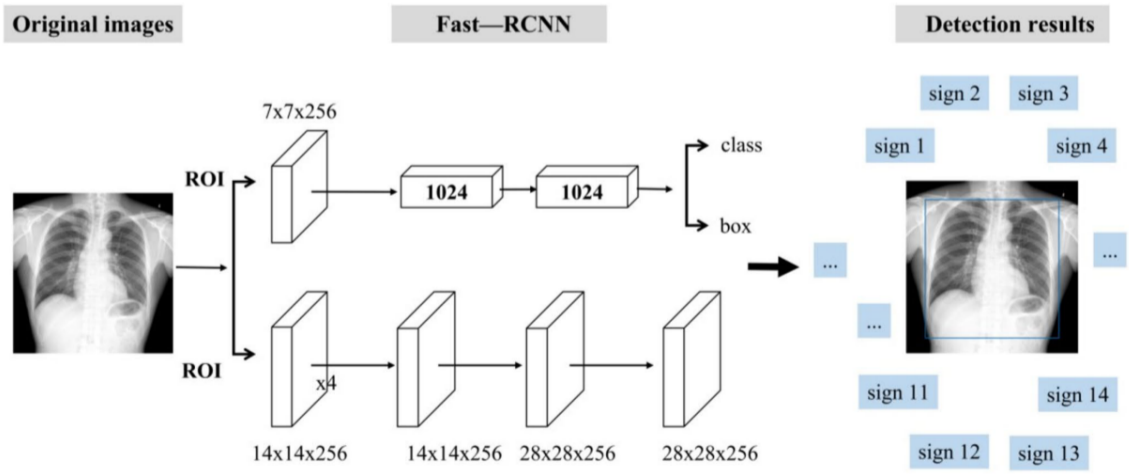


Figure 3.1: Evaluation flowchart. [5]

Dataset

This study is based on a chest X-ray image database of 4098 patients admitted to our hospital. For this experiment, the engineer of the Department of Medical Imaging retrieved the image data from April 2007 to June 2019 in the picture archiving and communication system (PACS) and the corresponding clinical information data in the RadiologyInformation System (RIS) [5].

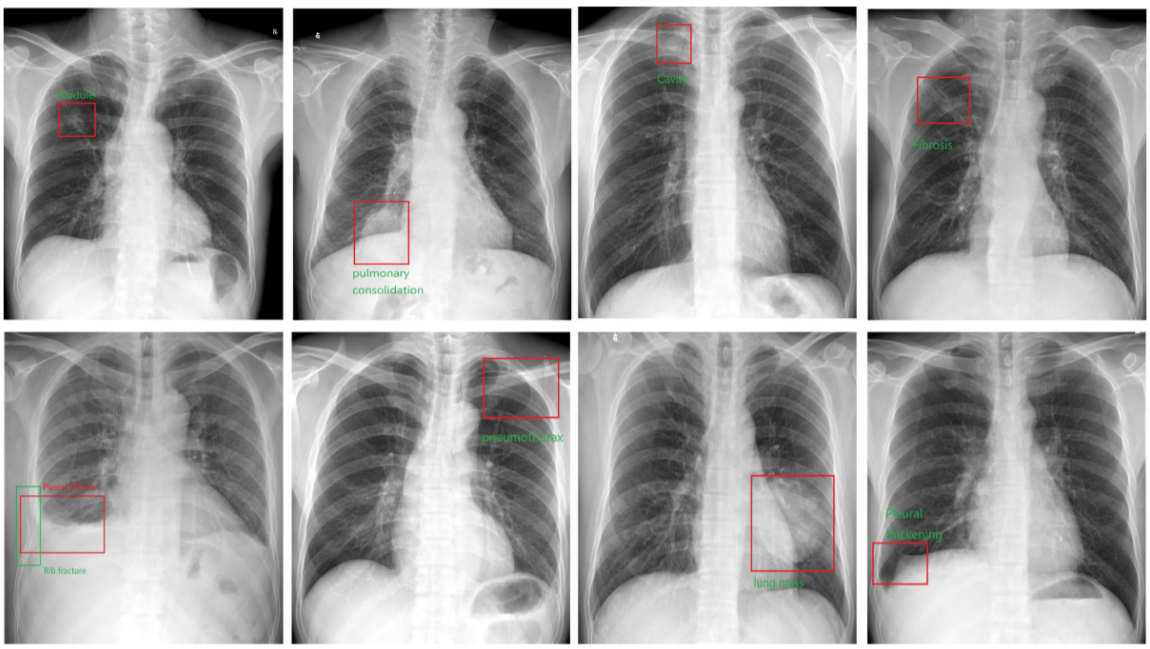


Figure 3.2: The AI system output.[5]

Experimental Results

The model achieved high performance in detecting and localizing various lesions on chest X-rays. As shown in Figure 3.2, the AI system outputs bounding boxes and labels of the lesions to assist radiologists. According to the results presented in Table 3.1, the AI system outperformed radiologists in identifying most of the evaluated signs, including normal cases, fibrosis, mass, pleural effusion, and pulmonary consolidation. However, radiologists demonstrated better performance in detecting aortic calcification, calcification, and cavities. These results highlight the AI model's strength in detecting common and clear radiographic features, while suggesting that radiologists may still have an edge in more subtle or complex findings [5].

Table 3.1: Comparison of AI and Radiologists on X-ray Signs [5]

X-ray Sign	AI (95% CI)	Radiologists (95% CI)	Advantage
Normal	1.000 (1.000–1.000)	0.991 (0.971–1.000)	AI
Fibrosis	0.950 (0.896–1.000)	0.900 (0.818–0.982)	AI
Heart shadow enlargement	0.991 (0.970–1.000)	0.980 (0.948–1.000)	AI
Mass	1.000 (1.000–1.000)	0.951 (0.896–1.000)	AI
Pleural effusion	0.993 (0.979–1.000)	0.949 (0.886–1.000)	AI
Pulmonary consolidation	0.982 (0.951–1.000)	0.904 (0.816–0.992)	AI
Aortic calcification	0.981 (0.953–1.000)	0.993 (0.978–1.000)	Radiologists
Calcification	0.915 (0.832–0.998)	0.933 (0.812–1.000)	Radiologists
Cavity	0.847 (0.742–0.952)	0.963 (0.906–1.000)	Radiologists
Nodule	0.881 (0.786–0.976)	0.923 (0.840–1.000)	Radiologists
Pleural thickening	0.895 (0.806–0.984)	0.957 (0.909–1.000)	Radiologists
Rib fracture	0.980 (0.937–1.000)	0.987 (0.958–1.000)	Radiologists
Subphrenic free air	1.000	1.000	No difference
Pneumothorax	1.000	1.000	No difference

2. **Rajpal et al. (2021) [6]:** Proposed a three-module hybrid framework Figure 3.3 for detecting COVID-19 from chest X-ray images by integrating handcrafted features with deep learning. In the first module, chest X-rays are preprocessed and passed through a ResNet-50 model (pretrained on ImageNet), extracting 2048 high-level features. Data augmentation techniques like zooming, flipping, and shearing are used to improve generalization. In the second module, 252 handcrafted features are extracted using statistical analysis on spatial (Gray-Level Co-occurrence Matrix (GLCM), Gray-Level Difference Matrix (GLDM)) and frequency domains (Fast Fourier Transform (FFT), Discrete Wavelet Transform (DWT)). These are reduced via PCA and passed through a neural network to generate a compact 16-dimensional feature vector. In the third module, the deep and handcrafted features (2048 + 16) are concatenated and passed through a fully connected layer and softmax classifier to predict one of three classes: COVID-19, pneumonia, or normal. This framework demonstrates the benefit of combining domain knowledge (via handcrafted features) with deep features to improve diagnostic accuracy.[6]

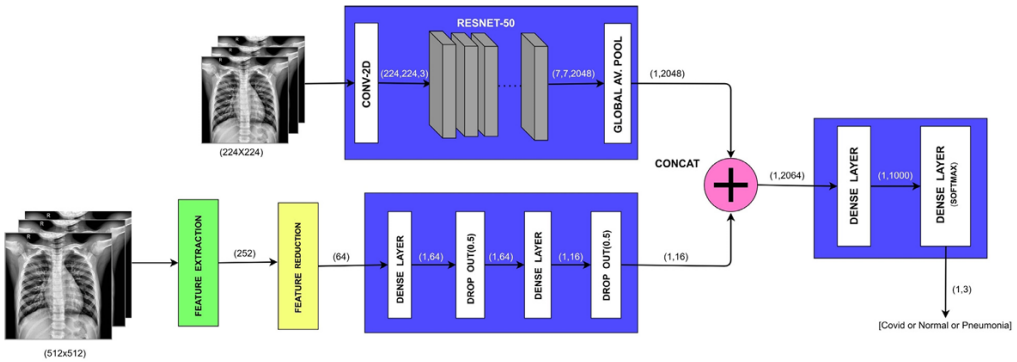


Figure 3.3: Architecture of the Proposed Network. [6]

Dataset

The study utilized a carefully curated dataset of 1560 chest X-ray images from multiple public sources to ensure robustness and class balance:

- **COVID-19 Radiography Database (Kaggle)** – 2905 images:
  - COVID-19: 219
  - Normal: 1341
  - Viral Pneumonia: 1345
- **COVID-19 Image Data Collection** – 760 images:
  - COVID-19: 538
  - ARDS: 14
  - Other: 222
- **COVID-chestxray-dataset** – 53 COVID-19 images.
- **Actualmed-COVID-chestxray-dataset** – 150 COVID-19 images.

To ensure uniformity, only frontal (PA and AP) view images were used. The final balanced dataset comprised 520 COVID-19, 520 pneumonia (bacterial + viral), and 520 normal chest X-rays, with an additional independent validation cohort of 157 COVID-19 samples for testing generalization.
Figure 3.4 which summarizes information about precision, recall, and F1-score metrics for 10-fold cross-validation for all three classes. Note that the proposed framework is able to label almost all COVID-19 patients correctly, thus achieving high average values of the precision, recall, and F1-score (≥ 0.987)across 10 different folds.[6]

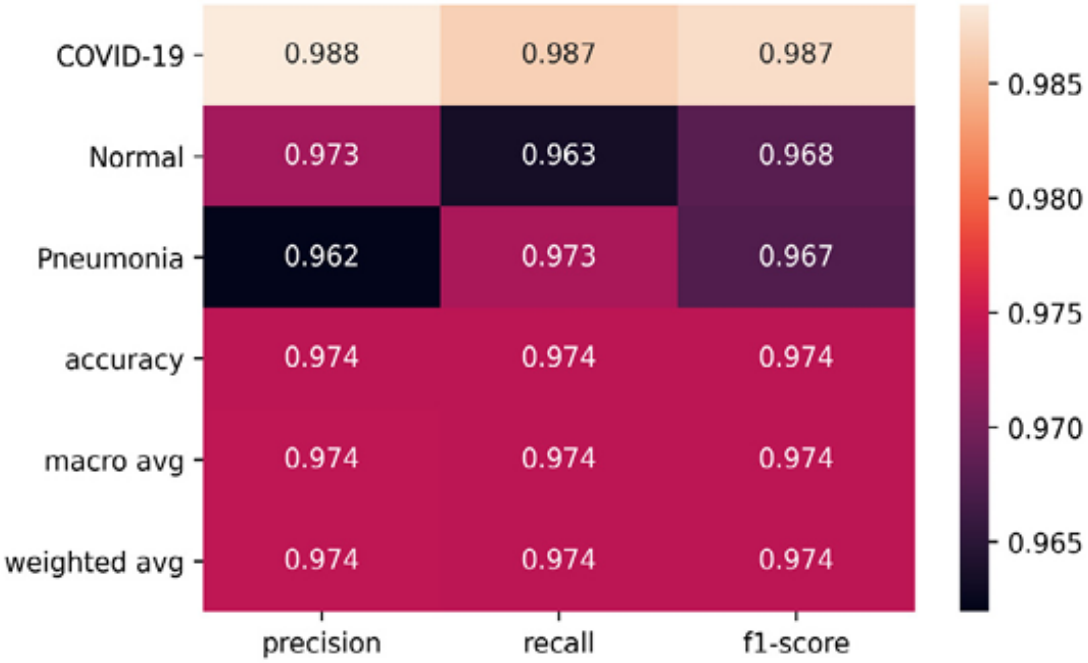


Figure 3.4: Heatmap of the proposed framework.[6]

3. **Hongyu et al. (2020) [7] :** In this study, the authors proposed CheXLocNet, a deep learning model for detecting and localizing pneumothorax in chest X-rays. They treated localization as a pixel-wise image segmentation task.

   CheXLocNet Figure 3.5 is built on the Mask R-CNN and has four main parts:

   - A ResNet-50 with FPN backbone extracts multi-scale features from the input image.
   - A RPN generates candidate regions likely to contain pneumothorax.
   - RoIAlign ensures the candidate regions are precisely resized for further processing.
   - Two parallel branches one for classification and one for segmentation masks output the final results.
   - A decoder module is also used to recover detailed semantic information. This design helps CheXLocNet achieve both accurate diagnosis and precise localization.

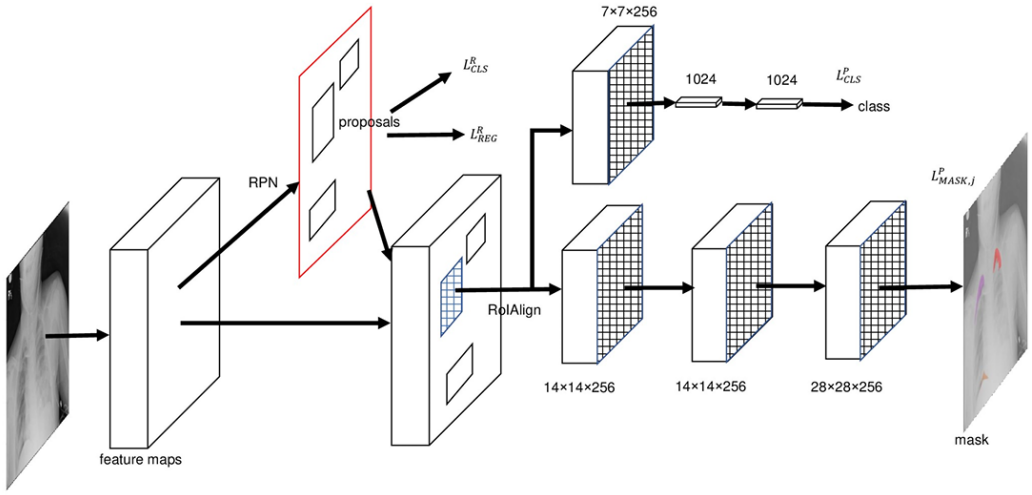


Figure 3.5: The framework of CheXLocNet.[7]

Dataset

For model development and evaluation, the authors utilized the SIIM-ACR Pneumothorax Segmentation dataset from Kaggle, which consists of 12,047 chest radiographs with pixel-level annotations. The dataset was divided into 75% training, 12.5% validation, and 12.5% test sets. Specifically, the training set included 2,079 positive (pneumothorax) and 7,250 negative cases; the validation set comprised 300 positive and 1,046 negative cases; and the test set contained 290 positive and 1,082 negative radiographs. The test set was the official competition test set, ensuring no overlap with the training or validation sets.

Experimental results

The authors trained six different CheXLocNets using distinct procedures, selecting optimal parameters based on the AP50 score on the validation set, which ranged from 0.20 to 0.36. CheXLocNet III achieved the highest AP50 score of 0.36. The classification performance of these models was evaluated and shown in Table 3.2 . CheXLocNet III demonstrated the best performance across multiple metrics, achieving: AUC: 0.86 F1 score: 0.64 Sensitivity: 0.82 (CI 0.78-0.87)

CheXLocNet V showed the best Specificity (0.92, CI 0.90-0.93) and PPV (0.65, CI 0.59-0.71).

Table 3.2: Classification performance of CheXLocNets on the validation set. [7]

Model	AUC	F1	Sensitivity	Specificity	PPV
CheXLocNet I	0.80	0.58	0.72 (0.67-0.77)	0.78 (0.76-0.81)	0.49 (0.44-0.53)
CheXLocNet II	0.83	0.63	0.68 (0.63-0.74)	0.86 (0.83-0.88)	0.58 (0.53-0.63)
CheXLocNet III	0.86	0.64	0.82 (0.78-0.87)	0.78 (0.76-0.81)	0.52 (0.48-0.57)
CheXLocNet IV	0.82	0.59	0.70 (0.64-0.75)	0.84 (0.82-0.86)	0.54 (0.49-0.59)
CheXLocNet V	0.81	0.59	0.66 (0.60-0.71)	0.92 (0.90-0.93)	0.65 (0.59-0.71)
CheXLocNet VI	0.80	0.57	0.54 (0.49-0.60)	0.79 (0.76-0.81)	0.48 (0.44-0.53)

3.3 Grad-CAM-based approaches

In recent years, Grad-CAM-based approaches have been developed to highlight important regions that contribute to a model's decision. .Several studies have extended Grad-CAM by integrating it with other approaches to improve region of interest detection . The following section presents some studies and techniques based on this approach.

1. **Zhou et al. (2024) [8] :** The authors proposed an unsupervised contrastive learning framework Figure 3.6 that integrates Grad-CAM to guide representation learning. The model consists of three main components: a backbone encoder, an instance discrimination branch, and a cluster branch. Grad-CAM is used not only for visualization but also to generate heatmaps that identify discriminative regions (e.g., lesions). These heatmaps help select strong positive pairs (local-global views) during contrastive training. The instance branch focuses on learning invariant features, while the cluster branch improves semantic consistency via KNN-based clustering. This approach enhances localization and classification performance without needing pixel-level annotations.

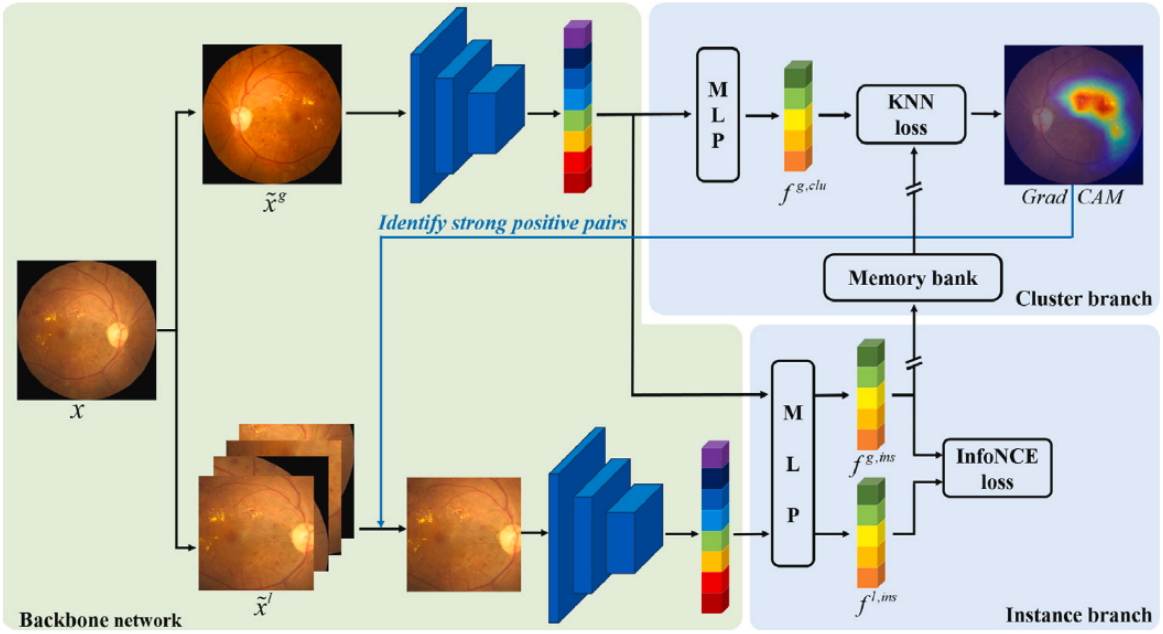


Figure 3.6: The overall network architecture of the proposed method. [8]

Dataset

The proposed method was evaluated through binary classification on five retinal disease datasets. For unsupervised comparison, three large-scale datasets (EyePACS, DDR, ADAM) were used without fine-tuning—labels were only used for testing. For transfer learning, two smaller datasets (BoVW-DR2 and IDRiD) were used with fine-tuning. The EyePACS dataset provided 1,857 high-quality fundus images for normal/abnormal classification. The DDR dataset (13,673 images) was used for DR/non-DR classification. The ADAM dataset focused on AMD classification with 5-fold cross-validation. For transfer learning, BoVW-DR2 (435 images) and IDRiD (516 images) were used for DR classification. [8]

Experimental results

"the experimental results of different pre-trained models transferred to the target IDRiD dataset Table 3.3. Our method maintains superior performance (83.27% Area Under Curve (AUC)), outperforming other pre-trained models by at least 3.69% in AUC. The IDRiD dataset contains many DR images with subtle lesions, making the classification of DR and non- DR images a more challenging task. Thus, the classification accuracy is much lower than the results on the BoVW-DR2 dataset. Most contrastive learning pre-trained models exhibit better performance than the supervised counterpart, which implies that contrastive learning may learn more generalized semantic features in some cases. In summary, experimental results on these two dataset show that our method is promising to be transferred to different small-scale fundus image datasets after pre-training on unlabeled datasets."[8]

Table 3.3: Quantitative Comparison of Different Pre-trained Models. [8]

Pre-train	**AUC**	**ACC**	**Precision**	**Recall**	**F1-score**
Random	75.98	72.82	72.98	75.98	72.05
ImageNet	79.56	79.61	77.28	79.56	77.98
Supervised	79.58	78.64	76.73	79.58	77.30
SimCLR [20]	77.34	78.64	75.97	77.34	76.51
MoCo v1 [18]	76.68	74.76	73.71	76.68	73.66
MoCo v2 [19]	77.30	80.58	78.19	77.30	77.71
CC [30]	78.05	80.58	78.05	78.05	78.05
C2AM [24]	79.56	79.61	77.28	79.56	77.98
LEWEL [22]	75.15	76.70	73.89	75.15	74.38
LoGo [25]	75.87	77.67	74.86	75.84	75.28
LD [33]	80.31	79.61	77.56	70.31	78.22
Rotation [11]	76.60	78.64	75.87	76.60	76.20
CAMCL (Ours)	**83.27**	**80.58**	**79.56**	**83.27**	**79.74**

2. **Kumaresan et al. (2023) [9] :** This study presents a hybrid deep learning–radiomics framework for classifying COVID-19 from chest X-ray images using ROI localization. The main steps include:

   - **Image Preprocessing and Augmentation:** Enhancing image quality and expanding the dataset to improve training.
   - **Feature Extraction with Deformable CNN:** A deformable convolutional neural network produces a preliminary classification result and a 1,024-dimensional latent feature vector.
   - **Lung Region Segmentation and Lung Class Activation Map (LCAM) Generation:** Lung areas are segmented, and Grad-CAM++ is applied to generate a LCAM.

- **ROI Mask Creation:** The LCAM is thresholded to create a mask highlighting diagnostically relevant regions.
- **Radiomics Feature Extraction:** Handcrafted radiomics features are extracted from the defined ROI.
- **Feature Fusion and Classification:** The CNN latent features and radiomics features are concatenated and passed into traditional machine learning classifiers (e.g., SVM, Random Forest) for final image classification.

As illustrated in Figure 3.7, (b) is for the lung region activation maps generated by the proposed network highlight the important areas contributing to the classification.

Dataset

Chest radiograph images were collected from various public datasets.

- **COVID-19 Cases:**
  - 1,570 images collected from multiple limited-access datasets.
  - No demographic information was included.
  - Manual review performed to avoid duplicate entries and ensure independent train/test splits.
- **Non-COVID-19 Pneumonia & Normal Cases:**
  - 1,700 images per category (non-COVID-19 pneumonia and normal) from the RSNA Pneumonia Detection Challenge dataset.
  - Images collected from diverse institutions and age groups to minimize bias.
- **Dataset Splits:**
  - **Training Set:** 3,770 images (1,300 normal, 1,300 non-COVID-19 pneumonia, 1,170 COVID-19 pneumonia)
  - **Validation Set:** 300 images (100 from each class) for hyperparameter tuning and monitoring.
  - **Testing Set:** 900 images (300 from each class) used exclusively for final evaluation.

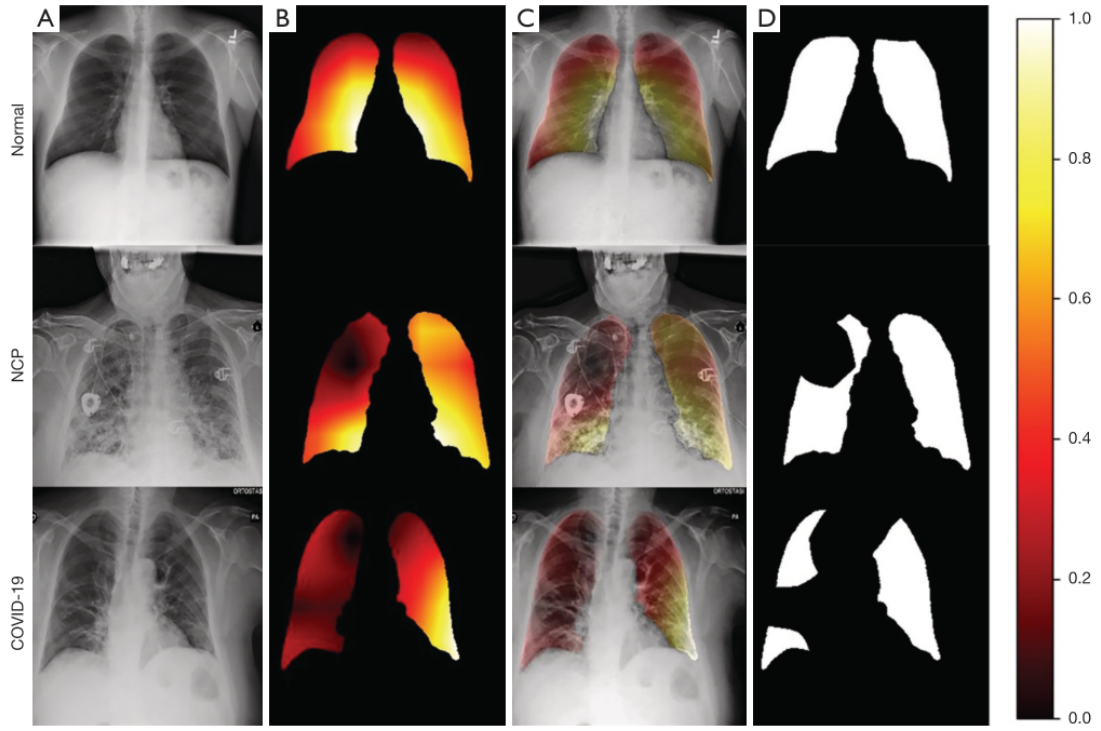


Figure 3.7: images generated by the proposed network.[9]

Evaluation of multi-class classification : Evaluation of Multi-class Classification: To evaluate the performance of the proposed multi-class classification framework, experiments were conducted over 50 train-test iterations, taking approximately 4 hours in total. The performance of various classification models with different feature combinations is summarized in Table 4.2. The results show that combining deep learning radiomics (DLR) features with additional radiomics features (ARF) generally improves classification accuracy across all models. Among them, the Multilayer Perceptron achieved the highest accuracy of $\mathbf{0.911 \pm 0.004}$, demonstrating the effectiveness of the combined feature set in the proposed framework.

Table 3.4: Comparison of Classification Models with Different Feature Sets. [9]

Model	Hyperparameters	Features	Accuracy (mean ± SD)	RSD (%)	P-value
Decision Tree	criterion = 'gini', min_samples_split = 2, min_samples_leaf = 1	DLR	0.842 ± 0.006	0.69	<0.0001
		DLR + ARF	0.854 ± 0.007	0.84	<0.0001
Random Forest	n_estimators = 100, criterion = 'gini'	DLR	0.904 ± 0.003	0.40	<0.0001
		DLR + ARF	0.910 ± 0.003	0.40	<0.0001
Linear SVC	penalty = 'l1', loss = 'squared_hinge', max_iter = 10000	DLR	0.867 ± 0.001	0.10	<0.0001
		DLR + ARF	0.881 ± 0.001	0.11	<0.0001
Multilayer Perceptron	hidden_layer_sizes = (100), activation = 'relu', solver = 'adam'	DLR	0.908 ± 0.004	0.47	<0.0001
		DLR + ARF	**0.911 ± 0.004**	0.42	<0.0001

3. **Zhuang et al. (2023) [10] :** proposed a multi-label classification model for retinal disease detection that integrates Grad-CAM-based attention mechanisms to guide image enhancement. The model aims to improve classification accuracy by focusing on lesion-relevant regions of the image. The method follows these key steps:

   - **Input and Feature Extraction:** The original fundus images are passed through a CNN to extract feature maps.
   - **Grad-CAM Heatmap Generation:** Using Grad-CAM, they generated heatmaps to highlight regions of interest (lesion areas) in the fundus images. The attention (heatmap) is used to crop and enhance ROI from the original images. These CAM-based enhanced images serve as additional training data.Figure3.8 show The process of extracting the lesion area from the image using the Grad-CAM method. Attention image crop of (a) the original image, (b) the Grad-CAM heatmap, (c) the overlay of the original image and Grad-CAM heatmap, (d) the identified attention area, (e) the clipped attention area, and (f) the attention area resized to match the original image dimensions.
   - **Dual CNN Architecture:** Two CNN backbones are used:
     - BaseModel1: VGG16
     - BaseModel2: ResNet50

     Each CNN is trained on the ROI-enhanced images independently.
   - **Ensemble Prediction:** The predictions from both networks are combined (ensemble learning) to generate the final multi-label classification output.

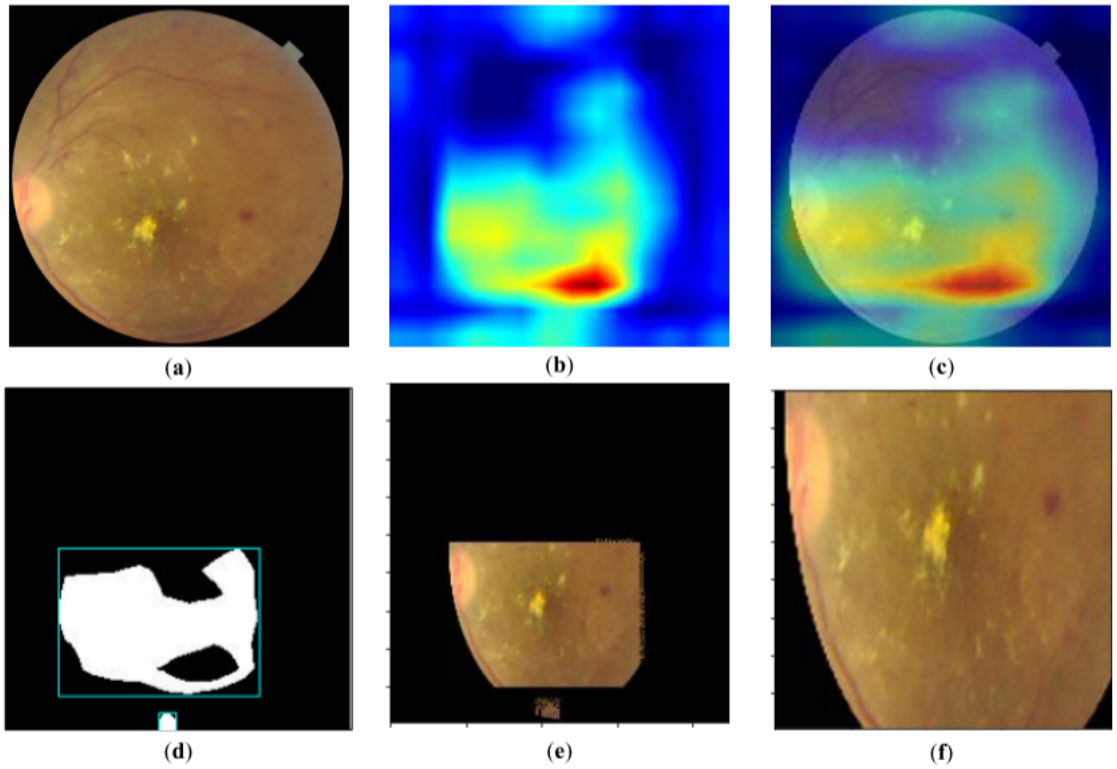


Figure 3.8: The process of extracting the lesion area. [10]

Dataset

To evaluate the model, experiments were conducted on the RFMiD, which includes 3,200 images labeled with 45 diseases. The dataset was split into 60% training, 20% validation, and 20% testing subsets. Among the labels, 26 diseases were treated as independent categories, while 19 others were grouped under the label "other." [10]

Experimental Results

Zhuang et al. [10] performed a module comparison experiment to analyze the effectiveness of each component of the model:

Table 3.5: Performance Comparison of Different Models on RFMiD Dataset.[10]

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
VGG16	0.8969	0.7292	0.7955	0.7609
ResNet50	0.9156	0.7451	0.8409	0.7378
Ensemble model	0.9172	0.7308	0.8083	0.8072
Ensemble + Image Enhance	**0.9737**	**0.7762**	**0.9167**	**0.8590**

4. **Dovletov et al.(2022) [11] :** The authors propose a novel framework that integrates Grad-CAM attention maps into a U-Net architecture for synthesizing pseudo-Computed Tomography (CT) images from MRI scans. The approach enhances the model's focus

on anatomically significant regions by using Grad-CAM to guide feature learning during training.

- **Base Model:** Uses a 3D U-Net architecture for pseudo-CT synthesis.
- **Grad-CAM Integration:** Grad-CAM is applied to encoder feature maps. Highlights important regions affecting the output.
- **Attention Mechanism:** Grad-CAM output is used as an attention map. Guides the decoder to focus on critical regions during reconstruction.
- **Loss Function:** Combines Mean Absolute Error (MAE) and Structural Similarity Index (SSIM) loss.

As illustrated in Figure 3.9, the proposed Grad-CAM guided U-Net approach leverages attention maps to improve the pseudo-CT synthesis process.

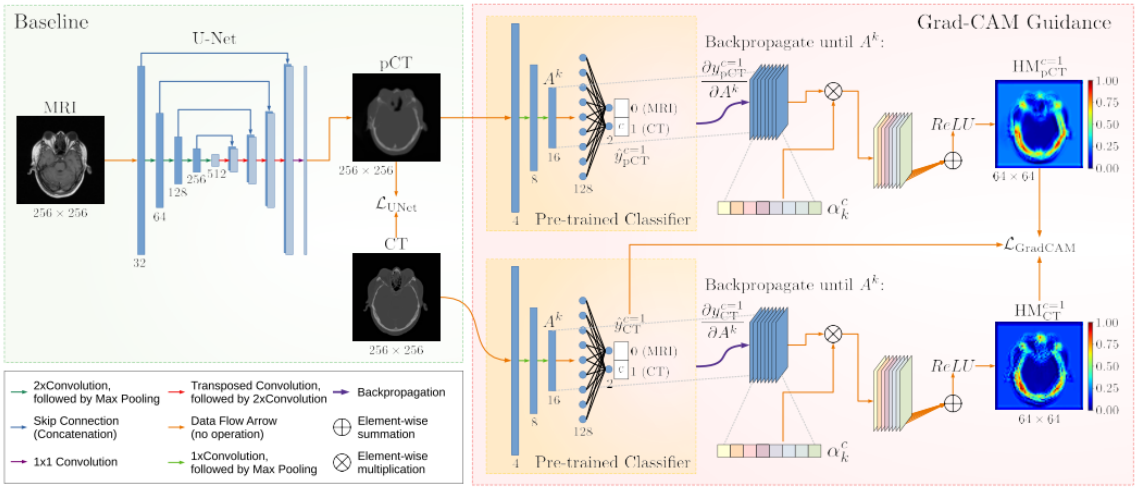


Figure 3.9: proposed Grad-CAM guided U-Net approach. [11]

Dataset

The proposed approach was evaluated using the publicly available RIRE dataset, which includes cranial image scans from 16 patients across multiple modalities. In this study, T1-weighted MRI images of size 256×256 were used alongside corresponding CT images of size 512×512. Since the MRI and CT volumes were not pre-aligned, a mutual-information-based multi-resolution registration method (Mattes algorithm) implemented in SimpleITK was applied, with CT images used as fixed references and MRI images as moving volumes. The registered images were resampled to isotropic voxel spacing and resized to 256×256 pixels. Due to differences in field of view, slices without valid counterparts were excluded. After visual inspection for quality control, a total of 553 well-aligned MRI-CT image pairs were selected for model training and evaluation.[11]

Experimental Results For evaluation, They used

- MAE – Lower values indicate better performance.
- SSIM – Higher values indicate better structural similarity to the ground truth.

Aa show in the table 3.6 The Grad-CAM Guided U-Net achieved lower MAE and higher SSIM compared to baseline U-Net and attention-less versions.

Table 3.6: Performance comparison between U-Net and GCG U-Net. [11]

Name	**Entire Image**		**Head Area**		**Bone Area**	
	MAE	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**
U-Net	101±35	69139±27664	180±30	131393±38343	595±120	532695±198330
GCG U-Net (ours)	96±34	61072±27413	169±32	116398±35502	477±106	372435±141401

Metric	**U-Net**	**GCG U-Net**
PSNR (dB)	24.3±1.9	25.0±2.1
SSIM	79.6±6.8	80.6±6.7

5. **Shuai Xu et al. (2021) [18] :** This paper presents a method that uses Grad-CAM to guide a dual attention mechanism for better image classification. The goal is to improve the model's ability to focus on important image regions by combining Grad-CAM with channel and spatial attention modules. The method follows these key steps:

   - Feature extraction using a backbone CNN (e.g., ResNet-50).
   - Grad-CAM generates attention maps highlighting key image regions.
   - A Channel-Spatial Attention Module is trained using these maps to:
     - Emphasize important feature channels.
     - Focus on relevant spatial areas.
   - Final classification is based on the enhanced features.

   Datasets The authors test their model on :

   - CUB-200-2011 (Bird species) – 200 classes, 11,788 images.
   - Stanford Cars – 196 classes, 16,185 images.
   - FGVC Aircraft – 100 classes, 10,000 images.

   Experimental Results As presented in Table 3.7, the proposed method shows strong performance across all three datasets. On the CUB-200-2011 dataset, it outperforms the second-best method, TASN, by 1.24% when using the VGG19 backbone. It also achieves better accuracy than ACNet when using ResNet50, improving results by 0.35%.

   For the FGVC-Aircraft dataset, the method achieves the highest accuracy of 93.42% with ResNet50, which is about 0.4% higher than the previous best (DCL). Even with VGG19, it slightly improves over earlier methods.

   On the Stanford Cars dataset, the method performs better than most other approaches, especially when using VGG19. While ACNet with ResNet50 achieves a slightly higher accuracy (0.19% more), it relies on a more complex model structure and training process, whereas the proposed method is simpler and more efficient.

Table 3.7: Comparison with State-of-the-art Methods. [18]

Datasets	**Base Model**	**CUB-200-2011**	**FGVC-Aircraft**	**Stanford Cars**
RA-CNN (CVPR17)	VGG19	85.30	88.20	92.50
MA-CNN (ICCV17)	VGG19	84.92	90.35	92.80
SENet (CVPR18)	VGG19	84.75	90.12	89.87
SENet (CVPR18)	ResNet50	86.78	91.37	93.10
CBAM (ECCV18)	VGG19	84.92	91.30	93.10
CBAM (ECCV18)	ResNet50	86.99	91.91	93.25
DFL (CVPR18)	ResNet50	87.40	91.70	93.10
NTS (ECCV18)	ResNet50	87.52	91.48	93.66
TASN (CVPR19)	VGG19	86.10	90.83	93.47
TASN (CVPR19)	ResNet50	87.90	92.00	93.90
DCL (CVPR19)	ResNet50	87.80	93.00	94.50
ACNet (CVPR2020)	ResNet50	88.10	92.40	**94.60**
the study	VGG19	87.34	91.55	93.32
the study	ResNet50	**88.45**	**93.42**	94.41

Summary of Selected Studies

The summary table 3.8 highlights key distinctions in learning strategies, model architectures, dataset modalities, and the role of Grad-CAM in ROI generation. Among studies that used Grad-CAM explicitly for ROI extraction such as Zhou et al., Kumaresan et al [9], Zhuang et al [10], and Dovletov et al[11]. notable improvements in classification accuracy or reconstruction quality were observed. For instance, Zhuang et al [10] achieved the highest accuracy (97.37 %) by enhancing training images with Grad-CAM-based lesion-focused cropping. Similarly, Kumaresan et al [9] reported 91.1% accuracy using a hybrid of Grad-CAM-guided radiomics and deep features. On the other hand, studies like Rajpal et al. and Guo et al. achieved strong results without Grad-CAM, relying instead on handcrafted features or object detection pipelines. While methods such as Faster R-CNN and Mask R-CNN (e.g., Guo et al., Hongyu et al.) provided explicit ROI through bounding boxes, they did not leverage Grad-CAM. However, from this comparison, we observe that there is actually less reliance on unsupervised.

Table 3.8: Summary of Selected Studies in Medical Image Analysis and Grad-CAM Usage.

Study	**Year**	**Supervision**	**Model**	**Dataset**	**Grad-CAM for ROI?**	**Accuracy (%)**
Zhou et al.	2024	Unsupervised	Contrastive Encoder with Grad-CAM	IDRiD, DDR (Retinal)	✓	80.58
Guo et al.	2024	Supervised	Faster R-CNN (ResNet-50 + FPN)	4098 CXRs (Chest)	✗	–
Kumaresan et al.	2023	Supervised	Deformable CNN + Radiomics + SVM/RF	RSNA, COVID sets (Chest)	✓	91.1
Zhuang et al.	2023	Supervised	Ensemble CNN (VGG16 + ResNet50)	RFMiD (Retinal)	✓	97.37
Dovletov et al.	2022	Supervised	3D U-Net + Grad-CAM Attention	RIRE (MRI–CT)	✓	–
Rajpal et al.	2021	Supervised	ResNet50 + Hand-crafted Features (GLCM, FFT)	1560 CXRs (Chest)	✗	∼98.7
Xu et al.	2021	Supervised	ResNet50 + Channel-Spatial Attention	CUB-200, Cars (Fine-Grained)	✓	88.45
Hongyu et al.	2020	Supervised	Mask R-CNN (CheXLocNet)	SIIM-ACR (Chest)	✗	–

3.4 Conclusion

The reviewed literature highlights the wide range of methodologies applied to ROI detection in medical imaging, including both explicit object localization techniques—such as Faster R-CNN and Mask R-CNN—and implicit Grad-CAM approache . Established ROI detection methods provide accurate region localization but often require detailed pixel-level annotations and high computational resources. In contrast, Grad-CAM-based methods offer visual interpretability and broader generalization, though they may lack fine-grained localization precision. These observations have guided our two-phase approach, which evaluates Grad-CAM as an unsupervised ROI detection method. By combining ROI-based image generation with comparisons across model architectures and fine-tuning strategies, this work addresses the underexplored question of whether using Grad-CAM during preprocessing can improve classification accuracy. The next chapters outline the system design and how it extends the methods discussed earlier.

4

Our contribution

Contents

4.1 Introduction

Building upon the findings discussed in the previous chapter, this chapter introduces the proposed two-phase approach developed to evaluate the impact of using Grad-CAM as an unsupervised ROI detection method in medical image classification. While supervised ROI detection techniques—such as segmentation and object detection—have shown strong performance when trained with annotated data, the use of unsupervised methods like Grad-CAM has been primarily limited to post-hoc interpretation, with limited quantitative evaluation of their direct impact on classification performance when employed during preprocessing as ROI detectors.

This chapter begins by detailing the data collection process and the experimental setup designed to ensure fair and consistent evaluation. In the **ROI Detection Phase**, Grad-CAM is employed to generate ROI-based versions of the original images without relying on any form of labeled supervision. Then, in the **Detected ROI Evaluation Phase**, we investigate whether training classification models on these ROI-transformed inputs—as opposed to full images—leads to measurable improvements in diagnostic performance across different backbone architectures and fine-tuning strategies. Rather than aiming to optimize model architectures or training procedures, the primary objective here is to determine whether using Grad-CAM as an unsupervised ROI detection technique results in improved classification accuracy when applied to chest X-ray and brain MRI datasets.

4.2 Experimental Environment and Tools

To implement our approach, we utilized a combination of local and cloud-based computing resources, along with widely adopted machine learning frameworks:

Hardware Setup

Our experiments were conducted using two main environments:

- **Local Workstation:** A high-performance computing station equipped with an Xeon Gold CPU with NVIDIA Quadro RTX 5000 GPU 32Gb RAM. This setup provided powerful computational capabilities for training deep learning models and was used at the Laghouat University Computer Science and Mathematics Laboratory.
- **Google Colab:** A cloud-based platform that offers free access to GPUs, enabling us to scale experiments and run models flexibly in a remote environment.

Software Environment

The experiments and model implementations in this study were conducted using widely adopted open-source machine learning frameworks:

- **TensorFlow:** An open-source platform for machine learning that simplifies the development and deployment of deep learning models [46].
- **Keras:** A high-level neural networks API, running on top of TensorFlow, which provides an intuitive and user-friendly interface for building and training models [47].
- **Python 3.10:** A versatile programming language widely used in scientific computing and machine learning, offering new syntax features and performance improvements beneficial for deep learning workflows.

4.3 Dataset Collection

To evaluate the robustness and generalizability of our models, we conducted experiments on multiple datasets. This approach allowed us to compare performance across varying data distributions and ensured that our results were not biased toward a single source.

1. **Brain Tumor :** We evaluated our models using the Brain Tumor Classification (MRI) dataset [48], which is publicly available under the MIT License. The dataset was originally curated by Mark Otto (2013) and later updated by Andrew Fong (2017). It contains T1-weighted contrast-enhanced MRI images, categorized into four tumor classes:
    - **Glioma tumor**
    - **Meningioma tumor**
    - **Pituitary tumor**
    - **No tumor**

2. **Chest X-Ray Multiclass Dataset :** We used the Chest X-Ray Multiclass Dataset with Non-X-ray Anomalies [49] to train and evaluate our classification models. However, in our experiments, we excluded the non-X-ray (anomaly) images to concentrate only on the medical categories:

   - **Normal**
   - **Pneumonia**
   - **Tuberculosis (TB)**

   This dataset combines chest X-rays from various global sources (Kermany, RSNA, NIAID, NLM, Belarus).

Data Preprocessing

To ensure consistent and fair evaluation across experiments, all datasets were split into training, validation, and test sets. Basic preprocessing steps were applied to align input dimensions with pretrained model requirements, without introducing complex augmentations.

Dataset Splits

To ensure fair comparisons across all model configurations and input types,Two different data splitting strategies were applied consistently throughout all experiments.

- **Chest X-ray Dataset (13,759 images):** approximately 85% of the images were allocated for training, around 10% for testing, and the remaining 5% for validation. This distribution was applied across all three categories Normal, Pneumonia, and Tuberculosis to ensure balanced learning and reliable performance evaluation.
- **Brain Tumor Dataset (7,023 images):** the data was divided to maintain a balanced and effective training process. Approximately 75% of the data was used for training, around 20% was allocated to testing, and the remaining 5% served as the validation set.

The training set was used to learn model parameters, the validation set supported hyperparameter tuning and overfitting control, and the test set was reserved for final performance assessment on unseen data.

Preprocessing

To prepare the images for input into the pretrained deep learning models, a set of basic preprocessing steps was applied consistently across both datasets. This was done to clearly observe the impact of the data itself—with and without ROI, without introducing additional effects from complex preprocessing steps.:

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

4.4 ROI Detection

The first phase of our approach focused on identifying the most effective backbone architectures for medical image classification and using them to detect diagnostically relevant regions via Grad-CAM-based ROI detection. By leveraging transfer learning, we utilized pre-trained CNN as fixed feature extractors, enabling faster convergence and improved generalization on limited medical datasets. This phase involved a thorough evaluation of each model's classification performance across multiple disease classes using standard metrics (recall 2.3, precision 2.2, and F1-score 2.4). The top-performing models were then selected for Grad-CAM-based ROI detection pipeline. That is built on the foundation of well-established architectures, as detailed in the following section. Figure 4.1 illustrates the overall workflow adopted in this phase.

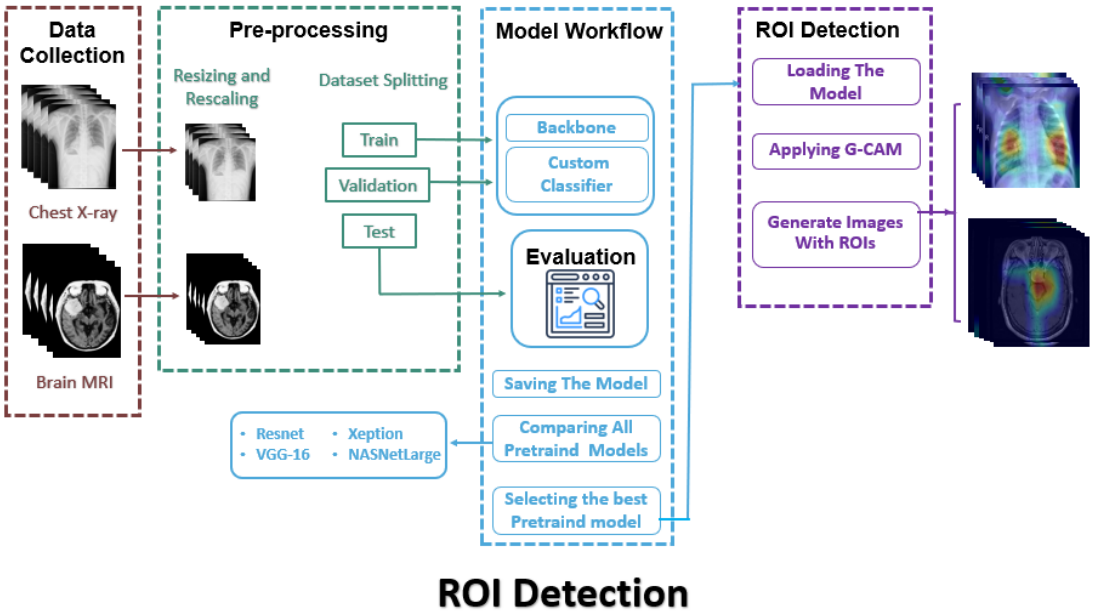


Figure 4.1: ROI Detection phase

4.4.1 Model Architecture Overview

The proposed architecture consisted of three main components: a pretrained convolutional backbone, a custom classification head, and a fixed training configuration.

Backbone Selection

We selected four pre-trained models as feature extractors: VGG16, Xception, NASNetLarge and ResNet50V2. which are trained on large-scale datasets like ImageNet. These models extract high-level features that improve training efficiency and performance—especially when working with limited medical data.

1. **VGG-16 :** is a convolutional neural network architecture (Figure 4.2) that was proposed by the Visual Geometry Group (VGG) at the University of Oxford. It is known for its deep structure, consisting of 16 layers.[30]

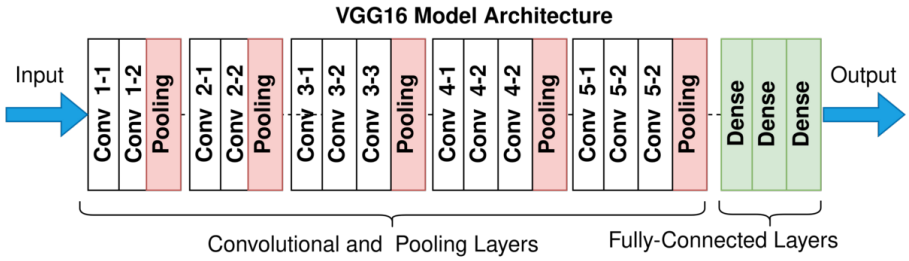


Figure 4.2: The VGG-16 architecture map.[12]

2. **Xception :** The Xception architecture (Figure 4.3) uses 36 convolutional layers grouped into 14 modules, most with residual connections. It is mainly built from depthwise separable convolutions.[50]

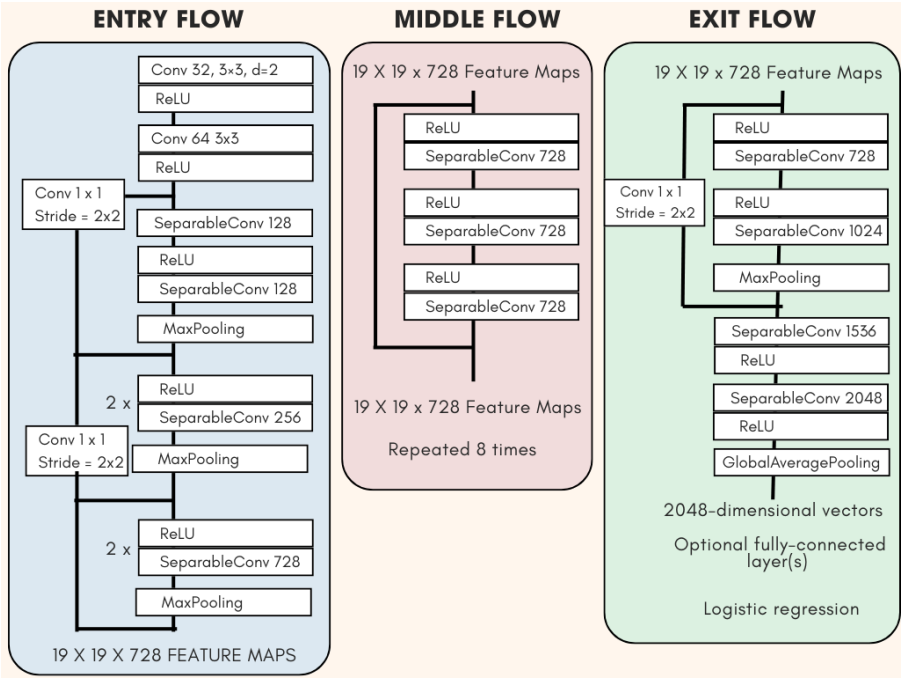


Figure 4.3: The Xception architecture map.

3. **NASNetLarge :** is a deep convolutional neural network architecture developed using Neural Architecture Search (NAS), a technique that automatically discovers the best model structure. NAS uses a controller RNN to search for optimal convolutional cells by training and evaluating multiple candidate networks. NASNetLarge is built by stacking these learned cells and is designed to scale efficiently to large datasets like ImageNet. Its architecture (Figure 4.4) includes two types of cells—normal and reduction—that are repeated throughout the network to handle high-resolution images.[51]

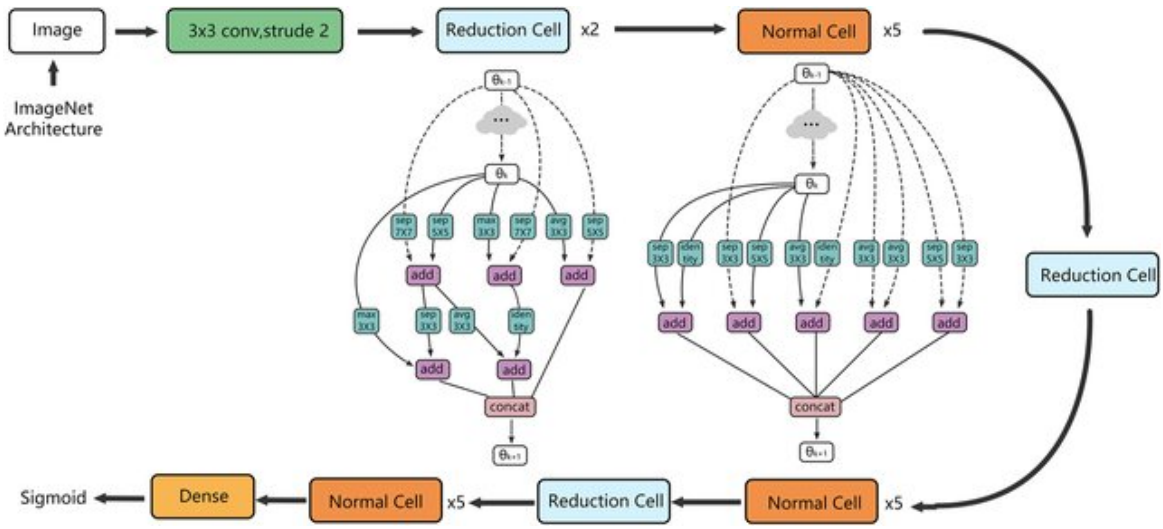


Figure 4.4: Network architecture of the improved NASNetLarge.[13]

4. **ResNet50V2 :**is a deep convolutional neural network that builds on the original ResNet architecture by introducing improvements like better normalization and residual connections. It uses a functional design where identity shortcuts (or skip connections) allow the network to learn residual mappings instead of direct functions.[29]

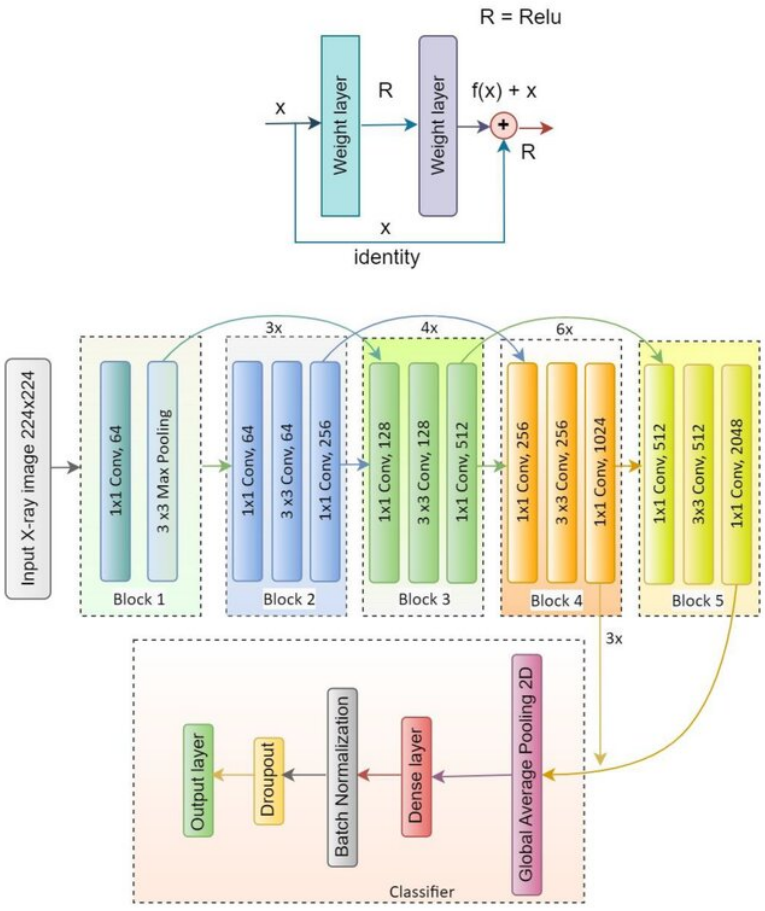


Figure 4.5: ResNet50V2 architecture. [14]

Custom Classification Head

After selecting the pretrained backbones, we proceeded with training on both the Chest X-ray and Brain Tumor datasets. Each model was used as a feature extractor by freezing its convolutional base and attaching a custom classification head (see Figure 4.6).

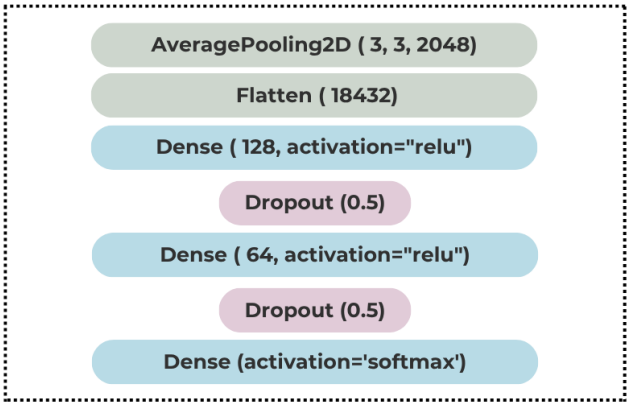


Figure 4.6: Custom Classification head.

4.4.2 Hyperparameter Configuration

Table 4.1: Training Configuration and Hyperparameters.

Parameter	**Value**
Image Size	224×224
Batch Size	8
Number of Epochs	30
Optimizer	Adam
Loss Function	Categorical Crossentropy
Dropout Rate	0.5
Preprocessing Technique	Rescaling (1./255)
Base Model	Frozen

The training was conducted using a fixed set of hyperparameters and settings, as detailed in Table 4.1. An input size of 224×224 was used to maintain sufficient resolution while ensuring computational efficiency. A batch size of 8 was chosen to balance memory constraints and model convergence. Training ran for 30 epochs to allow adequate learning. The Adam optimizer facilitated rapid convergence, while categorical crossentropy loss was appropriate for the multi-class classification tasks. A dropout rate of 0.5 was applied for regularization. Input images were rescaled to the [0, 1] range to stabilize gradient flow, which is particularly important for

producing meaningful Grad-CAM visualizations. No data augmentation was applied, ensuring consistency in model interpretability across original and ROI-focused inputs.

4.4.3 Obtained results and Discussion

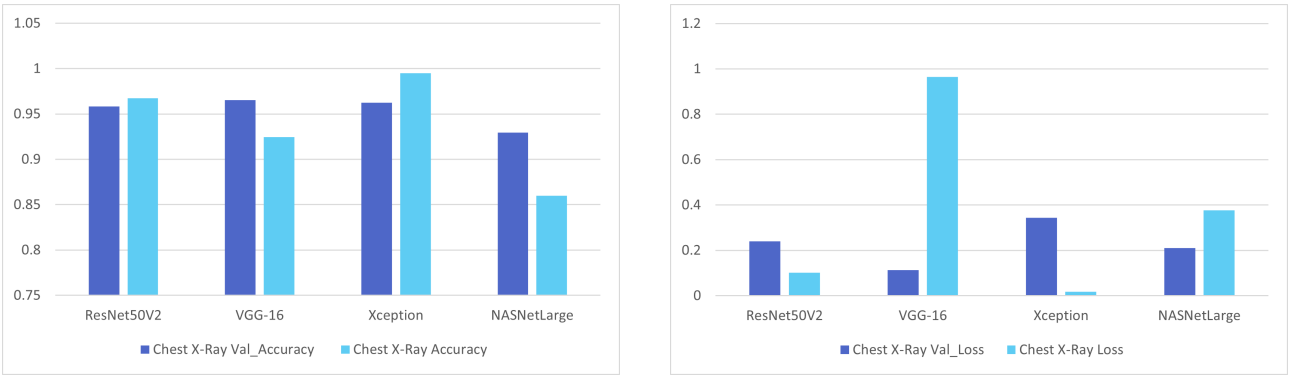
This section presents and interprets the results obtained from the conducted experiments across the chest X-ray and brain MRI datasets. It aims to evaluate model performance from multiple perspectives, including training behavior, classification accuracy, confusion matrices and per-class evaluation metrics. we assess the effectiveness of different backbone architectures and training strategies.

Training Analysis

To analyze learning behavior, we visualized the final training and validation accuracy and loss values for each model using grouped histograms across both datasets. This approach allows for a clear comparison of performance levels achieved at the end of training.

1. **Chest X-Ray Multiclass Dataset:** In (Figures 4.7a and 4.7b) Xception achieved the highest training accuracy (99.48%) and maintained a strong validation accuracy (96.25%), indicating excellent learning and generalization. It also had the lowest training loss (0.0181), though its higher validation loss (0.3441) hints at slight overfitting. VGG-16 demonstrated robust generalization, achieving the lowest validation loss (0.1129) while showing a higher validation accuracy (96.54%) than training accuracy (92.45%). This discrepancy may reflect regularization strategies such as dropout or early stopping.

   NASNetLarge displayed inconsistent training behavior, with a relatively low training accuracy (85.98%) and a surprisingly higher validation accuracy (92.94%), possibly influenced by augmentation or unstable convergence. The lower validation loss compared to training loss further supports this possibility.



(a) Final Accuracy for All Models

(b) Final Loss for All Models

Figure 4.7: Training and Validation Metrics for All Models.

2. **Brain Tumor Dataset:**

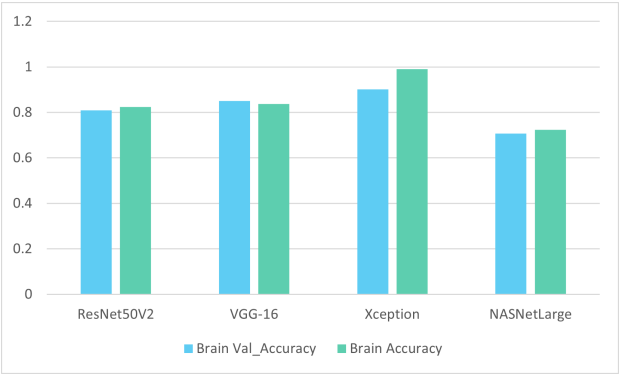
   According to (Figures 4.8a and 4.8b) Xception again outperformed the other models, with the highest training accuracy (99.04%) and validation accuracy (90.11%). Its exception-

ally low training loss (0.0369) suggests confident predictions, while the slightly elevated validation loss (0.4921) is indicative of potential overfitting.

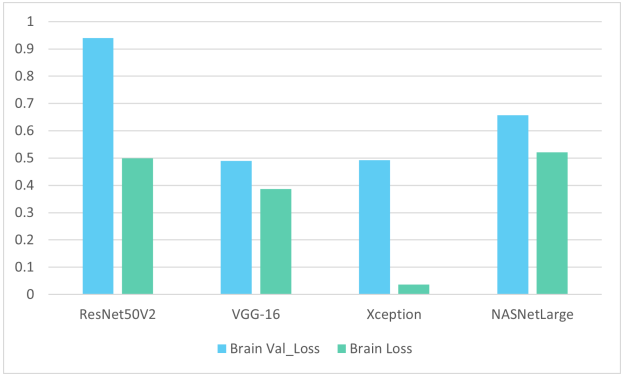
VGG-16 maintained balanced performance, with training and validation accuracies of 83.64% and 84.98%, respectively. It also achieved the lowest validation loss (0.4892), reinforcing its reliable generalization.

NASNetLarge lagged behind, with both training (72.35%) and validation (70.70%) accuracies being the lowest among all models. Its relatively high loss values (0.5206 for training and 0.6576 for validation) further demonstrate limited performance and suboptimal fit for this task.

Overall, Xception delivered the best results in terms of accuracy, while VGG-16 demonstrated stronger generalization, especially evident in its low validation loss. These outcomes highlight their suitability as backbone architectures for clinically interpretable and high-performing ROI detection models.



(a) Final Accuracy for All Models



(b) Final Loss for All Models

Figure 4.8: Training and Validation Metrics for All Models.

These histogram-based visualizations facilitated direct comparison of model performance at the end of training, highlighting relative effectiveness, convergence behavior, and signs of overfitting across the different CNN architectures on both datasets.

Testing Analysis

In the evaluation phase, the primary objective was to identify the most suitable backbone model for generating accurate and clinically relevant ROIs. To this end, four CNN architectures—ResNet50, VGG16, Xception, and NASNet—were assessed using the test sets of both the Chest X-ray and Brain Tumor datasets.

The comparative performance was assessed based on overall classification accuracy, loss, and computational efficiency (i.e., training time), as presented in Tables 4.2 and 4.3.

Table 4.2: Performance Summary of Backbone Models (Chest X-Ray).

Model	**Loss**	**Accuracy (%)**	**Training Time (min)**
ResNet50	0.22	96.00	210
VGG16	0.09	97.00	200
Xception	0.30	95.00	124
NASNet	0.23	91.00	250

- VGG16 emerged as the best-performing model with the highest accuracy (97%) and the lowest training loss (0.09), making it the most robust and stable choice for chest pathology detection.
- ResNet50 closely followed with 96% accuracy, although it had slightly higher loss (0.22) and longer training time.
- Xception had a shorter training time (124 min) but higher loss (0.30), suggesting faster convergence but less stability.
- NASNet, despite moderate training time, underperformed with 91% accuracy, making it the least favorable model in this domain.

Table 4.3: Performance Summary of Backbone Models (Brain).

Model	**Loss**	**Accuracy (%)**	**Training Time (min)**
ResNet50	0.34	88.00	350
VGG16	0.20	93.00	255
Xception	0.13	97.00	700
NASNet	0.48	77.00	100

- Xception achieved the best accuracy (97%) and lowest loss (0.13) but had a very high training time (700 min), making it computationally expensive but highly accurate.
- VGG16 again offered a strong balance, with 93% accuracy, low loss (0.20), and moderate training time (255 min).
- ResNet50 was less effective in this task, with 88% accuracy and higher loss.
- NASNet had the fastest training time but performed the worst, with 77% accuracy and highest loss (0.48).

Moreover, confusion matrices were utilized to provide deeper insight into class-specific performance and misclassification patterns (Figures 4.9 and 4.10).

- VGG-16 is the most consistent across both datasets, with strong class-wise performance.

- Xception shows excellent results on the brain dataset but slightly more confusion on chest images.
- ResNet50V2 maintains good overall accuracy with moderate confusion.
- NASNet has the weakest performance, particularly with high inter-class confusion.

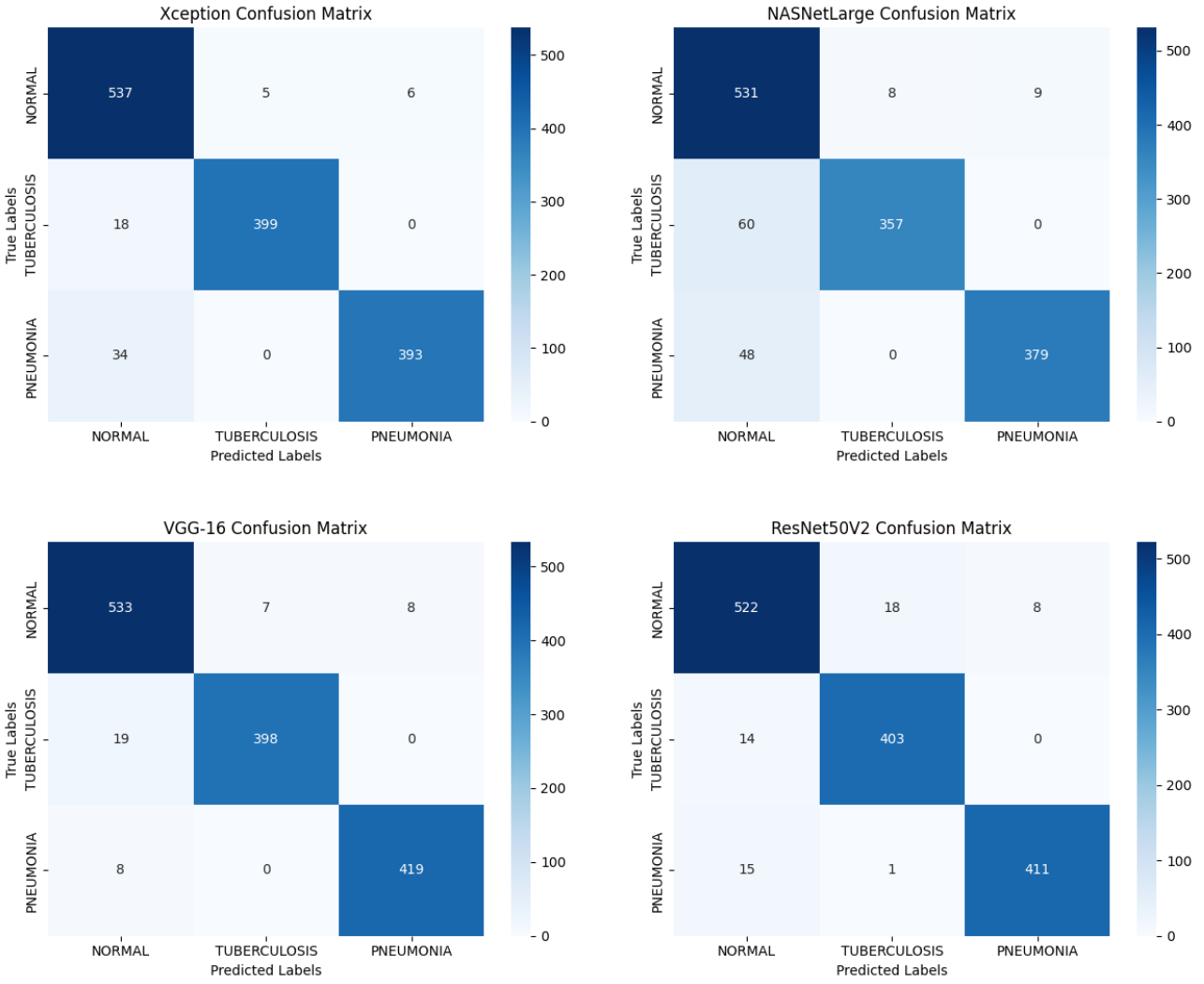


Figure 4.9: Confusion Matrices for All models (Chest X-Ray).

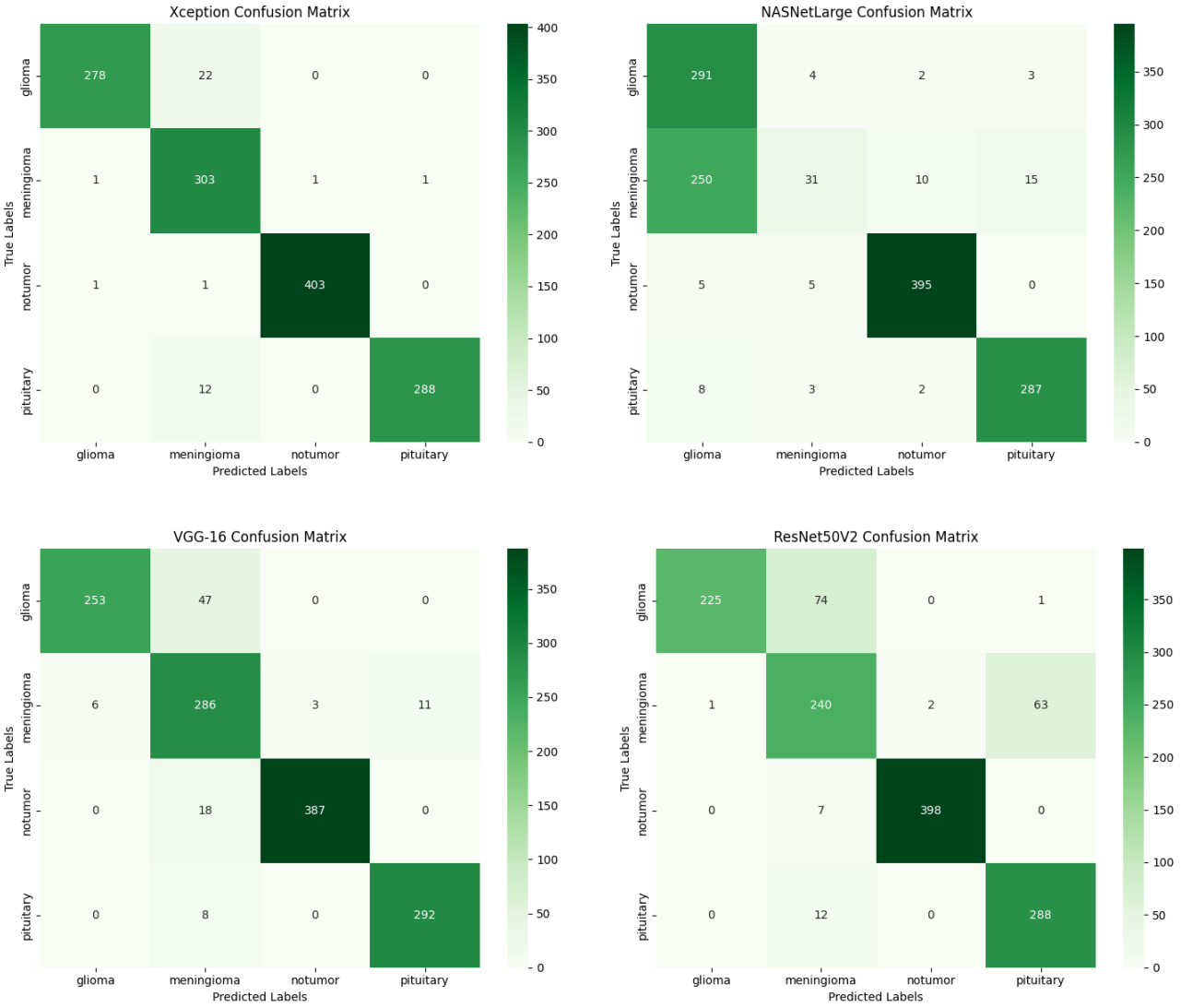
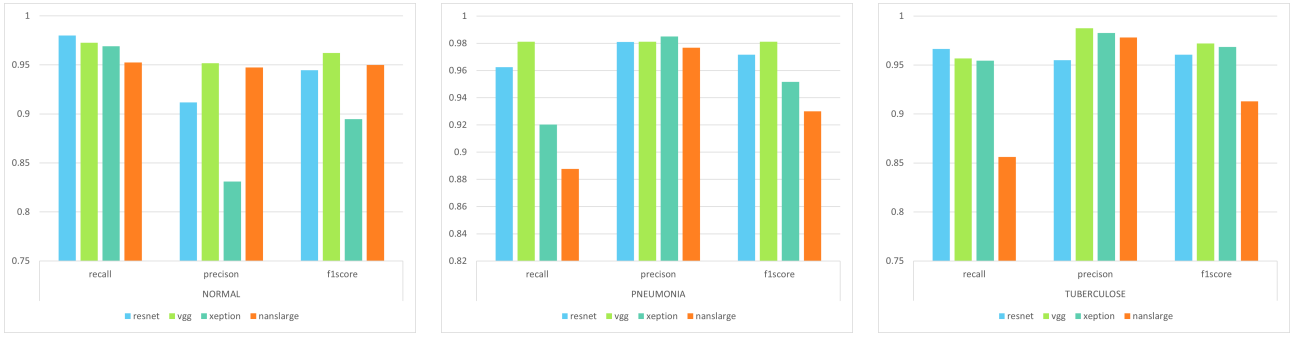


Figure 4.10: Confusion matrices for All models (Brain).

For a more granular evaluation, we computed and visualized class-wise Precision (PPV) 2.2, Recall (TPR) 2.3, and F1-score 2.4 using grouped bar charts, as shown in Figures 4.11 for the Chest X-ray dataset, and Figures 4.12 for the Brain Tumor dataset. These comparative visualizations revealed distinct differences in how well each model handled various pathologies, guiding the selection of the most suitable backbone for ROI detection in the subsequent stages of our pipeline.

1. **Chest X-Ray Multiclass Dataset :**



(a) Normal (b) Pneumonia (c) Tuberculosis

Figure 4.11: Grouped metriques for all models across the three classes.

Figure 4.11 compares the performance of four models (VGG16, ResNet50, NASNetLarge, and Xception) across three classes: Normal, Pneumonia, and Tuberculosis.

– **Normal Class:**

- **VGG16**: F1 = 96%, precision = 95%, recall = 97%. Most balanced performance with strong identification and minimal errors.
- **ResNet50**: F1 = 94%, precision = 91%, recall = 98%. High sensitivity but lower precision, indicating more false positives.
- **NASNetLarge**: F1 = 95%, precision and recall balanced. Performed well overall without leading in any single metric.
- **Xception**: F1 = 89%, precision = 83%, recall = 97%. Strong recall but low precision suggests over-detection and higher false positive rate.

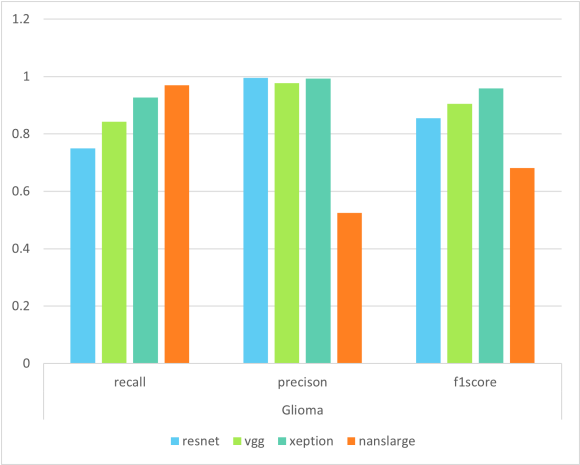
– **Pneumonia Class:**

- **VGG16**: F1 = 98%, precision = 98%, recall = 98%. Delivered highest accuracy across all metrics, confirming reliable detection.
- **ResNet50**: F1 = 97%, strong all-around performance, closely following VGG16.
- **NASNetLarge**: F1 = 93%, precision = 98%, recall = 89%. High precision offset by reduced recall, missing more true positives.
- **Xception**: F1 = 95%, precision = 98%, recall = 92%. Similar pattern to NASNetLarge—high accuracy in confirmed cases, but slightly less sensitive.

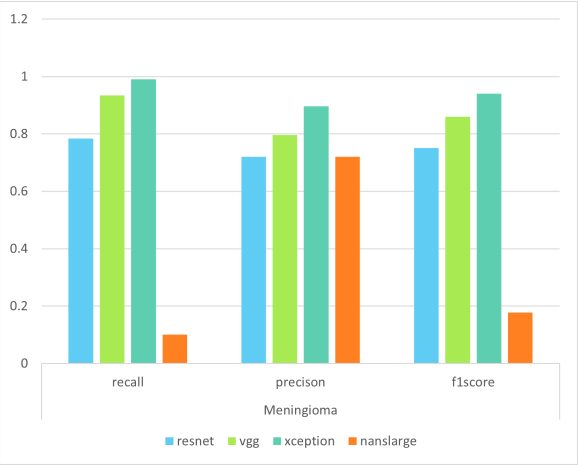
– **Tuberculosis Class:**

- **VGG16**: F1 = 97%, precision = 99%, recall = 96%. Highest overall score, combining excellent detection and minimal false positives.
- **Xception**: F1 = 97%, solid balance of metrics, closely matching VGG16 in classification quality.
- **ResNet50**: F1 = 96%, consistent and stable performance across all metrics.
- **NASNetLarge**: F1 = 91%, precision = 98%, recall = 86%. Excellent precision but reduced sensitivity, resulting in lower F1.

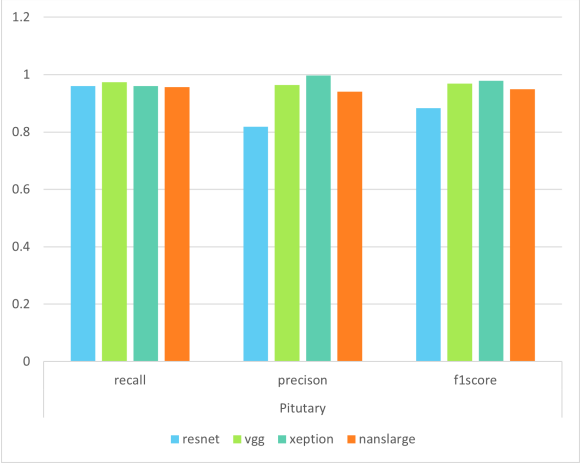
2. **Brain Tumor :**



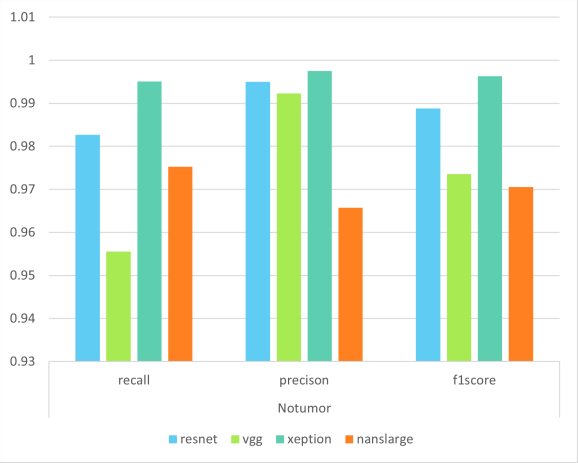
(a) Glioma



(b) Meningioma



(c) Pituitary



(d) No Tumor

Figure 4.12: Grouped metriques for all models across the four brain tumor classes.

Figure 4.12 illustrates the performance of four models across the Glioma, Meningioma, Pituitary, and No Tumor classes.

– **Glioma Class:**

- **Xception**: F1 = 96%, precision = 99%, recall = 93%. Achieved the best overall performance, balancing high sensitivity and precision.
- **VGG16**: F1 = 91%. Solid and consistent results with good precision-recall balance.
- **ResNet50**: F1 = 86%, precision = 99%, recall = 75%. High precision, but low recall affected overall score.
- **NASNetLarge**: F1 = 68%, precision = 53%, recall = 97%. High sensitivity but very poor precision resulted in many false positives.

– **Meningioma Class:**

- **Xception**: F1 = 94%, precision = 90%, recall = 99%. Best performance, accurately detecting nearly all true cases.
- **VGG16**: F1 = 86%. Delivered a balanced but lower result compared to Xception.
- **ResNet50**: F1 = 75%. Moderate performance with no standout metric.
- **NASNetLarge**: F1 = 18%, recall = 10%. Extremely low detection ability, showing severe underperformance.

– **Pituitary Class:**

- **Xception**: F1 = 98%, precision = 99.7%, recall = 96%. Outstanding results, leading in both precision and overall accuracy.
- **VGG16**: F1 = 97%. Strong and reliable classification.
- **NASNetLarge**: F1 = 95%. Maintained high-level accuracy.
- **ResNet50**: F1 = 88%, precision = 82%. Lowest score due to reduced precision.

– **No Tumor Class:**

- **NASNetLarge**: F1 = 99.6%. Best performance across all models in identifying normal brain cases.
- **ResNet50**: F1 = 99%. Excellent and consistent.
- **Xception & VGG16**: F1 = 97%. Both models demonstrated strong agreement in detecting healthy samples.

4.4.4 ROI Detection Using Grad-CAM

After a comprehensive evaluation across both datasets, VGG16 was identified as the most effective model for chest X-ray classification, demonstrating a strong balance between accuracy and generalization. In contrast, Xception consistently outperformed other models in brain tumor classification, achieving the highest F1-scores and maintaining a favorable precision-recall trade-off across different tumor categories. As a result, VGG16 was selected for chest X-ray ROI detection and Xception for brain MRI ROI detection within the Grad-CAM-based Region of Interest detection pipeline. This pipeline, illustrated in Figure 4.13, consists of four key stages: input image forwarding, feature map detection, gradient computation, and heatmap generation with overlay.

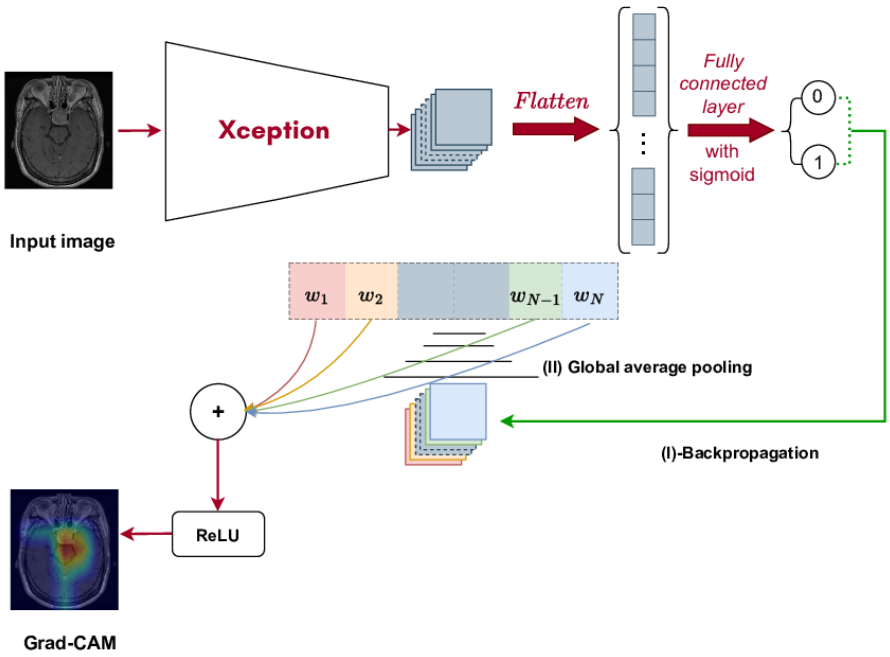


Figure 4.13: Grad-CAM pipeline.

This pipeline follows a structured sequence of computational steps centered around ROI detection,aimed at evaluating the impact of region-focused inputs on downstream classification performance. The methodology is presented in four principal stages:

1. **Image Preprocessing**

   - Image pixel values are normalized according to the preprocessing routine specific to the selected architecture (e.g., `xception.preprocess_input` for the Xception model).
   - Each image is converted into a four-dimensional tensor format (batch size, height, width, channels), ensuring compatibility with Keras/TensorFlow inference and visualization workflows.

2. **Grad-CAM Heatmap Generation**

   - A modified gradient model is instantiated, capturing both the activations of the final convolutional layer and the class-specific output logits.
   - The gradient of the predicted class score is computed with respect to the feature maps of the final convolutional layer.
   - These gradients are then subjected to global average pooling, producing a set of scalar importance weights corresponding to each feature channel.
   - A class activation map (heatmap) is generated by performing a weighted sum over the feature maps using these importance weights. This map is subsequently passed through a ReLU activation to retain only positively contributing features.
   - The heatmap is normalized to the range [0, 1], and a thresholding step (e.g., values below 0.3 are suppressed) is optionally applied to reduce background noise and sharpen the focus on discriminative regions.

3. **Heatmap Visualization and Fusion**

   - The generated heatmap is resized to match the dimensions of the original image for accurate spatial correspondence.
   - A color mapping is applied using OpenCV's `COLORMAP_JET` to enhance visual interpretability.
   - The colorized heatmap is then overlaid on the original image using alpha blending (transparency factor $\alpha = 0.3$), yielding a composite image that highlights the regions most influential to the model's decision.

4. **Dataset Generation for ROI-Based Classification**

   - The resulting Grad-CAM-enhanced images are saved into a structured directory hierarchy that mirrors the original dataset's organization.
   - These enhanced samples are subsequently employed as input data for a second-stage classification model, allowing for quantitative assessment of the ROI-driven approach.
   - Example samples from the Grad-CAM-generated dataset are shown in Figure 4.14, illustrating how salient regions were detected and emphasized for both chest X-ray and brain MRI images. The top row displays chest X-rays, while the bottom row presents brain MRI samples.

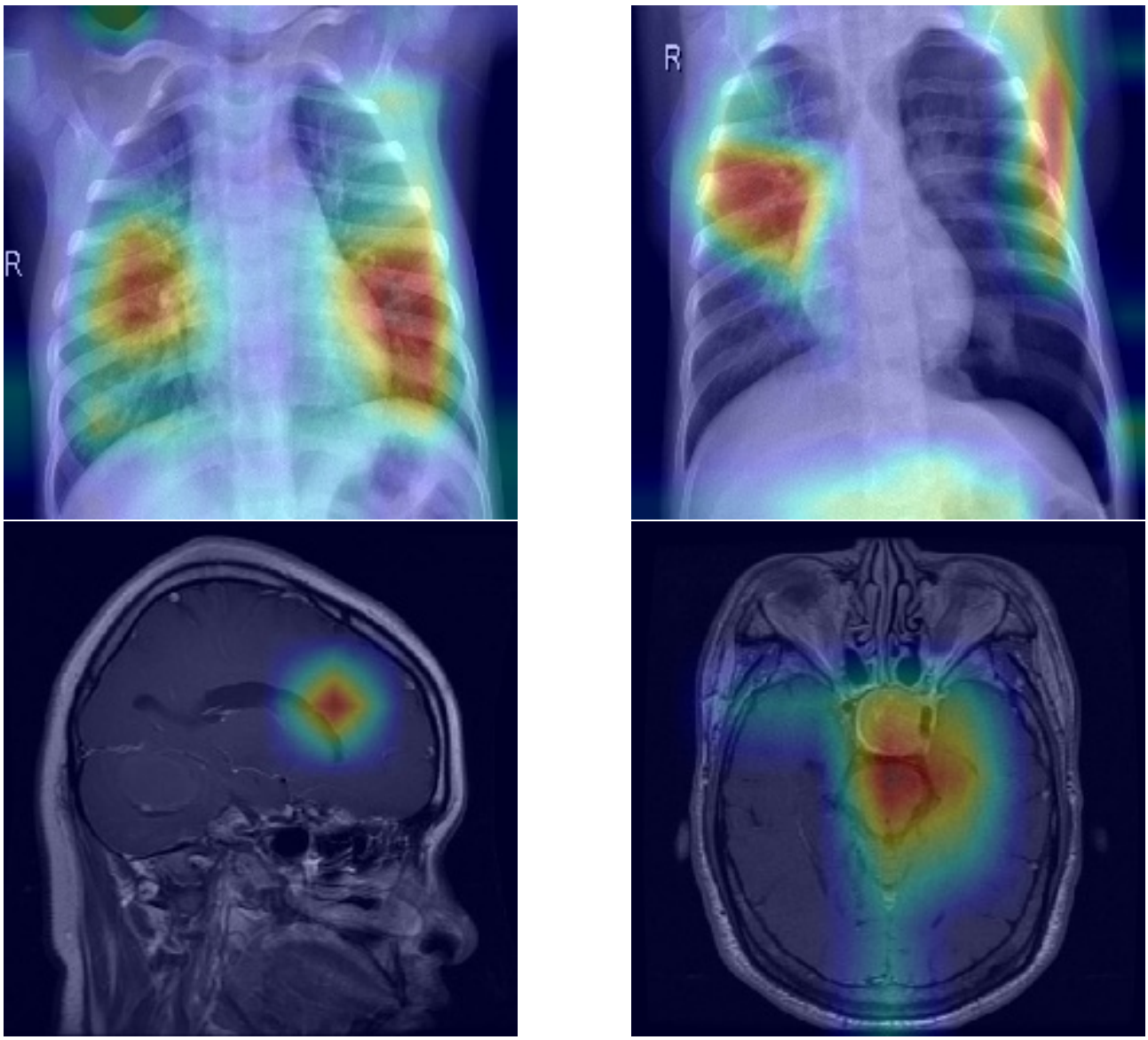


Figure 4.14: Grad-CAM-enhanced images for ROI-based classification.

This ROI-focused processing pipeline is designed to enable empirical evaluation of how emphasizing detected features affects classification outcomes. By generating modified datasets in which salient regions are highlighted, the approach facilitates a systematic investigation into whether such modifications influence model accuracy and robustness.

4.5 Detected ROI Evaluation

After generating region of interest data using Grad-CAM in the first phase, this phase focused on evaluating the impact of these ROI-based inputs on classification accuracy. Specifically, we trained models using the Grad-CAM-processed images and compared their classification results to those obtained using the full original images. This evaluation was conducted on two medical imaging datasets: chest X-rays and brain MRIs. All experiments relied on a consistent architecture that integrated a pretrained backbone, a task-specific classifier, and two

fine-tuning strategies tailored for medical image analysis. This phase involved assessing classification metrics under each configuration to determine the effectiveness of ROI-based training. Figure 4.15 illustrates the overall workflow adopted in this phase.

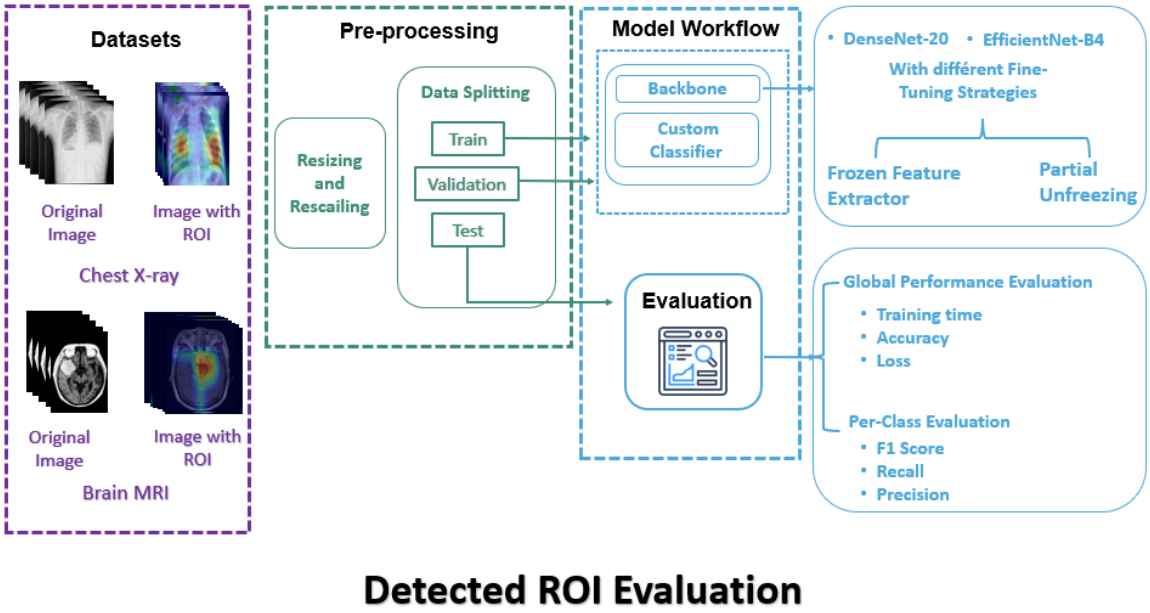


Figure 4.15: Evaluating Detected ROI phase.

4.5.1 Proposed Architecture

The proposed architecture consisted of three main components: a pretrained convolutional backbone, two distinct fine-tuning strategies, and a custom classification head. This structure allowed us to combine the strengths of transfer learning with domain-specific adaptation, ensuring fair and consistent evaluation across input types and medical datasets.

Backbones Selection

we selected two widely used convolutional neural networks: EfficientNetB4 and DenseNet201. Both are pretrained on ImageNet and known for their strong performance and efficiency in medical image tasks. Their inclusion allows us to evaluate how different backbone architectures affect classification performance under varied fine-tuning strategies and input types.

- EfficientNet-B4 : is part of the EfficientNet family of models. The main building block in EfficientNet is called MBConv, and it also includes a feature called squeeze-and-excitation, which helps the model focus on the most important parts of the image. To build larger versions like EfficientNet-B4 Figure 4.16, the creators used a compound scaling method. This method increases the model's depth, width, and input image size in a balanced way, helping to improve accuracy without requiring too much memory or computation time.[52]

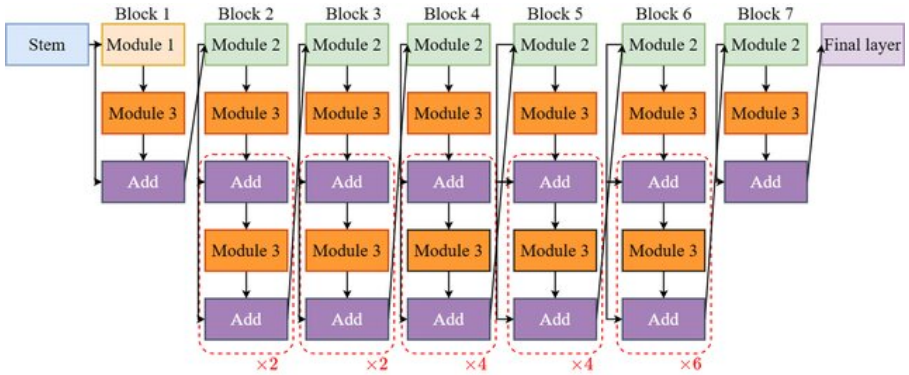


Figure 4.16: EfficientNet-B4 architecture.[15]

- DenseNet-201 : is a convolutional neural network where each layer connects to all previous layers, improving feature reuse and information flow. It uses dense blocks with batch normalization, ReLU, and convolution layers, along with transition layers to reduce size. This design Figure 4.17 makes DenseNet efficient and accurate with fewer parameters.[16]

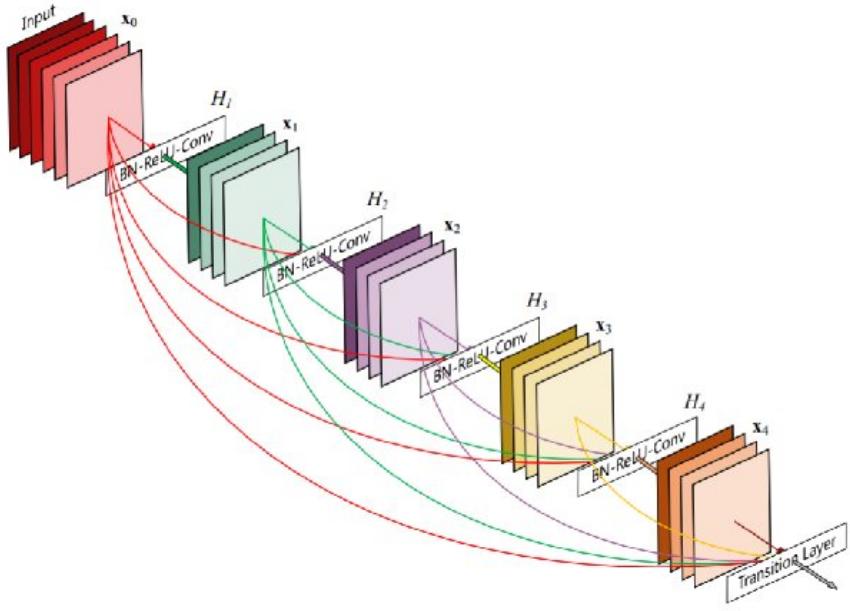


Figure 4.17: A visualization of the DenseNet-201 architecture.[16]

Fine-Tuning Strategies

To evaluate the impact of region of interest data and determine the most effective fine-tuning strategy for our specific medical imaging tasks, we implemented two distinct fine-tuning approaches using pretrained convolutional neural networks (EfficientNetB4 and DenseNet201):

1. **Frozen Feature Extractor:** In this strategy, all layers of the pretrained backbone were frozen, ensuring their weights remained unchanged during training. Only the custom classification head, initialized with random weights, was trainable. By restricting training to the classification head, we reduced the risk of overfitting and decreased computational requirements.

2. **Partial Unfreezing:** Recognizing that our medical imaging datasets—comprising chest X-rays and brain MRIs—contained domain-specific features not present in natural images, we adopted a more flexible adaptation strategy. Specifically, we unfroze the final 25% of the pretrained backbone layers, allowing the model to adjust its high-level feature representations to better capture the specialized patterns found in medical scans. This approach was grounded in the understanding that while the early layers of convolutional networks extract general-purpose features applicable across domains, the deeper layers benefit from adaptation to the target task. Unfreezing only the upper 25% of layers offered a balanced trade-off: it enabled effective domain-specific learning while maintaining the stability of the pretrained lower layers and reducing computational cost compared to full fine-tuning.

Since medical images (chest X-rays and brain MRIs) contain distinct domain-specific features that differ significantly from natural images, we unfroze only the final 25% of the pretrained model's layers. This strategy ensures that early layers retain their ability to detect general patterns (e.g., edges and textures), while deeper layers adapt to medical-specific details. By unfreezing only the upper 25% of layers , we balance task specialization with computational efficiency, avoiding the excessive resource demands of full network retraining.

Custom Classification Head Design

We used a lightweight custom classification head to adapt the pretrained backbone to our task. It began with a `GlobalAveragePooling2D` layer, which replaced the traditional `Flatten` layer and offered better generalization by reducing spatial dimensions without introducing dense parameter connections. This was followed by a dense layer with 256 units and `GELU` activation, which performed better than ReLU in medical imaging due to its smoother activation and improved gradient flow. The final output layer used softmax activation to handle multi-class predictions.

Hyperparameter Configuration

All models—whether trained on full images or Grad-CAM-based ROI data were trained using the same configuration for consistency. Table 4.4 summarizes the key parameters.

Table 4.4: Training hyperparameters used across all experiments.

Parameter	**Value**
Batch Size	8
Number of Epochs	20
Optimizer	Adam
Loss Function	Categorical Crossentropy
Dropout Rate	0.5
Preprocessing Technique	Rescaling (1./255)
Data Augmentation	None

A `batch size of 8` was chosen to fit GPU memory while preserving training stability. Training for `20 epochs` provided a balance between allowing sufficient learning and avoiding overfitting. We used the `Adam optimizer` for its efficiency in converging quickly on complex tasks, which was beneficial in medical imaging where model depth and input resolution were relatively high. `Categorical crossentropy` was appropriate for multi-class classification with softmax outputs. A `dropout rate of 0.5` was applied to prevent overfitting by randomly deactivating neurons during training. Finally, `rescaling inputs to [0, 1]` ensured stable gradient updates, and `no data augmentation` was applied to maintain consistency when comparing original and ROI-based data.

4.5.2 Obtained results and Discussion

This section presents a detailed evaluation of the experimental results obtained from training and testing the proposed models on chest X-ray and brain MRI datasets. The analysis includes training and validation behavior, test set performance, and per-class evaluation metrics. Special attention is given to the role of ROI-based preprocessing using Grad-CAM, an unsupervised technique, and its comparative effectiveness against standard full-image training. By examining the influence of backbone architecture, fine-tuning strategy, and input representation, we aim to determine which configurations yield the most reliable and diagnostically useful performance.

Training Analysis

To better understand the learning dynamics of the proposed models, we analyze the training and validation accuracy and loss curves across all experimental configurations. These include two backbone architectures (EfficientNetB4 and DenseNet201), each trained using both Frozen Feature Extractor and partially unfrozen strategies, and evaluated on original full images and Grad-CAM-based ROI inputs. All models were trained for 20 epochs using consistent procedures. The training behavior of each configuration is illustrated in Figure 4.18 for DenseNet201 and Figure 4.19 for EfficientNetB4.

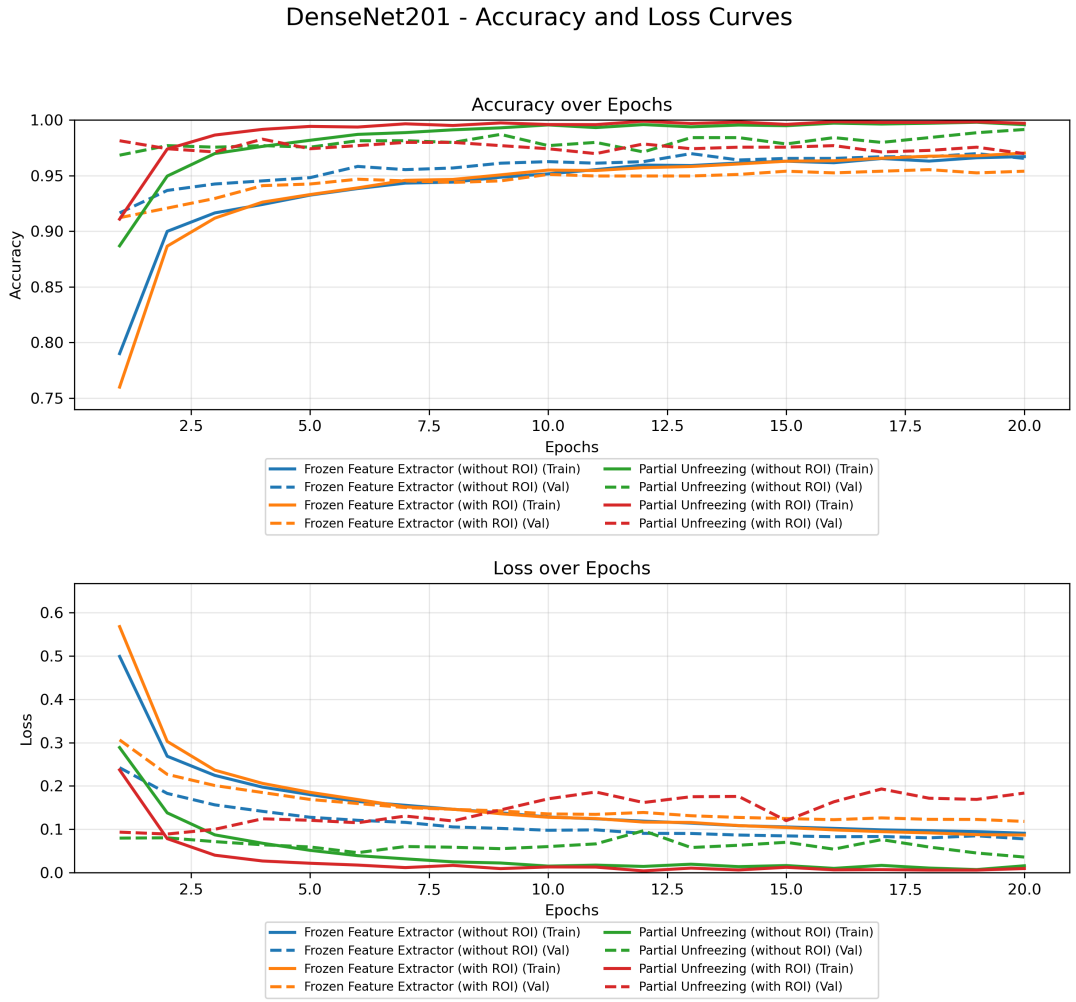


Figure 4.18: Training and validation accuracy/loss curves for DenseNet201 across all configurations.

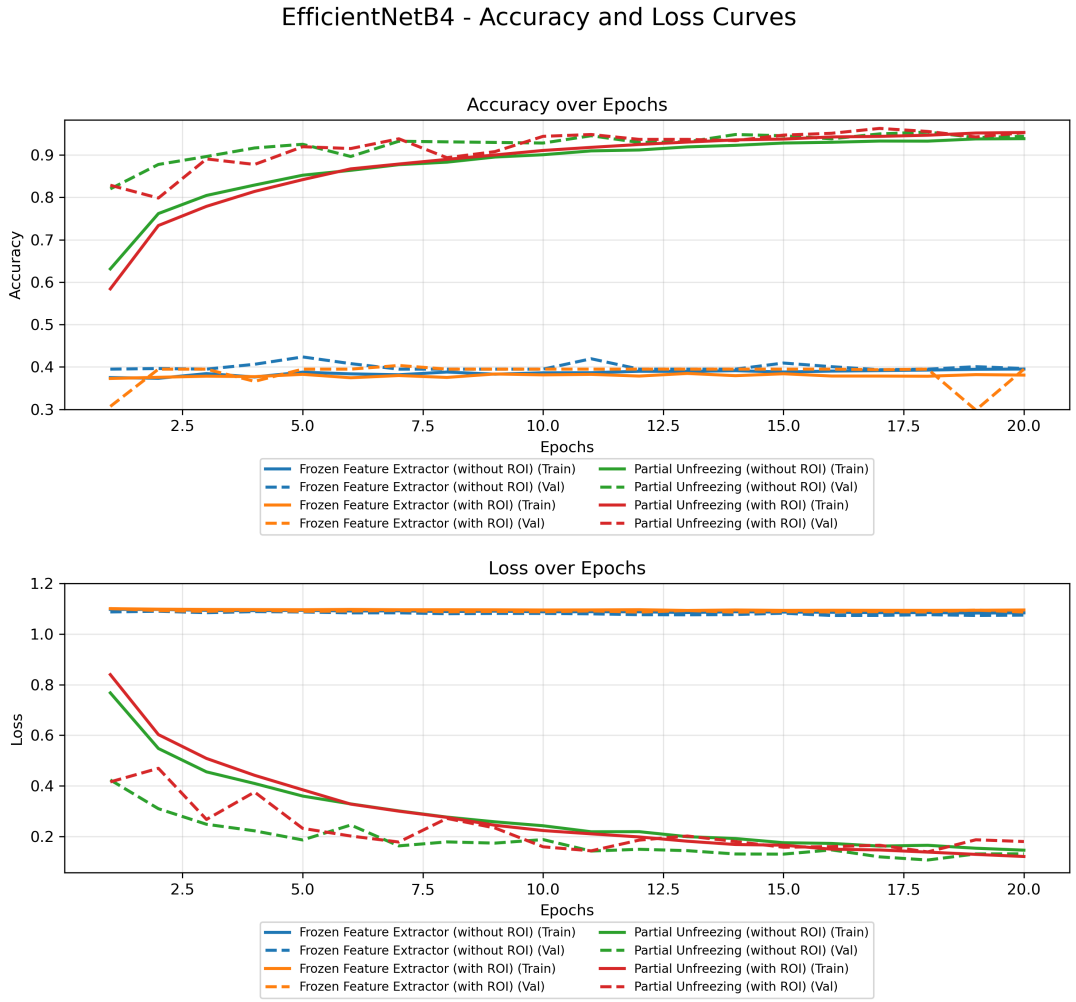


Figure 4.19: Training and validation accuracy/loss curves for EfficientNetB4 across all configurations.

- Models with partially unfrozen backbones, where only the top 25% of layers were trainable, generally exhibited faster convergence and better generalization than their frozen counterparts. This supports the idea that selectively adapting high-level features to the domain of medical imaging enhances performance without compromising the benefits of transfer learning. **DenseNet201** with partial unfreezing on original images achieved the highest validation accuracy—reaching over **99.00%**, followed closely by EfficientNetB4 under the same configuration with accuracy exceeding **95.00%**.
- Between the two architectures, **DenseNet201** consistently delivered superior results, with lower validation losses and more stable accuracy progression throughout training.
- Regarding input types, training on the full original images outperformed the ROI-based data in most cases. While ROI inputs were intended to emphasize diagnostically relevant areas, they may have reduced the availability of broader contextual features essential for robust classification. This effect was evident in the frozen configurations, where models

trained on original images yielded better validation accuracy—for instance, EfficientNetB4 attained approximately **42.00%**, compared to **39.00%** using ROI inputs.

These observations highlight the influence of backbone selection, fine-tuning depth, and input representation on training behavior in medical image classification tasks.

Testing Analysis

To assess the generalization capability of the trained models, we conducted a comprehensive evaluation on the test set. This analysis covers both overall performance and class-specific behavior. Table 4.5 presents a comparative summary of test accuracy, loss, and total training time across all evaluated configurations.

For more detailed performance insights, we computed class-level precision 2.2, recall 2.3, and F1-score 2.4, which are visualized as grouped bar plots. Figure 4.20 shows the evaluation results for the Chest X-ray dataset, while Figure 4.21 illustrates performance on the Brain MRI dataset. These visualizations enable clear comparisons between model architectures, fine-tuning strategies (S1: Frozen backbone, S2: Partially Unfrozen), and input types (original vs. ROI-based).

Table 4.5: Test accuracy, loss, and training time for all model configurations on both datasets.

Model Configuration	**Accuracy**	**Loss**	**Training Time (min)**
Chest X-ray Dataset			
EfficientNetB4 (No ROI, S1)	40.00%	1.0700	83
EfficientNetB4 (ROI, S1)	39.00%	1.0700	66
EfficientNetB4 (No ROI, S2)	94.00%	0.1800	400
EfficientNetB4 (ROI, S2)	94.00%	0.1800	470
DenseNet201 (No ROI, S1)	96.00%	0.0930	170
DenseNet201 (ROI, S1)	94.00%	0.1500	85
DenseNet201 (No ROI, S2)	**98.00%**	**0.1000**	**100**
DenseNet201 (ROI, S2)	96.00%	0.2800	100
Brain MRI Dataset			
EfficientNetB4 (No ROI, S1)	52.00%	1.1500	30
EfficientNetB4 (ROI, S1)	48.00%	1.2400	35
EfficientNetB4 (No ROI, S2)	95.00%	0.1400	42
EfficientNetB4 (ROI, S2)	91.00%	0.2400	35
DenseNet201 (No ROI, S1)	92.00%	0.2000	35
DenseNet201 (ROI, S1)	90.00%	0.2700	38
DenseNet201 (No ROI, S2)	**99.00%**	**0.0200**	**40**
DenseNet201 (ROI, S2)	97.00%	0.0900	33

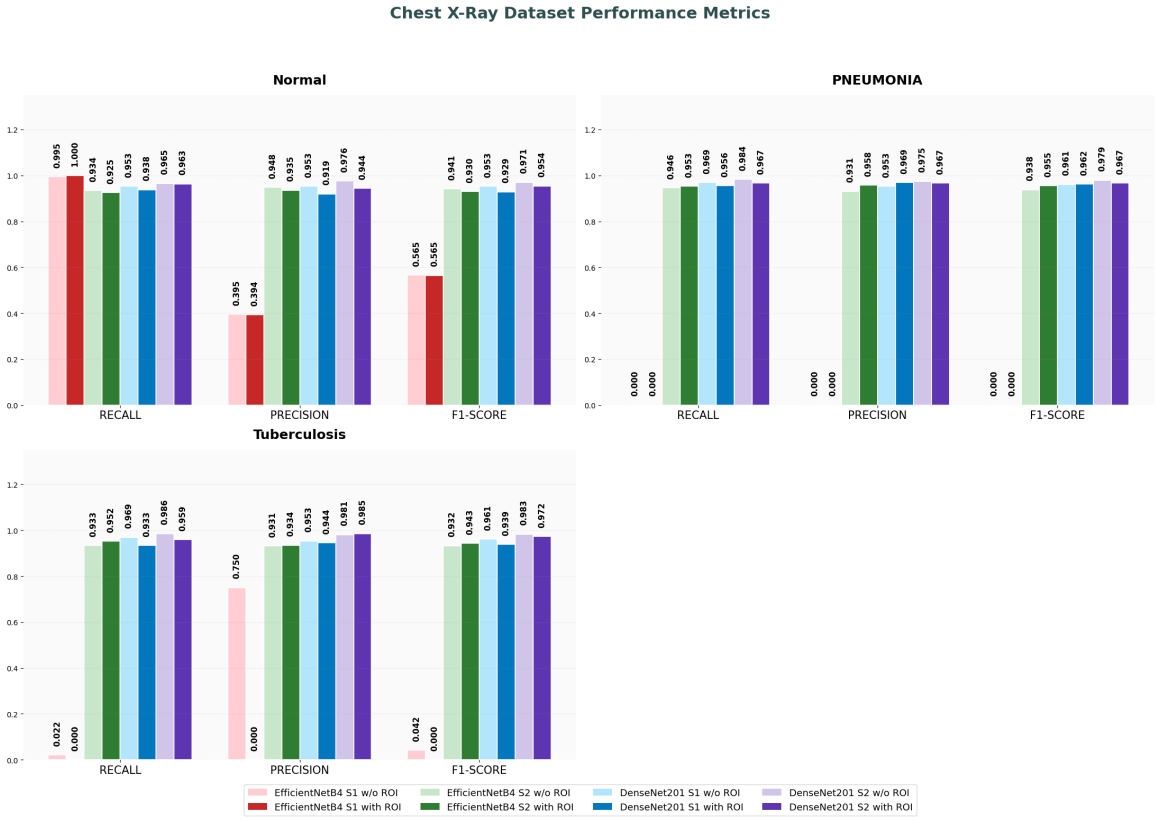


Figure 4.20: Class-wise metrics for Chest X-ray classification across all model configurations.

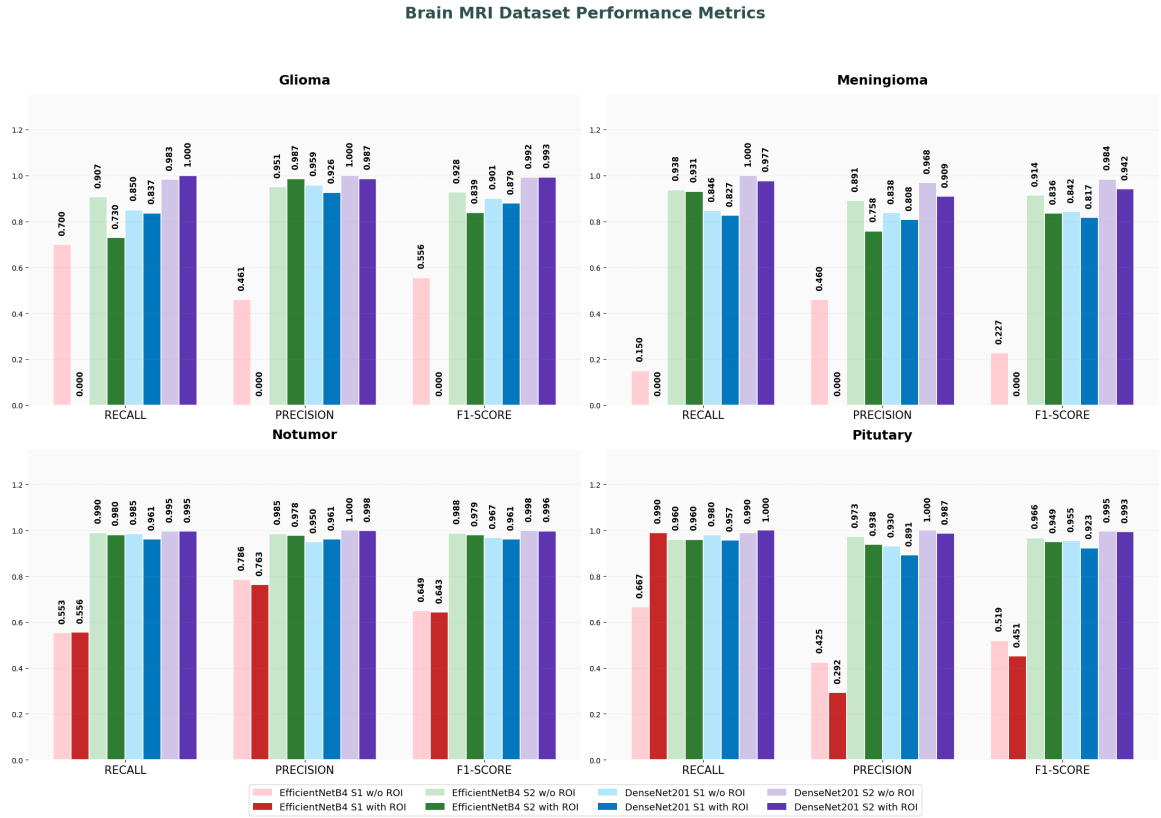


Figure 4.21: Class-wise metrics for Brain MRI classification across all model configurations.

Global Performance Evaluation

DenseNet201 with fine-tuning and no ROI consistently showed the best overall performance across both datasets. It reached the highest accuracy—**98.00%** for Chest X-ray with a training time of 100 minutes, and **99.00%** for Brain MRI with a training time of 40 minutes—while maintaining low loss values of **0.1000** and **0.0200**, respectively. In contrast, frozen models—especially those using Grad-CAM-based ROIs—exhibited noticeably lower performance, both in accuracy and training stability. This suggests that using full image input, combined with an adaptive fine-tuning strategy, provides the most effective learning setup in terms of both performance and efficiency for medical image classification tasks.

Per-Class Evaluation with Metric Interpretation

To gain deeper insight into the classification behavior of each model, we analyze class-level performance using precision, recall, and F1-score. These metrics, derived from confusion matrices, provide a detailed understanding of how effectively each disease class is identified. This analysis also allows us to assess how ROI-based preprocessing influences detection performance, particularly in terms of reducing false positives and improving sensitivity across different input types and training strategies.

- **Chest X-ray Dataset:**
  - **Normal Class:** DenseNet201 with fine-tuning and no ROI achieved the best results with **F1 = 97.06%**, **precision = 97.60%**, and **recall = 96.53%**, reflecting highly

accurate identification of healthy lungs with minimal false positives or negatives. Using ROI slightly reduced these values (**F1 = 95.38%**, **precision = 94.43%**, **recall = 96.34%**), indicating a minor loss of contextual information. Frozen models like EfficientNetB4 without ROI underperformed significantly (**F1 = 56.54%**).

- **Pneumonia Class:** DenseNet201 with fine-tuning and no ROI achieved the highest performance with **F1 = 97.90%**, **precision = 97.45%**, **recall = 98.36%**, indicating excellent detection of pneumonia with low misclassification. With ROI, the values slightly dropped to **F1 = 96.72%**, precision and recall = **96.72%**, suggesting some contextual loss. Frozen EfficientNetB4 models yielded **F1 = 0.00%**, highlighting the necessity of fine-tuning.
- **Tuberculosis Class:** DenseNet201 with fine-tuning and no ROI achieved top scores: **F1 = 98.33%**, **precision = 98.09%**, **recall = 98.56%**, indicating precise and sensitive TB detection. With ROI, performance dropped slightly to **F1 = 97.21%**, **precision = 98.52%**, and **recall = 95.92%**, suggesting a trade-off in contextual detail. Frozen configurations, especially EfficientNetB4 with ROI, showed very low F1 scores, down to **0.00%**.

- **Brain MRI Dataset:**
  - **Glioma Class:** DenseNet201 with fine-tuning and no ROI yielded **F1 = 99.16%**, **precision = 100.00%**, **recall = 98.33%**, denoting highly accurate tumor detection. With ROI, performance was comparable: **F1 = 99.34%**, **precision = 98.68%**, **recall = 100.00%**, showing excellent balance and no missed cases. In contrast, frozen EfficientNetB4 with ROI showed **F1 = 0.00%**.
  - **Meningioma Class:** DenseNet201 with fine-tuning and no ROI attained **F1 = 98.39%**, **precision = 96.84%**, **recall = 100.00%**, reflecting high reliability and no missed tumors. With ROI, values decreased to **F1 = 94.17%**, **precision = 90.88%**, **recall = 97.71%**, likely due to increased false positives.
  - **Notumor Class:** DenseNet201 with fine-tuning and no ROI reached nearly perfect results with **F1 = 99.75%**, **precision = 100.00%**, **recall = 99.51%**, ensuring accurate healthy brain classification. ROI-based inputs showed similar performance: **F1 = 99.63%**, **precision = 99.75%**, **recall = 99.51%**, indicating ROI had minimal impact for this class.
  - **Pituitary Class:** DenseNet201 with fine-tuning and no ROI achieved **F1 = 99.50%**, **precision = 100.00%**, **recall = 99.00%**, indicating excellent precision and sensitivity. With ROI, the model preserved high performance with **F1 = 99.34%**, **precision = 98.68%**, **recall = 100.00%**, confirming robustness across input types.

The observation that models trained on full images outperformed those using Grad-CAM-based ROIs may be attributed to several factors:

- **Incorrect Focus Areas:** Grad-CAM highlights the regions that the model relies on for its decision-making, but these areas may not truly represent the most relevant visual features for classification. The model might attend to unrelated patterns that do not contribute meaningfully to class separation.

- **Loss of Useful Information:** By restricting input to only the Grad-CAM-highlighted regions, other potentially valuable visual details are excluded. This may limit the model's capacity to learn diverse and informative representations necessary for accurate classification.

These insights suggest that while Grad-CAM is useful for visualizing model attention, its role as a standalone unsupervised ROI detection tool may be limited in enhancing classification performance without additional supervision or refinement.

4.6 Conclusion

This chapter investigated the effectiveness of Grad-CAM-based region of interest inputs, which were detected without ROI-level supervision, in enhancing classification performance for medical imaging tasks involving chest X-rays and brain MRIs. A two-phase experimental approach was adopted: first, generating ROI-enhanced images using Grad-CAM, and second, evaluating these against original full images across different backbone architectures and fine-tuning strategies.

To ensure fair evaluation, the chapter first outlined the data collection process and established a consistent experimental setup, including both hardware and software configurations. This standardization in data preparation and computational environment ensured reliable and reproducible comparisons.

The findings showed that, although ROI integration aimed to concentrate learning on diagnostically relevant areas, models trained on original full images generally outperformed their ROI-based counterparts—particularly when combined with the **Partially Unfrozen** strategy. Among all configurations, **DenseNet201** with the Partially Unfrozen strategy and **without ROI** consistently achieved the highest accuracy and fastest convergence.
These results indicate that, under the current setup, Grad-CAM-based unsupervised ROI detection may not sufficiently capture the critical diagnostic features needed for optimal classification. Therefore,using full-image inputs combined with partial fine-tuning outperformed both Grad-CAM-based ROI preprocessing and the frozen feature extractor configuration across both the chest X-ray dataset and the brain MRI dataset.

5 Conclusion and future perspectives

Contents

5.1 General conclusion

This thesis aimed to evaluate the effectiveness of region of interest detection in improving medical image classification, with a particular focus on chest X-rays and brain MRI modalities. The proposed two-phase deep learning approach was developed to systematically assess how ROI-guided inputs, detected using Grad-CAM, affected classification accuracy when compared to original full images. This evaluation was conducted using two widely adopted pretrained convolutional neural networks—EfficientNetB4 and DenseNet201—and two fine-tuning strategies: Frozen Feature Extractor and Partially Unfrozen.

In contrast to supervised ROI methods that depend on expert annotations, this study intentionally adopted Grad-CAM as an unsupervised ROI detection technique—without relying on ROI-level supervision—to investigate whether automatically detected regions could serve as effective substitutes for annotated ground truth during preprocessing. This aligns with one of the core goals of this thesis: to assess the real impact of such weakly guided ROI-based inputs on model performance.

The first phase involved generating ROI-enhanced inputs through Grad-CAM to emphasize diagnostically relevant areas. In the second phase, the performance of models trained on these ROI-based images was compared against models trained on unaltered full images. Evaluation across multiple configurations revealed that while ROI-focused inputs aimed to guide learning toward critical regions, models trained on full images consistently performed better. In particular, **DenseNet201** with **partial unfreezing (25%)** and **no ROI** yielded the highest accuracy across both datasets—98.00% on chest X-rays and 99.00% on brain MRIs—along with efficient training times and low loss values.

These results suggest that while Grad-CAM-based detection offers a lightweight and unsupervised means of highlighting informative regions, it may not yield sufficiently complete or reliable ROIs for use as primary training inputs. Additionally, the contextual information

present in full images appears essential for accurate classification. This study underscores the importance of backbone architecture, input design, and training strategy in achieving robust performance across medical imaging datasets.

5.2 Limitations

While this study provided meaningful insights into the role of region of interest integration in medical image classification, several limitations are acknowledged:

- **Supervised ROI Methods Were Not Within the Scope of This Study:** This work focused exclusively on Grad-CAM as an unsupervised ROI detection technique, without incorporating supervised alternatives such as segmentation or object detection. While such methods may offer more accurate ROI delineation when annotated data is available, comparing their performance was beyond the scope of this study and remains a valuable direction for future work.
- **Loss of Contextual Information in ROI-Based Inputs:** Using only Grad-CAM-highlighted regions as input may have led to the exclusion of broader contextual features necessary for accurate diagnosis. This was particularly relevant in cases involving diffuse or overlapping abnormalities, such as certain pulmonary diseases.
- **Limited Exploration of Alternative Unsupervised Methods:** Although Grad-CAM served as the primary unsupervised approach, other methods—such as clustering-based segmentation, contrastive learning, or self-supervised representation learning—were not explored. These may offer different advantages in ROI quality and generalization.
- **Lack of Clinical Expert Validation:** The ROI regions detected and the model predictions produced in this study were not reviewed by radiologists or other clinical experts. As a result, the clinical trustworthiness and diagnostic value of the highlighted areas remain to be verified.
- **Minimal Data Augmentation and Optimization:** To isolate the impact of ROI integration and fine-tuning strategies, no advanced data augmentation or hyperparameter optimization techniques were applied. While this ensured consistent comparisons, it may have limited the overall performance of the models.
- **Restricted Scope to Two Modalities and Datasets:** The evaluation was limited to chest X-ray and brain MRI datasets. Generalization to other modalities, anatomical regions, or rare conditions remains to be validated through broader testing in diverse clinical settings.

5.3 Future Perspectives

Based on the results and limitations of this study, several promising directions can be explored in future work to enhance the use of ROI-guided learning in medical image classification:

- **Comparative Evaluation with Supervised ROI Methods:** To better understand the trade-offs between supervised and unsupervised ROI detection, future studies could

include comparisons with supervised approaches such as U-Net, Mask R-CNN, or object detection models that rely on annotated ground truth. This would help quantify the value of supervision when available.

- **Fusion of ROI-Based and Full-Image Representations:** Instead of using ROI-based or full-image inputs separately, future models could integrate both in a hybrid architecture. For instance, dual-branch networks could process full images alongside Grad-CAM heatmaps or cropped ROIs, allowing the model to learn both global context and local focus.
- **Exploring Alternative Unsupervised ROI Techniques:** Other unsupervised or self-supervised ROI detection methods—such as clustering-based segmentation, attention mechanisms, or contrastive learning—could be explored as potential alternatives to Grad-CAM, possibly providing more complete or consistent region identification.
- **Adoption of Advanced Backbone Architectures:** Incorporating architectures like VITs, hybrid CNN-Transformer models, or self-supervised autoencoders could improve the model's ability to capture both fine-grained features and global patterns, especially when combined with ROI-aware preprocessing.
- **Integration of Data Augmentation and Hyperparameter Optimization:** Applying stronger data augmentation, regularization, or automated hyperparameter tuning may further improve classification performance, especially in limited-data settings common in medical imaging.
- **Clinical Validation of ROI Relevance:** Collaborating with radiologists to validate ROI regions and model predictions would improve clinical interpretability and trust in AI-based decision systems. Visual and diagnostic consistency could be assessed using expert feedback.
- **Extension to Broader Modalities and Conditions:** The framework developed here could be extended to other medical imaging modalities such as CT, PET, or ultrasound, as well as to multi-institutional datasets or rare pathologies, to assess generalization across diverse clinical settings.

Bibliography

[1] Seyed Mahdi Miraftabzadeh, Michela Longo, Federica Foiadelli, Marco Pasetti, and Raul Igual. Advances in the application of machine learning techniques for power system analytics: A survey. *Energies*, 14(16):4776, 2021. Extension of a conference paper presented at IEEE EEEIC 2020, Genova, Italy, 11–14 June 2019.

[2] Vivek Bhagat. Overview of convolutional neural networks. https://www.topcoder.com/thrive/articles/overview-of-convolutional-neural-networks?utm_source=thrive&utm_campaign=thrive-feed&utm_medium=rss-feed, May 2022. Accessed: 2025-06-21.

[3] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016.

[4] Kunal Dawn. Gradcam – enhancing neural network interpretability in the realm of explainable ai, December 2023. Accessed: 2025-06-18.

[5] Lili Guo, Changsheng Zhou, Jingxu Xu, Chencui Huang, Yizhou Yu, and Guangming Lu. Deep learning for chest x-ray diagnosis: Competition between radiologists with or without artificial intelligence assistance. *Journal of Imaging Informatics in Medicine*, 37:922–934, 2024.

[6] Sheetal Rajpal, Navin Lakhyani, Ayush Kumar Singh, Rishav Kohli, and Naveen Kumar. Using handpicked features in conjunction with resnet-50 for improved detection of covid-19 from chest x-ray images. *Chaos, Solitons & Fractals*, 2021.

[7] Hongyu Wang, Hong Gu, Pan Qin, and Jia Wang. CheXLocNet: Automatic localization of pneumothorax in chest radiographs using deep convolutional neural networks. *PLOS ONE*, 15(11):e0242013, 2020. eCollection 2020.

[8] Zhongchen Zhao, Huai Chen, Yu ping Wang, Deyu Meng, Qi Xie, Qi Yu, and Lisheng Wang. Retinal disease diagnosis with unsupervised grad-cam guided contrastive learning. *Neurocomputing*, 593:127816, 2024.

[9] Y. Kumaresan, D. Ren, R. Ni, Y. Huang, N. Lam, H. Sun, S. Wan, M. Wong, K. Chan, H. Tsang, L. Xu, T. Wu, F. Kong, Y. Wáng, J. Qin, L. Chan, M. Ying, and J. Cai. Deep

learning attention-guided radiomics for covid-19 chest radiograph classification. *Quantitative Imaging in Medicine and Surgery*, 13(2):572–584, Feb 2023.

[10] Z. Li, M. Xu, X. Yang, Y. Han, and J. Wang. A multi-label detection deep learning model with attention-guided image enhancement for retinal images. *Micromachines*, 14(3):705, 2023.

[11] Gurbandurdy Dovletov, Duc Duy Pham, Stefan Lörcks, Josef Pauli, Marcel Gratz, and Harald H. Quick. Grad-CAM Guided U-Net for MRI-based Pseudo-CT Synthesis. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 2071–2075, Glasgow, Scotland, United Kingdom, July 11–15 2022. IEEE. PMID: 36086041.

[12] Wikimedia Commons Contributors. VGG16.png - Wikimedia Commons, 2021. Accessed: 2025-07-01.

[13] Baixi Xing, Hanfei Cao, Lei Shi, Huahao Si, and Lina Zhao. AI-driven user aesthetics preference prediction for UI layouts via deep convolutional neural networks. *Cognitive Computation and Systems*, 4(1), March 2022. License: CC BY-NC-ND 4.0.

[14] Gouri Shankar Chakraborty, Salil Batra, Aman Singh, and Makul Mahajan. A novel deep learning-based classification framework for covid-19 assisted with weighted average ensemble modeling. *Diagnostics*, 13(10):1806, May 2023. License: CC BY 4.0.

[15] Satyajit Panigrahy and Subrata Karmakar. Enhancing condition monitoring of outdoor insulator through efficientnet classifier. In *2023 7th International Conference on Computer Applications in Electrical Engineering - Recent Advances (CERA)*, October 2023.

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

[17] Juan R. Terven, Diana M. Cordova-Esparza, Alfonso Ramirez-Pedraza, Edgar A. Chavez-Urbiola, and Julio A. Romero-Gonzalez. A comprehensive survey of loss functions and metrics in deep learning. *Artificial Intelligence Review*, 2025. Accepted: 13 March 2025 / Published online: 11 April 2025.

[18] Shuai Xu, Dongliang Chang, Jiyang Xie, and Zhanyu Ma. Grad-cam guided channel-spatial attention module for fine-grained visual classification. In *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Gold Coast, Australia, October 25–28 2021. IEEE. Published: 15 November 2021.

[19] Mounime El Kabbouri, Casa Bp Settat, Hassan Wassima Lakhchini, and Rachid Wahabi. Artificial intelligence machine learning in finance: A literature review. *International Journal of Accounting, Finance, Auditing, Management and Economics*, 3(6):1082–1095, 2022. ISSN 2658-8455.

[20] Diah Priharsari, Babak Abedin, and Emmanuel Mastio. Orchestrating firm sponsored communities of interest: A critical realist case study. In *Proceedings of the Pacific Asia Conference on Information Systems (PACIS)*, number 32, 2019.

[21] Michael O. Cord, Sarah Jane Delany, and Padraig Cunningham. Supervised learning. In Monique Thonnat and Jean-Paul Germain, editors, *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval*, Cognitive Technologies, pages 21–49. Springer, n.d.

[22] Samreen Naeem, Aqib Ali, Sania Anam, and Munawar Ahmed. An unsupervised machine learning algorithms: Comprehensive review. *International Journal of Computing and Digital Systems*, 13(1):911–921, April 2023.

[23] Friedhelm Schwenker and Mohamed F. Abdel Hady. Semi-supervised learning. In Monica Bianchini, Marco Maggini, and Lakhmi C. Jain, editors, *Handbook on Neural Information Processing*, volume 49 of *Intelligent Systems Reference Library*, pages 215–239. Springer, 2013.

[24] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[25] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, 2016.

[26] Afia Zafar, Muhammad Aamir, Nazri Mohd Nawi, Ali Arshad, Saman Riaz, Abdulrahman Alruban, Ashit Kumar Dutta, and Sultan Almotairi. A comparison of pooling methods for convolutional neural networks. *Applied Sciences*, 12(17):8643, 2022. Received: 12 July 2022; Accepted: 20 August 2022; Published: 29 August 2022.

[27] Enoch Arulprakash and Martin Aruldoss. A study on generic object detection with emphasis on future research directions. *Journal of King Saud University - Computer and Information Sciences*, 34(9):7347–7365, 2022.

[28] IBM. What is computer vision? https://www.ibm.com/think/topics/computer-vision, July 2021. Accessed: 2025-06-21.

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

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2015.

[31] Junnan Li, Dongxu Li, Caiming Xiong, and Steven CH Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *International Conference on Machine Learning (ICML)*, 2022.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[33] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2023.

[34] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, 2016.

[35] Huikai Wu, Junge Zhang, Kaiqi Huang, Kongming Liang, and Yizhou Yu. Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation. *arXiv preprint arXiv:1903.11816*, 2019.

[36] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Rethinking atrous convolution for semantic image segmentation. In *arXiv preprint arXiv:1706.05587*, 2017.

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[38] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Laura Rolland, Laurens van der Gustafson, Tete Xiao, Samuel Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[39] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[40] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(1):172–186, 2021.

[41] TensorFlow Team. Movenet: Ultra fast and accurate pose detection model. https://www.tensorflow.org/lite/models/pose_estimation/overview, 2021. Accessed: 2025-06-11.

[42] TensorFlow.js Team. Posenet: Real-time human pose estimation in the browser with tensorflow.js. https://github.com/tensorflow/tfjs-models/tree/master/pose-detection, 2018. Accessed: 2025-06-11.

[43] IBM. What is transfer learning? https://www.ibm.com/think/topics/transfer-learning, February 2024. Accessed: 2025-05-10.

[44] David M. W. Powers. Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

[45] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[46] TensorFlow. Create production-grade machine learning models with tensorflow [online]. https://www.tensorflow.org, 2023. Accessed: 2025-05-10.

[47] Keras. Keras [online]. https://keras.io, 2023. Accessed: 2025-05-10.

[48] Sartaj Bhuvaji, Ankita Kadam, Prajakta Bhumkar, Sameer Dedge, and Swati Kanchan. Brain tumor classification (mri), 2020.

[49] Rifatul Islam Majumder. Pneumonia &; tuberculosis with normal &; non-x-ray, 2024.

[50] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1251–1258. IEEE, 2017.

[51] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8697–8710, Salt Lake City, UT, USA, June 2018. IEEE.

[52] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pages 6105–6114, Long Beach, California, 2019. Proceedings of Machine Learning Research, PMLR.