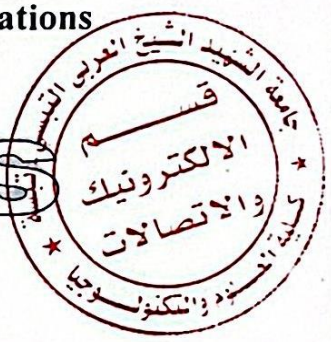
الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
جامعة الشهيد الشيخ العربي التبسي - تبسة
Echahid Cheikh Larbi Tebessi University-Tebessa
Faculty of Science and Technology
Department of Electronics and Telecommunications



Master's Thesis

Submitted for the degree of Academic Master

Field: Telecommunications

Specialty: Networks and Telecommunications

Presented by: (Hiba FILALI & Wissal Ghalloussi)

Title

Enhanced Medical Image Diagnosis Using a Deep Learning Approach Based on Autoencoders

Defended and evaluated on 09/06 /2026, by the jury composed of:

	First and last name	Rank	Position
Mr.	Abdallah MERAOUMIA	Prof	Chair
Mr.	Lotfi HOUAM	MCB	Supervisor
Mrs.	Malika OUACIFI	MCB	Reviewer

Dedications

بسم الله الرحمان الرحيم

First and foremost, all praise and gratitude belong to Allah, the Almighty, for granting me strength, patience, guidance, and countless blessings throughout my academic journey. Alhamdulillah for every challenge that strengthened me and every success that brought me closer to achieving this milestone.

*I dedicate this work to **my beloved father**, whose unwavering support, sacrifices, wisdom, and encouragement have been a constant source of motivation. His belief in my abilities has always inspired me to strive for excellence.*

*I also dedicate this work to **my dear mother**, whose endless love, prayers, care, and support have accompanied me every step of the way. Her dedication and kindness have been the foundation of my success and perseverance.*

*I dedicate this achievement to **myself**, in recognition of the hard work, determination, and resilience invested throughout this journey. Despite the challenges encountered, I remained committed to my goals and successfully reached this important milestone.*

*To my dear sister, **Minnat-Allah**, thank you for your love, encouragement, and constant presence. Your support has meant a great deal to me, and I am grateful to have you by my side.*

*To my dear sister, **Maissem**, thank you for your kindness, motivation, and the joyful moments we shared. Your support has always been appreciated and cherished.*

*I would also like to dedicate this work to my dear project partner, **Wissal**, for her collaboration, dedication, and teamwork throughout this project. It has been a pleasure sharing this academic experience with her.*

*To **my friends**, thank you for your friendship, encouragement, and the memorable moments that made this journey more enjoyable. Your support and positivity have been invaluable.*

*Finally, I dedicate this work to **my colleagues**, whose cooperation, support, and shared experiences contributed to making this academic journey both enriching and rewarding.*

To all of you, thank you for being part of this achievement.

This work is dedicated to you with love, gratitude, and appreciation.

Hiba FILALI

بسم الله الرحمان الرحيم

All praise and thanks are due to Allah, whose endless mercy, guidance, strength, and blessings have enabled me to complete this work. Without His support and wisdom, this achievement would not have been possible. Alhamdulillah for every opportunity, challenge, and success throughout this journey.

*I dedicate this work to **my beloved father**, whose unwavering support, sacrifices, encouragement, and belief in me have been a constant source of motivation. Your wisdom, hard work, and dedication have inspired me throughout my life. Thank you for always standing by my side and guiding me toward success.*

*I dedicate this work to **my dear mother**, whose endless love, care, patience, and prayers have been my greatest strength. Your sacrifices and unconditional support have helped me overcome every obstacle. I am forever grateful for everything you have done for me.*

*I dedicate this work to **myself** for the perseverance, determination, and hard work invested throughout this academic journey. Despite the challenges, I remained committed to my goals and continued moving forward with confidence and dedication.*

***To My Sister Amani,** Thank you for your love, encouragement, and support. Your kindness and belief in me have always motivated me to strive for excellence. I am grateful to have you by my side.*

***To My Sister Djihen, Her Husband, and Their Son,** I dedicate this work to you with love and appreciation. Thank you for your constant encouragement, support, and the happiness you bring into my life. Your presence has always been a source of comfort and motivation.*

***To My Brother Nour,** Thank you for your support, encouragement, and companionship. Your confidence in my abilities has inspired me to keep pursuing my dreams and ambitions.*

*I would like to express my sincere appreciation to my thesis partner, **Hiba**, for her cooperation, dedication, and teamwork throughout this project. It has been a pleasure*

sharing this academic journey with you, and I am grateful for all the efforts we have made together to achieve this accomplishment.

To My Friends,*Thank you for the wonderful memories, encouragement, support, and motivation you have provided throughout my studies. Your friendship has made this journey more enjoyable and meaningful.*

To My Colleagues,*I extend my gratitude to all my colleagues for the shared experiences, cooperation, and support throughout our academic journey. I wish you all continued success and achievement in your future endeavors.*

To everyone who has contributed, directly or indirectly, to my education and personal growth, I dedicate this work with heartfelt gratitude and appreciation.

Wissal GHALLOUSSI

Aknowlegements

First and foremost, all praise and gratitude are due to Allah for granting us the strength, patience, knowledge, and perseverance required to successfully complete this work. Alhamdulillah for His countless blessings and guidance throughout this academic journey.

*We would like to express our deepest gratitude to our supervisor, **Mr. Lotfi HOUAM**, for his invaluable guidance, continuous support, constructive feedback, and encouragement throughout the development of this project. His expertise, insightful advice, and dedication have greatly contributed to the successful completion of this work. We are sincerely grateful for the time and effort he devoted to supervising and supporting us throughout this research.*

We would also like to extend our sincere thanks to the members of the jury for accepting to evaluate our work and for the time and effort they dedicate to examining this dissertation. Their valuable comments and recommendations are highly appreciated.

Our heartfelt appreciation goes to all the professors and staff members of the department who have contributed to our academic and professional development throughout our years of study.

We would like to express our profound gratitude to our families for their unconditional love, patience, sacrifices, encouragement, and unwavering support. Their trust and belief in us have been a constant source of motivation and strength.

Special thanks are extended to our friends and colleagues for their encouragement, cooperation, and the memorable moments we shared during our academic journey.

We would also like to thank everyone who contributed directly or indirectly to the realization of this project and helped us overcome the challenges encountered along the way.

Finally, we are grateful to all those who have supported us throughout our educational journey and contributed to making this achievement possible.

Thank you all for your support, encouragement, and belief in us.

Table of content

Dedications ... **ii**

Aknowlegements ... **v**

Table of content ... **vi**

List of tables ... **xi**

List of figures ... **xii**

List of abbreviations ... **xiv**

Chapter 1 : Introduction ... **1**

1.1 Context and Background ... 1

1.2 Motivation and Problem Statement ... 2

1.3 Objectives of the Thesis ... 3

1.4 Methodology Overview ... 3

1.5 Thesis Organization ... 4

Chapter 2: Foundations of Medical Imaging and CAD Systems ... **6**

2.1 Introduction ... 6

2.2 Physical and Computational Foundations of Medical Imaging ... 7

2.2.1 Introduction to Clinical Visualization and Digital Representation ... 7

2.2.2 Digital Image Representation and Spatial Samplig Theory ... 8

2.2.3 Physical Principles and Mathematical Formulations of Imaging Modalities ... 9

2.2.3.1 Project Radiology (X-ray) ... 9

2.2.3.2 Computed Tomography (CT) ... 11

2.2.3.3 Magnetic Resonance Imaging (MRI) and Microscopic Relaxometriy ... 13

2.2.3.4 Ultrasound Sonography ... 15

2.2.3.4 Positron Emission Tomography (PET) ... 17

2.3 Artificial Intelligence in Medical Imaging 18
2.3.1 Foundations of Deep Learning in Diagnosis Systems 18
2.3.2 Clinical Diagnosis and Computer-Aided Support Systems 19
2.3.3 Classification and Taxonomy of Clinical Support Systems: CADe, CADx, and CADt 20
2.3.3.1 Computer-Aided Detection (CADe) 21
2.3.3.2 Computer-Aided Diagnosis (CADx) 21
2.3.3.3 Computer-Aided Triage and Notification (CADt) 22
2.3.4 Deep Learning Architecture for Semantic Segmentation 23
2.3.5 Loss Functions and Class Imbalance Metigation 24
2.3.6 Quantitative Evaluation Metrics in Medical Imaging Segmentation 25
2.3.7 Discipline Specific Best Practices: Validation Protocols, Patient-Level Splitting, and Data Privacy 26
2.3.7.1 Patient Data Privacy and DICOM De-Identification Standards 28
2.3.7.2 Model Explainability and Interpretability (Explainable AI) 29
2.4 Conclusion 29
Chapter 3 : Theorical Foundations of Deep Learning 31
3.1 Introduction 31
3.2 Artificial Intelligence (AI) 32
3.2.1 Machine Learning 33
3.2.2 Deep Learning 34
3.2.3 Difference between Machine Learning and Deep Learning 34
3.3 Neural Networks and Deep Learning 35
3.3.1 Artificial Neural Network (ANN) 35
3.3.2 Recurrent Neural Network (RNN) 36

3.4 Covolutional Neural Network 36
3.4.1 Overall Architecture 37
3.4.1.1 Input Layer 39
3.4.1.2 Convolutional Layer 39
3.4.1.3 Activation Function 41
3.4.1.4 Pooling Layer 44
3.4.1.5 Fully-Connected Layer (Dense Layer) 44
3.4.1.6 Softmax Layer 46
3.4.2 Top CNN Architecture for Medical Imaging 46
3.4.2.1 Residual Network (ResNet) 46
3.4.2.2 VGG-16 (Visual Geometry Group) 48
3.4.2.3 VGG-19 49
3.4.2.4 AlexNet 50
3.5 Autoencoders 51
3.6 Different Autoencoder Types 52
3.6.1 Under-complete Autoencoder 52
3.6.2 Sparce Autoencoder (SAE) 53
3.6.3 Covolutional Autoencoder (CAE) 54
3.6.4 Variational Autoencoder (VAE) 56
3.6.4.1 Definition 56
3.6.4.2 Mathematical Foundations 56
3.6.4.3 VAE Architecture 57
3.7 Conclusion 59
Chapter 4: Experimental Results and Discussions 61
4.1 Performance Analysis 61

4.1.1 Performance Evaluation Metrics 64

4.1.2 Study 1: Brain Tumor Classification in MRI Images using CNNs, AEs, and Transfer Learning 66

4.1.2.1 Classification Errors, Class Confusions, and Pathological Overlap 67

4.1.2.2 Discriminative Superiority of VGG Models and Transfer Learning 68

4.1.2.3 EfficientNetB0 Failure Mode 68

4.1.2.4 Autoencoder Feature Space (VAE vs. CAE) 69

4.1.3 Study 2: Pneumonia Detection in Chest X-ray Images using CNNs, AEs, and Transfer Learning 70

4.1.3.1 Clinical Diagnosis Metrics 71

4.1.3.2 Pathological and Radiographic Challengies 72

4.1.3.3 Transfer Learning and Feature Extraction 73

4.1.3.4 Autoencoder Performance in Binary Envirement 73

4.2 Learning Curves Analysis 74

4.2.1 Training Dynamics and Convergence Profiles 74

4.2.2 Effectiveness of Data Augmentation and Fine-tunning 78

4.3 Comparative Analysis of CNNs, AEs, and Transfer Learning Approaches 78

4.3.1 Quantitative Performance and Generalization 79

4.3.2 Robustness and Data Requirements 79

4.4 Clinical Relevance and Practical Implementations 79

4.4.1 Computer-Aided Detection (CADe) vs. Computer-Aided Diagnosis (CADx) 79

4.4.2 Multi-Parametric Integration and Real-World Neuro-Oncology 80

4.5 Comparison with Existing Literature 80

4.5.1 Multiclass Neuro-Oncology Classifications 80

4.5.2 Unsupervised Benchmarks in Radiodignostics 81

4.6 Study Limitations 81
4.6.1 Methodological and Dataset-Related Limitations 81
4.6.2 Algorithmic and Statistical Limitations 81
4.7 Conclusion 82
Conclusion 83
References 87
Appendix A 93
Appendix B 93
Appendix A : Convolutional Neural Networks 95
Appendix B : Variational and Convolutional Autoencoders 111
Appendix C : Pipelines of This Study 123
Appendix D : ROC Curves 125

List of tables

Table 2-1: Impact of Slice-Level and Patient-Level Paradigms on Model Evaluation 27

Table 3-1 : Difference between Machine Learning and Deep Learning 35

Table 3-2 : Comparative table between non-linear activation functions [52, 53]..... 43

Table 4-1 : Comparative table that shows the performance of CNN and AE models in brain tumor classification. 62

Table 4-2 : Comparative table that shows the performance of CNN and AE models in pneumonia detection. 63

Table 4-3 : Descriptive table of Evaluation metrics 64

Table 4-4 : Distribution of images used in this study 67

Table 4-5 : Distribution of images used in this study 71

Table 4-6 : Comparative Assessment of Deep Learning Paradigms in Medical Image Classification..... 78

List of figures

Figure 2-1: Projection Radiograph of a Human Hand. 10

Figure 2-2: Non-Contrast Computed Tomography (CT) of the Abdomen. 13

Figure 2-3: Axial Brain Magnetic Resonance Imaging (MRI). 15

Figure 2-4: Magnetic Resonance Microscopy (MRM). 15

Figure 2-5: B-Mode Obstetric Ultrasound Scan. 17

Figure 2-6: Hierarchical Architecture of Machine Learning and Deep Learning 19

Figure 2-7:Saliency Mapping via Gradient-Weighted Class Activation Mapping(Grad-CAM) 29

Figure 3- 1: Relationship between AI, Machine Learning and Deep Learning 32

Figure 3- 2: Artificial Neural Network 35

Figure 3- 3: Recurrent Neural Network architecture 36

Figure 3- 4: Architecture of convolutional network 38

Figure 3- 5: Schematic diagram of basic Convolutional Neural Network architecture 39

Figure 3- 6: Convolution Operation on an X-ray image 39

Figure 3- 7: The convolution operation. 40

Figure 3-8: ZeroPadding and Convolution in CNNs. 41

Figure 3-9: The impact of ReLU function in brain MRI image 42

Figure 3-10: Graphs of Sigmoid function, Tanh function, ReLu function, and softplus function. 43

Figure 3-11: Average , Max , and Min Pooling 44

Figure 3-12: The Dense Layer in CNNs. 45

Figure 3-13: The Flattening Process in CNNs. 45

Figure 3-14: ResNet Architecture. 47

Figure 3-15: The VGG-16 Architecture 49

Figure 3-16: The VGG-19 Architecture 50

Figure 3-17: AlexNet Architecture 51

Figure 3-18: The Representation of an Auto-Encoder. 51

Figure 3-19: An example of Auto-Encoding process. 52

Figure 3-20: Architecture of Under-Complete Auto-Encoder. 53

Figure 3-21: The Sparse Auto-Encoder architecture 54

Figure 3-22: Convolutional Auto-Encoder Architecture 54

Figure 3-23: Variational Autoencoder architecture 57

Figure 4-1 : Elements of Confusion matrix 65

Figure 4-2 : Class distribution of train, test and validation sets in this study 71

Figure 4-3 : Loss and Accuracy curves for VGG-16 models in brain tumor classification 74

Figure 4-4 : Loss and Accuracy curves for VGG-19 in brain tumor classification 75

Figure 4-5 : Loss and Accuracy curves for EfficientNet-B0 in brain tumor classification 75

Figure 4-6 : Loss and Accuracy curves for InceptionV3 in pneumonia detection 76

Figure 4-7 : Loss and Accuracy curves for ResNet50 in pneumonia detection 76

Figure 4-8 : Loss and Accuracy curves for DenseNet121 in pneumonia detection 77

Figure 4-9 : Loss and Accuracy curves for VAE in Pneumonia detection 77

Figure 4-10 : Loss and Accuracy curves for CAE in pneumonia detection 77

List of abbreviations

AI	Artificial Intelligence
MRI	Magnetic Resonance Imaging
PET	Positron Emission Theory
CNN	Convolutional Neural Network
ROC-AUC	Receiver Operating Characteristics-Area Under the Curve
CT	Computed Tomography
RGB	Red , Green , Blue
MRM	Magnetic Resonance Microscopy
RF	RadioFrequency
CAD	Computer-Aided Diagnosis
DSS	Decision Support System
ADR	Adenoma Detection Rate
ML	Machine Learning
DL	Deep Learning
LSTM	Long Short-Term Memory
ANN	Artificial Neural Network
RNN	Recurrent Neural Network
KNN	K-Nearest Neighbors
SVM	Support Vector Machine
ReLU	Rectified Linear Unit
FC	Fully Connected
ResNet	Residual Network
VGG	Visual Geometry Group
AE	Auto-Encoder
SAE	Sparse Auto-Encoder
CAE	Convolutional Auto-Encoder
VAE	Variational Auto-Encoder
KL	Kullback-Leibler

Chapter 1 : Introduction

1.1 Context and Background

Medical imaging has become one of the most important technologies in modern healthcare, providing clinicians with valuable information for disease diagnosis, treatment planning, and patient monitoring. Various imaging modalities, including X-ray, Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and Ultrasound, allow physicians to visualize internal body structures and identify pathological abnormalities. The increasing availability of medical imaging data has significantly improved healthcare services and contributed to more accurate and timely diagnoses.

Despite these advancements, the interpretation of medical images remains a challenging task. The analysis process often requires highly trained specialists and can be affected by factors such as fatigue, workload, and subjective judgment. Furthermore, the growing volume of medical images generated daily places additional pressure on healthcare systems, creating a demand for automated solutions capable of assisting medical professionals in clinical decision-making.

In recent years, Artificial Intelligence (AI) has emerged as a transformative technology in the healthcare sector. AI techniques have demonstrated remarkable capabilities in analyzing large amounts of data, identifying complex patterns, and supporting medical diagnosis. Among AI approaches, Machine Learning (ML) and Deep Learning (DL) have achieved particularly impressive results in medical image analysis. Deep learning models, especially Convolutional Neural Networks (CNNs), have become the dominant approach for image classification tasks due to their ability to automatically extract relevant features from raw image data.

Alongside CNNs, Autoencoders have gained increasing attention for their ability to learn compact and meaningful representations of images. These architectures can be used for feature extraction, dimensionality reduction, denoising, and anomaly detection. Their integration with deep learning classification frameworks has opened new possibilities for enhancing the performance of medical image diagnosis systems.

As a result, intelligent Computer-Aided Detection (CADe) and Computer-Aided Diagnosis (CADx) systems are increasingly being developed to support healthcare professionals in identifying diseases more accurately and efficiently. These systems contribute to reducing diagnostic errors, improving patient outcomes, and optimizing healthcare resources.

1.2 Motivation and Problem Statement

The accurate diagnosis of diseases from medical images is a critical step in clinical practice. Conditions such as pneumonia and brain tumors require early detection and precise classification to ensure appropriate treatment and improve patient prognosis. However, manual interpretation of medical images can be time-consuming and may lead to inconsistencies due to differences in expertise among radiologists and clinicians.

Traditional image processing and machine learning methods often rely on handcrafted feature extraction techniques, which may not adequately capture the complex visual patterns present in medical images. Furthermore, medical datasets are frequently characterized by limited sample sizes, class imbalance, and high variability in image quality, making classification tasks even more challenging.

Deep learning approaches offer a promising solution to these limitations by automatically learning hierarchical feature representations directly from images. Nevertheless, selecting appropriate architectures, optimizing model performance, and ensuring reliable classification remain important research challenges.

This thesis addresses the problem of medical image classification by investigating advanced deep learning techniques based on Convolutional Neural Networks and Autoencoders. The study focuses on two classification problems: binary classification for pneumonia detection using chest X-ray images and multiclass classification for brain tumor recognition. The objective is to evaluate the effectiveness of different deep learning architectures and identify suitable approaches for improving diagnostic performance.

1.3 Objectives of the Thesis

The primary objective of this thesis is to develop and evaluate deep learning models for automated medical image diagnosis. More specifically, the research aims to investigate the performance of Convolutional Neural Networks and Autoencoder-based architectures in classifying medical images and assisting disease diagnosis.

To achieve this goal, several specific objectives have been defined. First, the study seeks to understand the theoretical foundations of Artificial Intelligence, Machine Learning, Deep Learning, and medical image analysis. Second, it aims to explore different CNN architectures, including VGG16, VGG19, ResNet50, DenseNet121, EfficientNetB0, and InceptionV3, for medical image classification tasks. Third, the research investigates the role of Autoencoders, particularly Convolutional Autoencoders (CAE) and Variational Autoencoders (VAE), in feature learning and representation extraction.

Another objective is to apply preprocessing techniques, data augmentation strategies, and transfer learning methods to improve model performance and generalization capabilities. Finally, the study aims to compare the performance of the proposed models using multiple evaluation metrics, including Accuracy, Precision, Recall, F1-score, and AUC, in order to determine their effectiveness for medical diagnosis applications.

1.4 Methodology Overview

To accomplish the objectives of this research, a systematic methodology was adopted. The study began with data collection and preparation using publicly available medical imaging datasets related to pneumonia detection and brain tumor classification.

The preprocessing stage involved image normalization, resizing, and quality enhancement procedures to ensure consistency across the datasets. To overcome limitations associated with insufficient training samples and improve model robustness, data augmentation techniques were applied. These techniques generated additional training examples through transformations such as rotation, flipping, zooming, and shifting.

The datasets were divided into training, validation, and testing subsets using an 80%, 15%, and 5% distribution, respectively. Transfer learning was employed by utilizing pre-trained CNN architectures that had previously learned general visual features from large

image datasets . Fine-tuning strategies were then applied to adapt these models to the specific medical imaging tasks.

For binary classification, several CNN architectures as well as Autoencoder-based models were implemented and evaluated. For multiclass classification, multiple state-of-the-art CNN models were trained and compared. The performance of all models was assessed using standard classification metrics to ensure a comprehensive evaluation of their diagnostic capabilities.

The obtained results were analyzed and interpreted to identify the most effective architectures and techniques for medical image classification.

1.5 Thesis Organization

This dissertation is organized into four chapters.

Chapter 1: Introduction presents the general background of the study and highlights the importance of artificial intelligence in healthcare. It introduces the research motivation, defines the problem statement, and outlines the objectives of the thesis.

Chapter 2: Foundations of Medical Imaging and Computer-Aided Diagnosis Systems provides a comprehensive overview of medical imaging and its role in modern medicine. The chapter introduces the concept of medical images and describes the main imaging modalities, including X-ray, CT, MRI, MRM and Ultrasound. It also discusses the application of artificial intelligence in medical imaging and presents Computer-Aided Detection (CADe) and Computer-Aided Diagnosis (CADx) systems as important tools for diagnostic support.

Chapter 3: Theoretical Background of Machine Learning and Deep Learning presents the fundamental concepts underlying the proposed research. The chapter begins with an introduction to Artificial Intelligence, Machine Learning, and Deep Learning. It then discusses Artificial Neural Networks and traditional machine learning classification algorithms such as K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Decision Trees, and Random Forests. Furthermore, the chapter provides a detailed explanation of Convolutional Neural Networks, their architectures, and popular models including VGG-16 , VGG-19 AlexNet, and ResNet. It also introduces Autoencoders and their variants, including

Sparse Autoencoders (SAE) , Convolutional Autoencoders (CAE) , and Variational Autoencoders (VAE), as well as the main hyperparameters involved in deep learning model optimization.

Chapter 4: Experimental Results and Discussion describes the practical aspects of the research work . It presents the datasets used , preprocessing procedures , data augmentation techniques , transfer learning strategies , and implementation details . The chapter also explains the developed models , experimental setup , evaluation metrics , obtained results , and their interpretation . Finally , a comparative analysis of the different architectures is provided to identify the most effective approaches for medical image diagnosis.

Through these chapters , this thesis aims to demonstrate the potential of deep learning technologies in improving the accuracy and reliability of medical image classification systems and contributing to the development of intelligent diagnostic support tools for healthcare applications.

Chapter 2

Foundations of Medical Imaging and CAD Systems

2.1 Introduction

Medical imaging has become an indispensable component of modern healthcare, enabling physicians to visualize internal anatomical structures and detect pathological abnormalities in a non-invasive manner . Imaging modalities such as X-ray , Computed Tomography (CT) , Magnetic Resonance Imaging (MRI) , and Ultrasound play a crucial role in disease diagnosis, treatment planning, and patient monitoring [1], [2]. The continuous growth in medical imaging data generated by healthcare institutions has significantly increased the demand for automated and intelligent analysis systems capable of assisting clinicians in their diagnostic tasks [3].

In recent years, Artificial Intelligence (AI), particularly Machine Learning (ML) and Deep Learning (DL), has emerged as a transformative technology in medical image analysis. Deep learning models have demonstrated remarkable performance in various diagnostic applications, including disease detection, classification, segmentation, and prognosis prediction [4], [5]. Among these approaches, Convolutional Neural Networks (CNNs) have achieved state-of-the-art results in image classification tasks due to their ability to automatically learn hierarchical feature representations directly from raw images [6].

In addition to CNNs, Autoencoders have attracted increasing research interest because of their ability to learn compact and meaningful latent representations from medical images. These architectures have been successfully applied to feature extraction, dimensionality reduction, denoising, anomaly detection, and image reconstruction tasks [7], [8]. Consequently, integrating CNNs and Autoencoder-based approaches offers promising opportunities for improving the accuracy and robustness of computer-aided diagnosis systems.

The development of intelligent Computer-Aided Detection (CADe) and Computer-Aided Diagnosis (CADx) systems has therefore become an active research area. Such systems aim to support healthcare professionals by reducing diagnostic errors, improving clinical efficiency, and enhancing patient outcomes [9]. This thesis investigates the potential of deep learning architectures based on CNNs and Autoencoders for enhancing medical image diagnosis in brain tumor classification and pneumonia detection tasks.

2.2 Physical and Computational Foundations of Medical Imaging

2.2.1 Introduction to Clinical Visualization and Digital Representation

Medical diagnosis has progressed considerably through the emergence of advanced imaging technologies capable of producing accurate, non-invasive, and highly detailed visual information about internal anatomical structures and physiological functions. Whether based on electronic detectors, digital imaging arrays, or computer-assisted reconstruction, these systems enable physicians to observe, measure, and evaluate organs and tissues with a level of precision that supports early detection, characterization, therapeutic planning, intraoperative guidance, and longitudinal monitoring of disease.

A medical image should not be considered a simple photographic representation of the examined organ. Rather, it is a sophisticated multidimensional visualization of biological tissues generated from physical signals and computational transformations. Projection radiography, Computed Tomography (CT), Magnetic Resonance Imaging (MRI), ultrasound sonography, and nuclear medicine modalities such as SPECT and PET rely on distinct physical principles and therefore provide complementary clinical information. This diversity of mechanisms explains why medical imaging is closely linked to digital image processing, including enhancement, segmentation, feature extraction, classification, and quantitative analysis.

The term image, derived from the Latin imago, refers generally to the visual representation of an object, scene, or physical phenomenon. In medicine, however, the image represents spatial and, in some cases, temporal variations of measurable physical properties such as electron density, nuclear magnetic resonance behavior, acoustic impedance, or radiotracer concentration. When temporal evolution is explicitly modeled, the image is considered dynamic, making it suitable for visualizing physiological processes such as

perfusion, cardiac motion, respiratory dynamics, and tracer kinetics. When time is not considered, the image is static and is primarily used for morphological assessment, lesion detection, and anatomical evaluation. Static and dynamic images therefore play complementary roles in diagnostic practice.

2.2.2 Digital Image Representation and Spatial Samplig Theory

A digital image is defined as a discrete representation of a continuous spatial scene, composed of a regular array of finite elements known as pixels (picture elements) for 2D images, or voxels (volume elements) for 3D volumetric scans[1]. For this reason, a digital image is represented mathematically as a matrix or a multi-dimensional array where each element carries a specific discrete value representing intensity or color components. These pixels are organized in a structured manner to form a coherent visual representation that can be processed and analyzed by computer systems.

Formally, a 2D digital image can be defined as a discrete function ***I(x, y)*** mapped onto a bounded 2D coordinate space, where $x \in [0, M-1]$ and $y \in [0, N-1]$ denote the discrete spatial coordinates of the image matrix. The value of the function at any point ***(x, y)*** corresponds to the localized signal intensity:

$$I(x,y) \in B \tag{2.1}$$

Where ***B*** represents the bit depth of the image, defining the dynamic range of gray levels. In standard medical imaging modalities, ***B*** typically ranges from 12 bits (representing 4,096 discrete intensity values in CT and radiography) to 16 bits, enabling the preservation of subtle contrast variations.

The most common representation of a digital image is based on a regular rectangular grid, where pixels are arranged along two orthogonal axes to form a matrix structure. In this matrix, each element corresponds to a pixel located at a specific spatial position and containing a numerical value that describes its intensity (in grayscale images) or color components (in RGB images). This structured representation facilitates efficient storage, manipulation, and processing of images using standard digital algorithms.

Although the rectangular grid is the most widely used format in practice due to its simplicity and computational efficiency, other sampling geometries also exist in image processing theory. Hexagonal and triangular sampling grids have been proposed in certain research contexts, as they offer unique advantages in terms of sampling uniformity, isotropic neighborhoods, and reduced spatial distortion.

Specifically, a hexagonal grid provides a higher packing density, isotropic connectivity (each pixel has six equidistant neighbors, eliminating the distinction between orthogonal and diagonal neighbors), and superior representation of curved biological boundaries. However, due to the historical design of hardware acquisition sensors and computational architectures, these alternative representations are less common in standard medical systems.

In general, a digital image refers to any image that has been acquired, processed, and stored in a coded numerical form . This means that the image is represented using discrete values, typically in binary format (0s and 1s), which can be interpreted and processed by digital systems . The conversion of real-world continuous physical phenomena into this digital format is achieved through imaging devices, analog-to-digital converters, and discrete spatial-frequency sampling processes , enabling advanced computer-aided clinical operations.

2.2.3 Physical Principles and Mathematical Formulations of Imaging Modalities

2.2.3.1 Project Radiology (X-ray)

Projection radiography is one of the earliest and most widely used diagnostic techniques in medical imaging. It operates by employing ionizing electromagnetic radiation (photons with energies typically ranging from 20 to 150 keV) to produce two-dimensional projection representations of internal anatomical structures. During the imaging process, a collimated X-ray beam is emitted from a vacuum tube, passes through the human body, and is differentially absorbed depending on the density, thickness, and atomic composition of the traversed tissues.

The physical decay of the X-ray beam intensity as it propagates through biological media is mathematically governed by the Beer-Lambert Law. For a monochromatic beam

passing through a homogeneous tissue of thickness ***x***, the transmitted intensity ***I(x)*** is formulated as:

$$I(x) = I_0 e^{-\mu x} \tag{2.2}$$

Where $\mathbf{I_0}$ is the incident X-ray intensity, and $\boldsymbol{\mu}$ is the linear attenuation coefficient of the tissue [2]. The value of $\boldsymbol{\mu}$ is highly dependent on the effective atomic number ***z*** of the tissue and the photon energy ***E***, primarily driven by the photoelectric effect and Compton scattering:

$$\mu \propto p\left(\frac{Z^3}{F^3}\right) \tag{2.3}$$

Where **ρ** represents physical density. In clinical applications, the human body acts as a heterogeneous medium, requiring the integration of varying attenuation coefficients along the X-ray trajectories :

$$I = I_0(-\int_{path} \mu(s)ds\ ) \tag{2.4}$$

As a result of this differential attenuation, denser structures with high effective atomic numbers, such as cortical bone (rich in calcium, Z = 20), absorb a high proportion of the incident radiation, allowing fewer photons to reach the detector. These regions are represented as bright or light gray on the radiograph.

Conversely, soft tissues , muscles , adipose tissues , and air-filled structures (such as the lungs) have low density and low effective atomic numbers, allowing more radiation to pass through. These regions are represented in darker shades of gray.



Figure 2-1: Projection Radiograph of a Human Hand.

Conventional radiography remains a fundamental tool in clinical practice due to its high spatial resolution, rapid acquisition speed, and relatively low cost. It is extensively used for the detection and evaluation of bone fractures, assessment of pulmonary pathologies (such as pneumonia, pleural effusion, or mass lesions), and identification of dental conditions.

Despite its wide utility, X-ray imaging has substantial limitations. One of the main drawbacks is its low contrast resolution for soft tissues, which limits its effectiveness in visualizing subtle differences between organs of similar physical densities.

Furthermore, because it is a 2D projection technique, 3D anatomical structures are superimposed onto a single plane, causing diagnostic ambiguity. Lastly, the use of ionizing radiation introduces potential health risks; cumulative exposure over time may induce double-strand DNA breaks, increasing the long-term risk of carcinogenesis.

2.2.3.2 Computed Tomography (CT)

Computed Tomography (CT) is a cross-sectional imaging modality that uses rotating X-ray tubes and solid-state detectors to generate high-resolution, three-dimensional volumetric representations of the human body [1]. Unlike conventional radiography, which produces a single projection image, CT reconstructs multiple adjacent cross-sectional slices, eliminating the clinical limitations associated with anatomical superimposition.

During a CT scan, the patient is placed on a motorized table that moves through a rotating gantry. An X-ray source rotates rapidly around the patient, emitting a fan-shaped or cone-shaped beam of radiation. Directly opposite the source, a multi-row curved detector array measures the transmitted X-ray intensity along thousands of angular projections [2].

The mathematical basis for CT image reconstruction from projection data is the Radon Transform, introduced by Johann Radon in 1917. If a 2D cross-sectional tissue slice is represented as a spatial distribution of linear attenuation coefficients $\boldsymbol{\mu(x, y)}$, a single projection measurement $\boldsymbol{g(\rho, \theta)}$ at an acquisition angle $\boldsymbol{\theta}$ and detector position $\boldsymbol{\rho}$ is defined as the line integral of $\boldsymbol{\mu(x, y)}$along the ray path:

$$g(\rho,\theta) = R[\mu](\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mu(x,y)\delta(x\cos\theta + y\sin\theta - \rho)dsdy \qquad (2.5)$$

Where **δ** is the Dirac delta function defining the linear integration path. The complete set of these multi-angle projections forms a 2D coordinate space known as a **sonogram** [24].

To reconstruct the original spatial distribution ***μ(x, y)*** from the sinogram, the Fourier Slice Theorem is applied [4]. This theorem states that the 1D Fourier transform of a projection ***g(ρ, θ)*** with respect to **ρ** is equal to a radial slice of the 2D Fourier transform of the original image ***μ(x, y)*** at angle ***θ*** [4]:

$$G(\omega,\theta) = F_{1D}\{g(\rho,\theta)\} = M(\omega\cos\theta + \omega\sin\theta) \tag{2.6}$$

Where ***M(u, v)*** is the 2D Fourier transform of ***μ(x, y)***. In clinical CT systems, the Filtered Back Projection (FBP) algorithm is used to mathematically reconstruct the image[4]. This algorithm filters the projection data in the frequency domain using a ramp (Ram-Lak) or Shepp-Logan filter to eliminate blurring, and then back-projects the filtered signals across the image reconstruction grid:

$$\mu(x,y) = \int_0^{\pi}\left[\int_{-\infty}^{\infty}|\omega|G(\omega,\theta)e^{j2\pi\omega(x\cos\theta+y\sin\theta)}d\omega\right]d\theta \tag{2.7}$$

To ensure standardization across different CT scanner manufacturers and acquisition protocols, the reconstructed linear attenuation coefficients ***μ(x, y)*** are converted into normalized Hounsfield Units (HU), which defines the physical density of tissues relative to water [2]:

$$HU = 1000 \times \frac{\mu_{tissue} - \mu_{water}}{\mu_{water} - \mu_{air}} \tag{2.8}$$

Under standard calibration settings, air is defined as -1000 HU, pure water as 0 HU, dense cortical bone ranges from +1000 HU to +3000 HU, and soft tissues typically fall within the range of +20 HU to +80 HU.

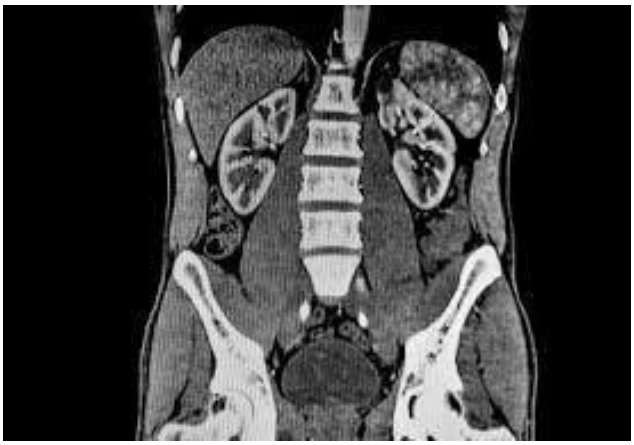


Figure 2-2: Non-Contrast Computed Tomography (CT) of the Abdomen.

CT imaging is widely used for evaluating acute trauma, vascular structures (via CT angiography), pulmonary parenchymal disorders, and oncological staging. Its main advantages are rapid scan times (under 5 seconds for complete organ volumes) and excellent spatial resolution for bone and soft-tissue boundaries.

However, because CT relies on multiple rotation projections, patient exposure to ionizing radiation is significantly higher than in projection radiography, raising clinical concerns regarding cumulative radiation dosage.

2.2.3.3 Magnetic Resonance Imaging (MRI) and Microscopic Relaxometriy

Magnetic Resonance Imaging (MRI) and Magnetic Resonance Microscopy (MRM) are advanced non-invasive modalities based on the principles of nuclear magnetic resonance. Unlike X-ray-based imaging, MRI does not rely on ionizing radiation, making it particularly valuable for repeated clinical examinations and for applications requiring high soft-tissue contrast. Its physical basis relies on the interaction between a strong static magnetic field, typically 1.5 T to 3.0 T in clinical systems, and the magnetic moments of hydrogen nuclei within water molecules [25].

Under normal conditions, proton spins are randomly oriented. When the patient is placed inside the static magnetic field B_0, these spins align either parallel or anti-parallel to

the field, generating a net macroscopic magnetization vector along the longitudinal axis. The aligned spins precess around B0 at the Larmor frequency, expressed as [9]:

$$\omega_0 = \gamma B_0 \tag{2.9}$$

Where $\boldsymbol{\omega_0}$ is the precessional frequency and γ is the gyromagnetic ratio. A radiofrequency (RF) pulse applied at the Larmor frequency perturbs this equilibrium by tipping the magnetization vector into the transverse plane. After the RF pulse is removed, the spin system returns to thermal equilibrium while emitting weak RF signals that are detected by receiver coils. The macroscopic magnetization dynamics are classically described by the Bloch equations, which incorporate the static magnetic field and the longitudinal and transverse relaxation time constants T1 and T2 [8, 9].

T1 relaxation describes the recovery of longitudinal magnetization as excited spins transfer energy back to the molecular lattice, while T2 relaxation describes the decay of transverse magnetization caused by spin dephasing and local magnetic field fluctuations [26]. In simplified form, the longitudinal and transverse relaxation processes may be represented as:

$$M_z(t) = M_0\left(1 - e^{\frac{t}{T_1}}\right) and\ M_{xy}(t) = M_{xy}(0)e^{\frac{t}{T_2}} \tag{2.10}$$

Image contrast is controlled mainly by the Repetition Time (TR) and Echo Time (TE). T1-weighted imaging uses a short TR and short TE, allowing tissues with short T1 values such as adipose tissue to appear bright, while long-T1 tissues such as cerebrospinal fluid appear dark [6]. T2-weighted imaging uses a long TR and long TE, making fluids and edematous tissues with long T2 relaxation times appear hyper-intense [7].

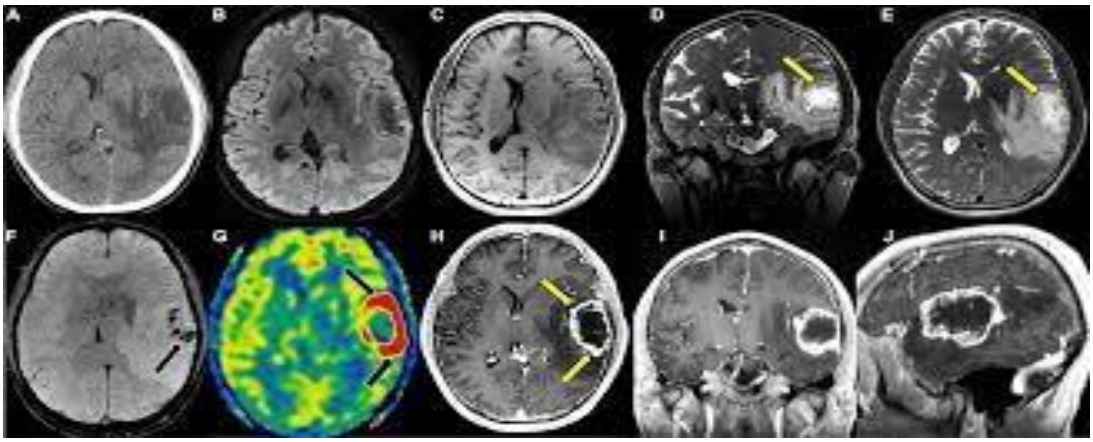


Figure 2-3: Axial Brain Magnetic Resonance Imaging (MRI).

MRI is therefore highly valued for its exceptional soft-tissue contrast, multi-planar acquisition capability, and absence of ionizing radiation. It remains a clinical gold standard in neuro-imaging, musculoskeletal assessment, and cardiovascular evaluation. Magnetic Resonance Microscopy extends these principles to microscopic scales, allowing the visualization of small structural patterns with spatial resolution fewer than 100 micrometers in research and specialized diagnostic contexts.

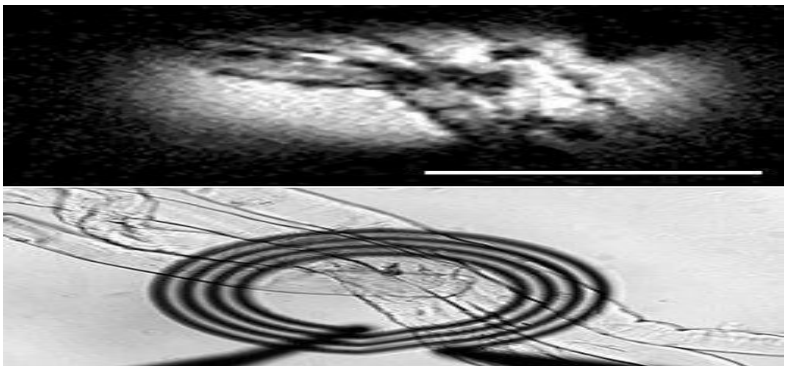


Figure 2-4: Magnetic Resonance Microscopy (MRM).

2.2.3.4 Ultrasound Sonography

Ultrasound sonography is a real-time, non-invasive imaging modality that uses high-frequency mechanical acoustic waves, typically between 2 and 18 MHz, to map internal anatomical structures. Its operation depends on the piezoelectric effect: crystals in the transducer transform electrical energy into acoustic pulses and then convert returning echoes into electrical signals. As these waves propagate through tissues, they encounter boundaries

between media with different acoustic impedances. The acoustic impedance Z of a tissue depends on physical density ρ and the speed of sound c in that medium:

$$Z = \rho \cdot c \tag{2.11}$$

When an acoustic wave reaches an interface between two media of impedance Z1 and Z2, part of the energy is reflected and part continues to propagate. The intensity reflection coefficient R is expressed as:

$$R = \left(\frac{(Z_2 - Z_1)}{(Z_2 + Z_1)}\right)^2 \tag{2.12}$$

Small impedance differences, as commonly observed between soft tissues, produce low-amplitude echoes and darker gray regions. Large impedance mismatches, especially at interfaces containing bone or air, reflect most of the acoustic energy and create bright hyperechoic boundaries followed by acoustic shadowing. For vascular assessment, Doppler ultrasound measures the frequency shift caused by moving red blood cells, allowing estimation of blood flow direction and velocity in real time:

$$\Delta f = \left(\frac{2 f_0 v cos\theta}{c}\right) \tag{2.13}$$

Small impedance differences, as commonly observed between soft tissues, produce low-amplitude echoes and darker gray regions. Large impedance mismatches, especially at interfaces containing bone or air, reflect most of the acoustic energy and create bright hyperechoic boundaries followed by acoustic shadowing. For vascular assessment, Doppler ultrasound measures the frequency shift caused by moving red blood cells, allowing estimation of blood flow direction and velocity in real time:



Figure 2-5: B-Mode Obstetric Ultrasound Scan.

2.2.3.4 Positron Emission Tomography (PET)

Positron Emission Tomography (PET) belongs to the field of nuclear medicine and is primarily a functional imaging technique. Unlike anatomical modalities such as CT and MRI, PET maps the spatial and temporal distribution of intravenously administered radiopharmaceuticals, thereby providing information about metabolic activity and physiological pathways. The most common clinical example is Fluorine-18 labeled fluorodeoxyglucose (18F-FDG), a glucose analogue used to evaluate tissues with increased glucose uptake[6].

PET relies on radionuclides undergoing beta-plus decay, in which a proton is converted into a neutron, a positron, and an electron neutrino:

$$p \rightarrow n + e + + ve \tag{2.14}$$

After emission, the positron travels a short distance in tissue until it interacts with a free electron. The resulting annihilation event converts both particles into two 511 keV gamma photons emitted in approximately opposite directions:

A PET scanner contains a ring of scintillation detectors connected by coincidence

$$e + + e - \rightarrow \gamma + \gamma \quad (2.15)$$

detection circuits. When two 511 keV photons strike opposing detectors within a narrow time window, the system records a true coincidence event along a Line of Response (LOR). By collecting millions of LORs, the scanner reconstructs the spatial distribution of the radiotracer, commonly using iterative algorithms such as Ordered Subset Expectation Maximization (OSEM).

PET is highly sensitive at picomolar tracer concentrations and is essential in oncology for staging and treatment response monitoring, as well as in neurology for studying dementia and other functional disorders [6].

2.3 Artificial Intelligence in Medical Imaging

2.3.1 Foundations of Deep Learning in Diagnosis Systems

Artificial Intelligence (AI) has emerged as a key technology in the healthcare sector, particularly in the fields of medical imaging and computer-aided diagnosis (CAD). In recent years, the rapid growth of digital medical image datasets, combined with advances in machine learning and deep learning algorithms, has significantly enhanced the capabilities of intelligent diagnostic systems. These AI-based techniques are able to process and analyze large volumes of complex medical data with high speed and efficiency, thereby assisting healthcare professionals in improving the accuracy and reliability of clinical decision-making.

The evolution of clinical diagnostic support systems can be categorized into two major technological paradigms [27]:

Classical Handcrafted Feature Workflows: Traditional computer-aided diagnosis systems relied on manual, step-by-step feature engineering [27]. This required image processing experts to manually design mathematical filters to extract specific textures, shapes, and edge gradients (such as Gray-Level Co-occurrence Matrices or Scale-Invariant Feature Transforms)[27]. These features were then passed to classical classifiers, such as Support Vector Machines (SVMs) or Random Forests, to detect or categorize abnormalities.[10] These traditional systems often struggled to handle the natural variability of biological structures and changes in imaging protocols across different clinics [11].

Deep Representation Learning: Modern deep learning systems utilize deep neural networks to automatically learn hierarchical representations directly from raw pixel arrays [27]. As the medical image propagates through successive layers of a deep network, the initial layers extract simple features such as edges and textures [28]. Deeper layers then progressively combine these representations to detect more complex structures, such as organ boundaries, vascular configurations, and specific pathological lesions. This end-to-end learning paradigm eliminates the subjectivity of manual feature extraction, enabling the development of more robust and generalizable diagnostic tools [27].

These advanced deep learning models serve as the computational foundation for modern computer-aided detection (CADe), computer-aided diagnosis (CADx), and computer-aided triage (CADt) systems [10]. By automating routine quantitative tasks and flagging high-risk cases, these intelligent tools help reduce diagnostic errors, improve workflow efficiency, and support more consistent clinical evaluations [1].

2.3.2 Clinical Diagnosis and Computer-Aided Support Systems

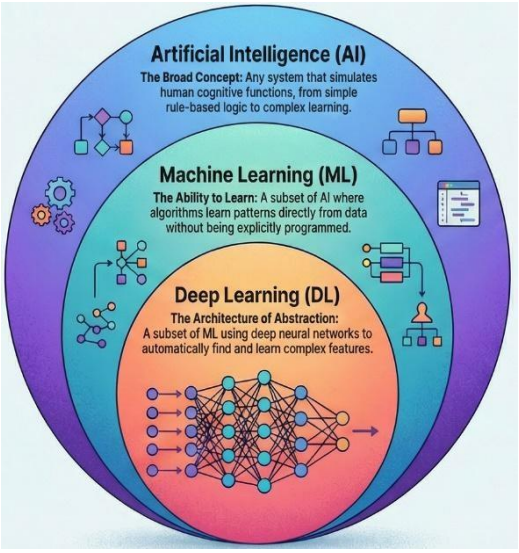


Figure 2-6: Hierarchical Architecture of Machine Learning and Deep Learning.

The term "diagnosis" originates from the Greek word meaning the identification of a condition through observable signs and symptoms. It refers to the intellectual and clinical process through which a healthcare professional identifies a disease by analyzing patient examination results, reported symptoms, clinical signs, and additional medical investigations.[1]

This process is not limited to a simple observation but involves critical reasoning and interpretation of multiple, complex sources of information.

The physician integrates clinical findings with laboratory tests and medical imaging results in order to reach a precise and reliable conclusion, which serves as a fundamental step in medical practice by guiding treatment decisions and determining the appropriate therapeutic strategy for the patient.

Clinical Decision Support Systems (CDSS), also known as computer-aided diagnostic systems, are designed to assist healthcare professionals in the clinical decision-making process. These systems are defined as electronic or non-electronic software tools designed to aid directly in clinical decision-making, in which patient-specific assessments or recommendations are generated and presented to clinicians for consideration.

These systems aim to enhance the medical diagnostic process by analyzing individual patient data, including clinical symptoms, medical history, and examination results. By processing and interpreting this information, diagnostic support systems are able to generate relevant recommendations that help clinicians in identifying possible diseases and selecting appropriate diagnostic or therapeutic strategies.

Rather than replacing the physician's role, these systems serve as intelligent assistants that improve the quality and accuracy of medical decisions, contributing to reducing diagnostic errors, supporting evidence-based medicine, and ensuring more consistent clinical evaluations.

2.3.3 Classification and Taxonomy of Clinical Support Systems: CADe, CADx, and CADt

This section defines the distinct clinical goals, mathematical outputs, and validation pathways of CADe and CADx, and introduces Computer-Aided Triage (CADt) systems to complete the clinical workflow picture [10].

Clinical Diagnostic Support Systems are specialized software tools designed to enhance clinical workflow by providing an automated "second opinion" for healthcare professionals. These systems are classified into distinct categories based on whether their clinical goal is to detect an abnormality, characterize its properties, or prioritize the study on a work list [1].

2.3.3.1 Computer-Aided Detection (CADe)

The primary objective of a Computer-Aided Detection system is the automatic localization and identification of suspicious lesions within medical images. CADe systems analyze images to highlight potential abnormal regions that may indicate the presence of diseases such as lung nodules, breast microcalcifications, bone fractures, or colorectal polyps [1].

The system analyzes medical images using digital image processing and pattern recognition techniques to detect regions of interest (ROIs) and mark them for further evaluation by radiologists. CADe acts as a visual support tool by drawing attention to areas that require closer inspection, which is particularly important in complex scans where subtle abnormalities may be difficult to spot. It also helps reduce the likelihood of missed diagnoses caused by human fatigue, distraction, or the high complexity of surrounding anatomical structures.

A standard CADe system typically operates through four critical stages:

i. **Preprocessing and Segmentation:** Isolating the organ of interest (e.g., the lung parenchyma) from surrounding tissues to constrain the search space [1].
ii. **Candidate Detection:** Searching for abnormal structures that may represent disease, such as pulmonary nodules or breast calcifications.
iii. **Feature Analysis:** Evaluating the intensity, morphology, size, and texture of the identified candidate regions.
iv. **False Positive Elimination:** Using classifiers to distinguish between real pathological lesions and non-pathological anatomical structures to improve accuracy [1].

The performance of CADe systems is evaluated primarily using lesion level sensitivity, the ability to correctly identify as many true lesions as possible balanced against the average number of False Positive detections per case [1].

2.3.3.2 Computer-Aided Diagnosis (CADx)

While CADe systems focus on the identification and localization of suspicious lesions, Computer-Aided Diagnosis (CADx) systems are designed to characterize and evaluate the nature of these detected abnormalities. Once a potential lesion has been identified, CADx

systems analyze its biological properties in greater detail to estimate the probability that the lesion is benign or malignant [1].

CADx provides a comprehensive assessment by extracting and analyzing quantitative features from the medical image, such as margin irregularity, internal density, texture patterns, and vascular perfusion dynamics [1]. Advanced CADx systems may also combine these image-based features with non-imaging biomarkers, such as patient age, smoking status, and genetic profiles, to improve diagnostic accuracy.

Instead of just marking a spot, CADx provides a continuous probability score or a ranked list of potential conditions based on how well the patient's data matches known pathological patterns [1]. Performance is typically assessed using the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC), which balances sensitivity and specificity.

This allows CADx systems to assist clinicians in making more informed diagnostic decisions, helping reduce unnecessary invasive procedures, such as biopsies, by correctly identifying benign findings that mimic cancer.

2.3.3.3 Computer-Aided Triage and Notification (CADt)

In modern emergency medicine, Computer-Aided Triage and Notification (CADt) systems are widely used. Unlike CADe and CADx, which assist the clinician during active image interpretation, CADt runs asynchronously in the background as soon as a scan is acquired [10].

The primary objective of a CADt system is to analyze medical images for time-sensitive, life-threatening findings, such as intracranial hemorrhage, acute large-vessel occlusion, or tension pneumothorax [12].If a critical finding is detected, the CADt system flags the case and automatically reorganizes the radiologist's worklist, prioritizing the urgent scan for immediate review [10].

Importantly, CADt systems do not display diagnostic annotations or place marks on the images; their output is a binary notification designed solely to reduce time-to-treatment in critical situations [10].

2.3.4 Deep Learning Architecture for Semantic Segmentation

Convolutional Neural Networks (CNNs) represent the central architectural foundation of modern computer vision and medical image analysis. Unlike fully connected networks, CNNs preserve spatial organization by applying learnable filters across local regions of the image, thereby extracting meaningful structural patterns while reducing the number of model parameters through weight sharing [28]. For a two-dimensional input image X and kernel K, the convolution operation produces a feature map S at each spatial location:

$$s(i,j) = (X * K)(i,j) = \sum_{m}\sum_{n} X(i-m, j-n)K(m,n) \tag{2.16}$$

This operation enables translation invariance, allowing the network to recognize a pathological or anatomical feature regardless of its position within the scan [28]. Non-linear activation functions such as the Rectified Linear Unit (ReLU) allow the network to model complex relationships, while pooling layers reduce computational cost, enlarge the receptive field, and improve robustness to minor spatial variations [18].

For medical image segmentation, where each pixel or voxel must be labeled as belonging to an anatomical or pathological class, U-Net remains landmark architecture. Introduced by Ronneberger et al., U-Net follows a symmetric encoder-decoder structure [12], [31]. The contracting path progressively reduces spatial resolution while learning high-level semantic context. The expanding path then restores spatial resolution through upsampling or transpose convolutions. Skip connections concatenate high-resolution features from the encoder with corresponding decoder layers, preserving boundary details that are essential for delineating tumors, vessels, and small anatomical structures [14], [32].

Although CNNs are highly effective, their receptive field is generally local and constrained by the kernel size or network depth. Vision Transformers address this limitation by modeling long-range dependencies through self-attention mechanisms [13]. In self-attention, input features are projected into Queries (Q), Keys (K), and Values (V), and the scaled dot-product attention is computed as [33]:

$$Attention(Q,K,V) = Softmax\left(\frac{(QK^T)}{\sqrt{dk}}\right)V \quad (2.17)$$

This mechanism allows image patches to attend dynamically to other regions, making it possible to capture global anatomical relationships. For three-dimensional medical image segmentation, Swin UNETR combines Transformer-based global context modeling with convolutional localization. Its hierarchical Swin Transformer encoder computes attention inside local non-overlapping 3D windows to reduce computational cost, while shifted-window partitioning permits cross-window communication. Multi-resolution encoder features are then connected to a fully convolutional decoder through skip connections, enabling precise segmentation of complex organs and lesions in volumetric multi-modal scans [14], [34].

2.3.5 Loss Functions and Class Imbalance Metigation

Training deep segmentation networks requires loss functions that can guide optimization under severe class imbalance. In clinical three-dimensional datasets, the target lesion may occupy less than 0.1% of the voxels, while background and healthy anatomy dominate the volume. Under such conditions, an unsuitable loss function can encourage the model to predict only the majority class, producing deceptively low training loss while failing to detect clinically meaningful regions [15], [32].

Binary Cross-Entropy (BCE) treats each pixel or voxel as an independent binary classification problem. For N samples, ground-truth labels yi, and predicted probabilities pi, BCE is defined as [30]:

$$L_{BCE} = -\left(\frac{1}{N}\right)\sum_i [y_i \log(p_i) + (1-y_i)\log(1-p_i)] \quad (2.18)$$

Although BCE can be effective for balanced data, it is often dominated by the large number of easy background voxels in medical segmentation. Dice Loss addresses this limitation by evaluating regional overlap rather than independent pixel-wise classification. Derived from the ScrenseenDice coefficient, it is formulated as [17]:

$$L_{Dice} = 1 - \frac{(2\sum_i p_i y_i + \varepsilon)}{(\sum_i p_i + \sum_i y_i + \varepsilon)} \quad (2.19)$$

The smoothing term **ε** prevents division by zero and stabilizes gradient computation. Since Dice Loss emphasizes overlap between the predicted and ground-truth masks, it is naturally more robust to class imbalance. Focal Loss further mitigates imbalance by reshaping cross-entropy according to prediction confidence. It down-weights easy examples and focuses optimization on hard or ambiguous voxels, such as lesion boundaries [16]:

$$L_{Focal} = -\alpha t(1 - pt)\gamma \log(pt) \quad (2.20)$$

Here, γ is the focusing parameter and αt is an optional class-balancing term. The modulating factor (1 − pt)γ reduces the contribution of well-classified background samples, preventing them from dominating gradient updates during training [16].

2.3.6 Quantitative Evaluation Metrics in Medical Imaging Segmentation

The performance of medical image segmentation models must be assessed by quantitatively comparing predictions against high-quality ground-truth annotations. This evaluation is based on voxel-wise confusion matrix outcomes: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) [19], [20]. In this context, TP denotes pathological voxels correctly identified as pathological, FP corresponds to healthy voxels incorrectly predicted as pathological, TN represents correctly identified healthy voxels, and FN denotes missed pathological voxels.

The Dice Similarity Coefficient (DSC), equivalent to the F1-score in segmentation, measures the spatial overlap between the predicted mask S and the ground-truth mask GT [19]:

$$DSC = \frac{2|GT \cap S|}{|GT| + |S|} = \frac{2TP}{2TP + FP + FN} \quad (2.21)$$

DSC values range from 0, indicating no overlap, to 1, indicating perfect alignment. Intersection over Union (IoU), also known as the Jaccard Index, measures the ratio between the intersection and the union of the two masks [19]:

$$IoU = \frac{|GT \cap S|}{|GT| + |S|} = \frac{TP}{TP + FP + FN}$$
$$IoU = \frac{DSC}{(2 - DSC)} \quad (2.22)$$

IoU penalizes spatial disagreement more strictly than DSC. Sensitivity, or Recall, measures the ability to detect pathological voxels and reduce missed findings, while Specificity measures the ability to correctly identify healthy tissue and reduce false alarms [15], [19]:

$$Sentivity = \frac{TP}{TP + FN} \qquad Specificity = \frac{TN}{TN + FP} \quad (2.23)$$

Standard voxel-wise accuracy should generally be avoided in medical segmentation because it is highly biased by the predominance of healthy background voxels. A model may obtain an artificially high accuracy by predicting background everywhere, despite failing to segment the actual lesion. Consequently, overlap-based metrics such as DSC and IoU, which exclude True Negatives, are preferred for objective evaluation of clinical segmentation tasks [15].

2.3.7 Discipline Specific Best Practices: Validation Protocols, Patient-Level Splitting, and Data Privacy

Medical imaging studies involving deep learning require validation protocols that prevent methodological bias and protect patient privacy. Among the most critical risks is data leakage, which occurs when information from the validation or testing set influences the model during training. This produces overly optimistic performance estimates during development but poor generalization when the system is deployed on new patients in clinical settings [21], [22].

Slice-Level vs. Patient-Level Splitting: In volumetric CT or MRI data, each patient scan is composed of multiple contiguous two-dimensional slices. These slices are highly correlated because they share the same anatomy, acquisition noise, scanner characteristics, and patient-specific texture. If the dataset is split at the slice level, images from the same patient may appear in both training and testing partitions. Under these conditions, the model can memorize identity-specific anatomy rather than learn generalizable pathology, a phenomenon known as identity confounding [21], [23].

A rigorous validation strategy therefore requires patient-level splitting, where all slices from a given subject are assigned to one partition only. This rule also applies to derived views, reformatted slices, and augmented variants, ensuring that the test set represents truly unseen patients rather than near-duplicates of training data [21], [23].

Table 2- 1: Impact of Slice-Level and Patient-Level Paradigms on Model Evaluation

Parameter	Slice-Level Split (Methodologically Flawed)	Patient-Level Split (Clinical Best Practice)
Partitioning Rule	3D volumes are decomposed into 2D slices, which are randomly shuffled and split [21].	Dataset partitioning is performed strictly at the subject identifier level [23].
Risk of Data Leakage	Critical: highly correlated slices from the same patient are shared across training and testing sets [21].	None: slices from a single patient are strictly isolated within one partition [23].
Inflated Performance Bias	Severe: yields overly optimistic test accuracies, often 95%-99% [21].	None: reflects the model's true diagnostic performance, typically 60%-90% [21].
Identity Confounding	High: the model memorizes patient-specific anatomical textures [23].	None: the model is forced to learn generalizable pathological features [23].
Augmentation Protocol	Often applied before partitioning, exacerbating target leakage [21].	Strictly applied only to the training split after partitioning [21].
Clinical Generalizability	Poor: the model fails when applied to scans from new patients or other clinics [22].	High: provides a reliable estimate of performance in clinical practice [22].

Late Splitting of Augmented Data: Another major source of leakage occurs when augmentation is applied to the entire dataset before splitting. Transformations such as rotation, cropping, scaling, or intensity modification may create near-duplicate versions of the same image. If these augmented clones are distributed across training and validation sets, the model is evaluated on examples that are nearly identical to those seen during training, resulting in inflated metrics and reduced credibility [21]. The appropriate protocol is to split patients first and apply augmentation only to the training partition.

Global Preprocessing Contamination: Global preprocessing contamination occurs when normalization, feature selection, or intensity scaling is computed across the entire dataset before splitting. This allows statistics from validation or testing data, such as global means and variances, to influence the training process. To prevent this contamination, all preprocessing parameters must be estimated only from the training set and then applied unchanged to validation and test sets [21].

2.3.7.1 Patient Data Privacy and DICOM De-Identification Standards

Clinical imaging datasets must be handled under strict privacy frameworks, including HIPAA in the United States and GDPR in the European Union [35]. Medical images are commonly stored in the Digital Imaging and Communications in Medicine (DICOM) format, which contains both pixel data and metadata describing patient identity, acquisition parameters, scan dates, institutional information, and clinical context [36]. Before research use or data sharing, Protected Health Information (PHI) must be removed or masked from DICOM headers and, when necessary, from the image pixels themselves [36], [37].

Under the Safe Harbor approach and DICOM de-identification procedures, key identifiers must be handled systematically. The Patient Identity Removed tag (0012,0062) should be set to "YES"; Patient Name (0010,0010) should be replaced with a generic placeholder such as "ANONYMOUS"; and Patient ID (0010,0020) should be substituted by a secure pseudo-random code that permits longitudinal tracking without revealing the hospital identifier [37].

When disease progression must be studied over time, temporal relationships between sequential scans should be preserved while absolute dates are shifted by a consistent randomized offset for each patient. The Longitudinal Temporal Information Modified tag (0028,0303) is then set to "MODIFIED" [37]. Privacy protection must also extend to pixel-level PHI, such as patient names or dates burned into ultrasound or radiography images,

which must be permanently redacted. Finally, volumetric head CT or MRI data may require defacing algorithms because surface renderings can reconstruct facial features and introduce re-identification risk through facial recognition software [35].

2.3.7.2 Model Explainability and Interpretability (Explainable AI)

Because deep learning models operate as "black boxes," clinical adoption requires providing transparent explanations for their diagnostic predictions [29]. Clinicians must be able to understand the underlying reasoning of an AI model before relying on its recommendations for patient care [29].

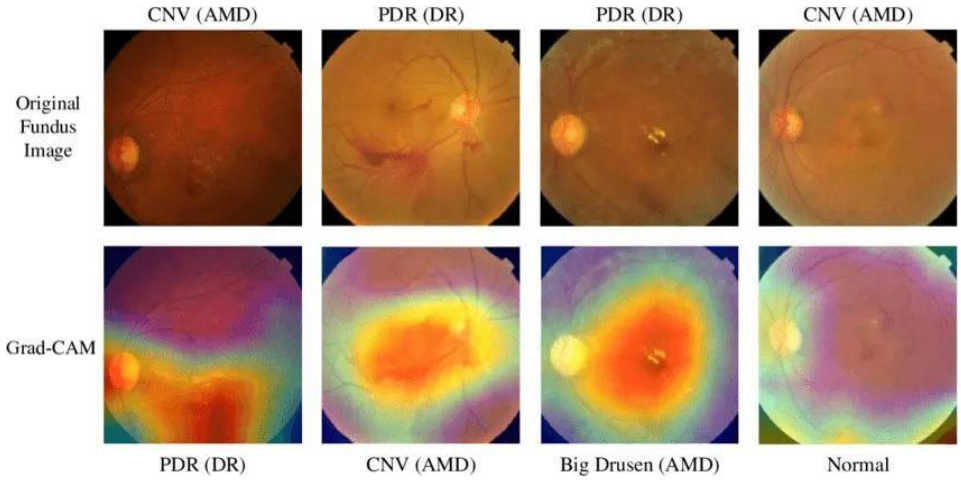


Figure 2-7: Saliency Mapping via Gradient-Weighted Class Activation Mapping (Grad-CAM).

To achieve this, **Gradient-Weighted Class Activation Mapping (Grad-CAM)** is widely used [29]. Grad-CAM uses the gradients of a target clinical class flowing into the final convolutional layer of a CNN to produce a coarse 2D heatmap [29]. This heatmap highlights the regions of the medical image that most influenced the model's prediction, helping clinicians verify that the network is focusing on actual pathology rather than background artifacts [30].

While useful, researchers must acknowledge that these saliency maps are coarse visual aids and can lack the pixel-level precision required to detect subtle, micro-scale abnormalities, necessitating ongoing research into high-fidelity explainability methods [30].

2.4 Conclusion

This chapter presented the essential foundations of medical imaging and computer-aided diagnostic systems. It first showed that medical images are produced through complex

physical principles and digital processing methods specific to each imaging modality, including radiography, CT, MRI, ultrasound, and PET.

It then highlighted the transition from traditional handcrafted-feature approaches to modern deep learning methods, which can automatically learn meaningful representations from medical images. CADe, CADx, and CADt systems illustrate the growing role of artificial intelligence in lesion detection, disease characterization, and clinical case prioritization.

Finally, this chapter emphasized the importance of suitable segmentation architectures, adapted loss functions, reliable evaluation metrics, and rigorous validation protocols. Patient-level data splitting, prevention of data leakage, medical data protection, and model explain ability are essential conditions for ensuring the reliability and clinical acceptability of AI-based diagnostic systems.

Therefore, the concepts discussed in this chapter provide the theoretical basis for the design, evaluation, and integration of deep learning methods in medical image analysis.

Chapter 3
Theatrical Foundations of Deep Learning

3.1 Introduction

Artificial Intelligence (AI) is the broad field focused on creating systems capable of mimicking aspects of human intelligence, such as recognition, reasoning, problem-solving, and decision-making. To achieve this objective, Machine Learning (ML) emerged as one of the most important branches of AI, allowing computers to learn from data and improve their performance over time without being explicitly programmed for every individual task. Rather than depending on fixed instructions, ML algorithms analyze data, identify hidden patterns, and use these patterns to make predictions or decisions automatically.

As the complexity and volume of data increased, traditional ML approaches faced limitations in handling highly unstructured information, which led to the development of Deep Learning (DL). DL extends ML by employing deep neural networks (multi-layered computational architectures inspired by the human brain) that are capable of automatically extracting complex and hierarchical features from large datasets.

At the core of deep learning are neural networks, which consist of interconnected artificial neurons arranged into input, hidden, and output layers. During the learning process, these networks adjust the weights of the connections between neurons using techniques such as backpropagation and gradient descent in order to minimize errors and improve prediction accuracy. Because of their ability to model complex relationships within data, neural networks have become highly effective in applications such as image recognition, speech processing, natural language understanding, and predictive analytics. Building upon these neural network architectures, several specialized models have been developed to address specific learning tasks.

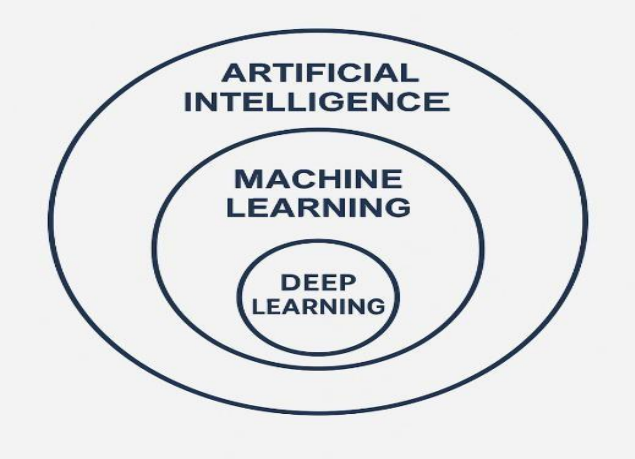


Figure 3- 1: Relationship between AI, Machine Learning and Deep Learning.

Among these models, autoencoders represent a unique type of neural network specifically designed for unsupervised learning. Their objective is to reconstruct the input data at the output layer while passing through a bottleneck layer that compresses the information into a lower-dimensional representation. Consequently, the network learns the most significant and meaningful features of the data while ignoring redundant information. Owing to this capability, autoencoders are widely applied in tasks such as noise removal, feature extraction, dimensionality reduction, data compression, and unsupervised anomaly detection, where detecting unusual patterns is essential.

3.2 Artificial Intelligence (AI)

Artificial Intelligence (AI) is an interdisciplinary field focused on developing systems that emulate human cognitive functions: learning, reasoning, pattern recognition, decision-making, and problem-solving, through computational and mathematical methodologies. Modern AI leverages frameworks such as TensorFlow , Keras , and Scikit-learn to implement learning paradigms including supervised, unsupervised, and deep learning. Key technical processes involve data preprocessing , feature extraction, dimensionality reduction, and representation learning to enhance model performance.

Prominent deep learning architectures such as autoencoders , convolutional neural networks (CNNs), and long short-term memory (LSTM) networks enable automatic pattern discovery in structured and unstructured data. Consequently, AI is extensively applied in time series forecasting, computer vision, natural language processing, speech recognition,

healthcare analytics, and intelligent automation, where adaptive learning and accurate prediction are critical.

3.2.1 Machine Learning

Machine Learning (ML) constitutes a subfield of artificial intelligence concerned with the design of algorithms and systems that autonomously learn from empirical data, improving their performance through experience without reliance on explicit, task-specific programming. Rather than depending on manually encoded rules, ML models discern latent patterns, statistical regularities, and structural relationships within data to perform prediction, classification, or decision-making tasks. This paradigm is particularly efficacious for data-intensive and ill-defined problems, exemplified by computer vision where traditional rule-based approaches are infeasible due to the complexity and variability of unstructured visual information.

In machine learning, the primary objective is to learn a mapping from input patterns to output values, typically represented as a function $y(x)$. For example, given image data defined by pixel intensities and corresponding categorical labels, the learning algorithm aims to approximate this mapping such that it can correctly predict outputs for unseen inputs. However, this mapping is often highly complex and nonlinear, rendering it inaccessible to models restricted to linear transformations [39]. To address this, researchers traditionally employ hand-optimized features, which transform the input space into a representation from which a simpler, learnable mapping can be derived. Nevertheless, this approach is task-specific, and feature engineering remains a difficult and non-transferable process [39]. Consequently, alternative algorithms that learn feature representations directly from data are also of interest.

Model parameters are tuned using a training set n input vectors $x_1, x_2 \ldots, x_n$ and , where applicable , corresponding target vectors. This training (or learning) phase adjusts the parameters that define the mapping $x_i \mapsto y(x_i)$. Following training, the model's performance is evaluated on a separate test set to assess its generalization the ability to predict accurately on previously unseen samples. Since the training set typically represents only a small fraction of the possible input space, achieving strong generalization is a central objective in machine learning [38].

Practical applications span image analysis, speech recognition, natural language processing, recommender systems, predictive analytics, autonomous control, and medical diagnosis. Depending on data characteristics and task requirements, ML employs supervised, unsupervised or reinforcement learning paradigms. The advent of deep learning architectures has further enabled automated feature extraction from massive structured and unstructured datasets, solidifying ML's central role in advancing intelligent systems and enabling novel solutions to increasingly complex problems.

3.2.2 Deep Learning

Deep Learning is a specialized subset of machine learning, which itself is a branch of artificial intelligence (AI). At its core, deep learning seeks to mimic the workings of the human brain by processing data through architecture of interconnected nodes, or neurons, arranged in multiple layers, hence the term "deep." Unlike traditional algorithms that rely on structured, labeled data and manual feature extraction, deep learning models automatically discover the hierarchical features needed for tasks like classification, prediction, or generation.

For example, a deep learning model analyzing an image might identify simple edges in its earliest layers, then combine those into shapes like eyes or wheels in intermediate layers, and finally assemble those shapes into recognizable objects, such as a face or a car, in the deepest layers.

This ability to learn directly from raw data, such as text, sound, or pixels, without human intervention, has driven breakthroughs in areas ranging from autonomous driving and voice assistants to medical diagnosis and language translation.

3.2.3 Difference between Machine Learning and Deep Learning

The difference between machine learning and deep learning, can be summarized as fellow:

Table 3-1 : Difference between Machine Learning and Deep Learning.

Machine Learning	Deep Learning
Can train on smaller dataset	Requires large amounts of data
Requires more human intervention to correct and learn	Learns on its own from environment and past mistakes
Shorter training and lower accurancy	Longer training and higher accurancy
Makes simple and linear correlations	Makes non-linear and complex correlations

3.3 Neural Networks and Deep Learning

3.3.1 Artificial Neural Network (ANN)

Artificial Neural Network is a network which is inspired by biological neural systems and consists of interconnected nodes similar to neurons. it is defined by three main components: node characteristics, network topology, and learning rules [40].

Node characteristics: determine how each node processes information, including the number of inputs and outputs and the weights assigned to them, and the used activation function.

Network topology: refers to how the nodes are structured and connected.

Learning rules: govern how the weights are initially set and how they are updated during training.

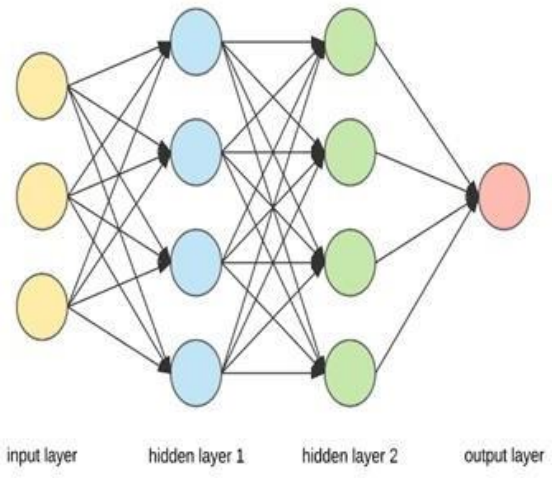


Figure 3- 2: Artificial Neural Network

3.3.2 Recurrent Neural Network (RNN)

Recurrent Neural Network is a type of deep learning model that is great at handling sequences of information. What makes RNNs special is that they have a simple form of memory. Unlike regular neural networks that forget everything after each piece of data, RNNs remember what they just saw and use that to understand the next piece. Think of it like reading a sentence: you don't forget the first word when you get to the second word. Your brain keeps the context. RNNs work the same way. This makes them perfect for tasks like predicting the next word as you type on your phone, translating languages, or understanding the mood of a movie review. The main weakness, however, is that RNNs have trouble remembering things from a very long time ago in a long sequence.

This figure illustrates the architecture of a Recurrent Neural Network. The inputs $x_1, x_2, \ldots, x_n$ are processed sequentially through hidden states $h_1, h_2, \ldots h_n$, where each hidden state keeps information from the previous step . The network then generates predictions $y_1, y_2, \ldots, y_n$.

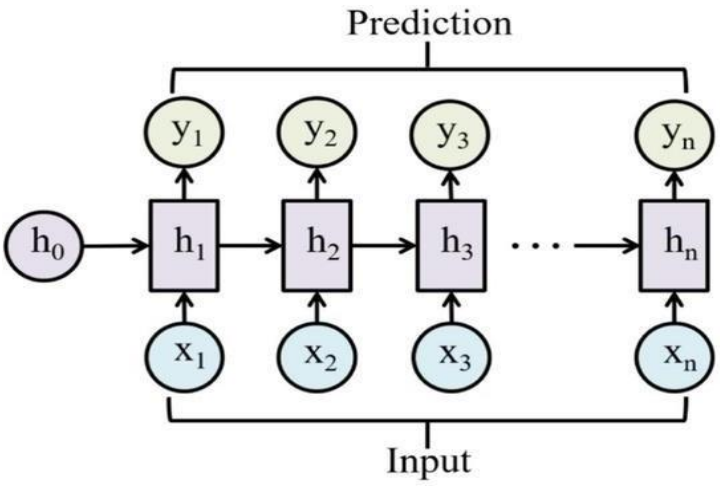


Figure 3- 3: Recurrent Neural Network architecture.

3.4 Convolutional Neural Network

Convolutional Neural Networks (CNNs) represent a crucial breakthrough in deep learning, primarily designed to solve complex image classification problems. Taking cues from the human visual system, CNNs mimic the way we process visual information

hierarchically, making them highly effective at handling spatial data structures. This inherent architecture makes them exceptionally suitable for interpreting and analyzing visual imagery across numerous fields.

A core strength that establishes CNNs as foundational in computer vision is their capacity for automated feature extraction. Unlike older methods that required manually selecting and engineering features, CNNs learn to identify the most relevant features directly from the input data itself. This capability has radically improved image classification , driving significant progress in object recognition, video processing, and medical diagnostics. By autonomously learning these features during training , CNNs deliver substantial gains in both the efficiency and accuracy of image classification tasks [42].

3.4.1 Overall Architecture

Convolutional Neural Network is a type of artificial intelligence system built on multi-layer neural networks, capable of recognizing, classifying, detecting, and segmenting objects within images. CNNs can be trained directly on input data without relying on manual feature extraction.

They are widely employed in areas such as visual recognition, medical images analysis, image segmentation, and various other fields. Their advantage over traditional networks lies in their ability to automatically identify and extract important features from the input data, eliminating the need for human intervention; they combine both features extraction and classification [43].

The structure of a Convolutional Neural Network (CNN) is built by stacking five main types of layers: input layer, convolutional layer, pooling layer, and fully-connected layer and an output layer. When arranged together, these layers form the complete CNN architecture and they are divided into two parts: features extraction and classification [44, 45].

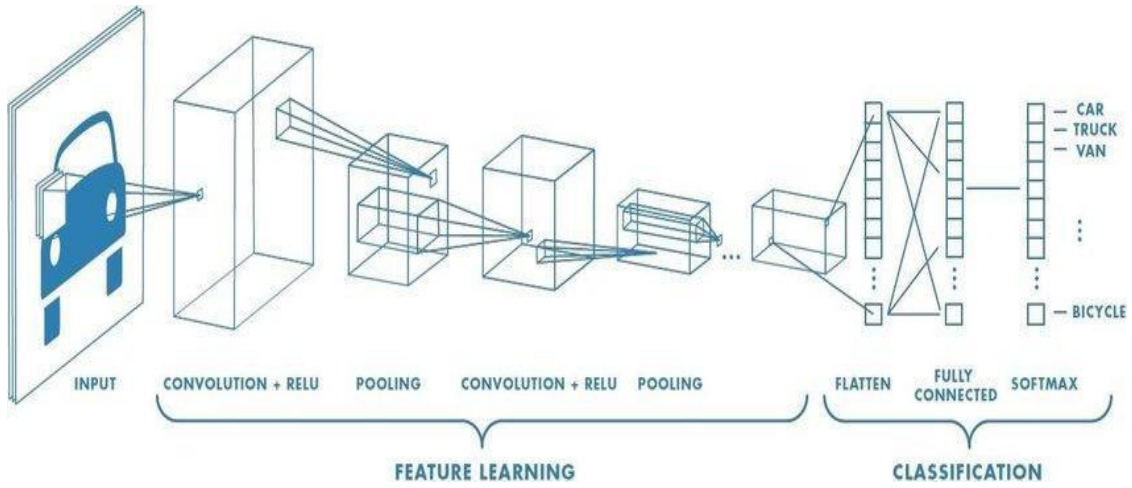


Figure 3- 4: Architecture of convolutional network

CNNs use multiple convolution and pooling layers instead of just one because each layer helps the network learn increasingly complex features.

The first layers detect simple patterns such as edges and corners, while deeper layers capture more advanced features like shapes, textures, and complete objects. Pooling layers gradually reduce the size of the feature maps, preserving the most important information and improving computational efficiency.

During these stages, activation functions (such as ReLU) are applied after convolution operations to introduce non-linearity into the network. This non-linearity enables the CNN to learn complex and non-linear relationships within the data, which would not be possible using only linear transformations. After the feature extraction process, the network uses fully connected layers, followed by a Softmax layer, which converts the final outputs into probability values for each class. This layer helps the model make a clear classification decision by selecting the class with the highest probability. Using several layers in sequence allows the network to understand the input step by step, from simple features to high-level representations, something a single layer cannot achieve.

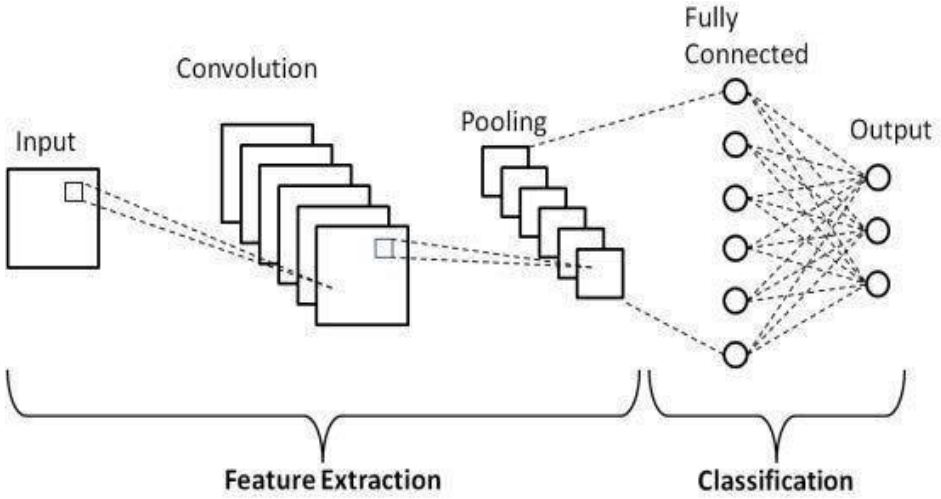


Figure 3- 5: Schematic diagram of basic Convolutional Neural Network architecture.

The fundamentals operations of a CNN can be summarized into these stages:

3.4.1.1 Input Layer

Similar to other Artificial Neural Networks (ANNs), this layer holds the pixel values of the input image.

3.4.1.2 Convolutional Layer

which uses a specialized linear operation called convolution, this involves applying a small , optimizable grid of parameters, known as the kernel (or filter) that slide across the spatial dimensions of the input data to extract relevant features[46].

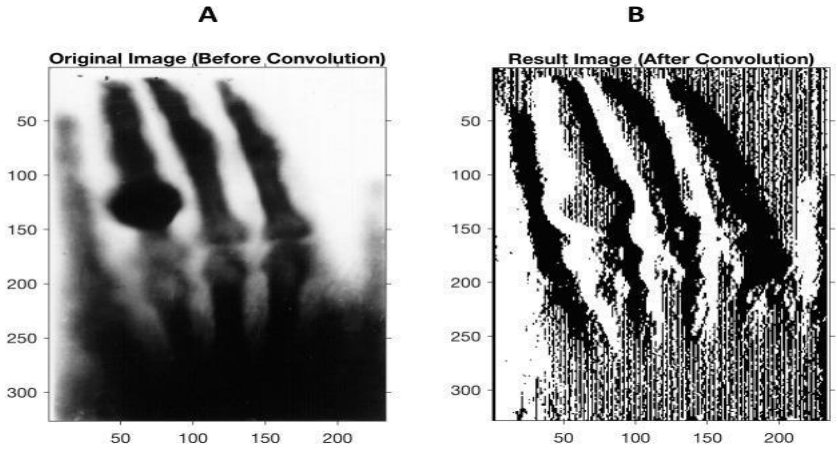


Figure 3- 6: Convolution Operation on an X-ray image

Here's how the convolution works [46]:

-The kernel which is a small grid of numbers slides over the input data

-At each time, the numbers in the kernel are multiplied by the numbers which are cover the input data , and all those results are added up.

-This sum becomes one number in the new output map.

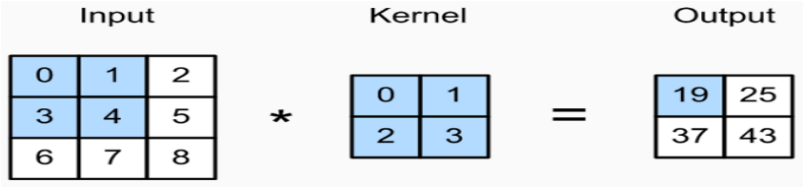


Figure 3- 7: The convolution operation**.**

In figure 7, the 19 on the blue background in the output is calculated as follow:

19 = (0*0)+(1*1)+(3*2)+(4*3)

- Here is the input is **k=(3×3)** and the kernel is **L=(2×2)**

- The output matrix is based on the equation: **(k-L+1)** which is 2 in this example so the output then is **(2×2)**

- Also, the kernel here is with the stride of 1 (designating the desired step size).

The goal of convolution is to automatically extract important features from input data.

One limitation of the convolution operation is the potential loss of information along the image boundaries, as edge pixels are only partially covered by the filter and may not be fully processed.

A simple and effective way to address this issue is the use of **ZeroPadding**, where additional zero-valued pixels are added around the image borders. In addition to preserving edge information, zero padding also allows control over the spatial dimensions of the output feature map.

Why we need padding before applying convolution?

- **Preserve spatial size**: Without padding, each convolution reduces the size of the feature map. Padding allows the output to have the same width and height as the input.

- **Better edge detection**: Padding lets the convolution filter see the edges and corners of the input properly. Without it, the borders are used less in feature extraction.
- **Control output dimensions** : By adjusting padding, we can control how the size of the feature map changes through layers.

The padding operation follows the formula:

$$output\ size = \frac{K - L + 2P}{s} + 1 \tag{3.1}$$

K: the input image size ; L: the kernel size ; P: the padding and s: the stride.

K, L, P and s are so called, the hyperparameters that can be set for this layer.

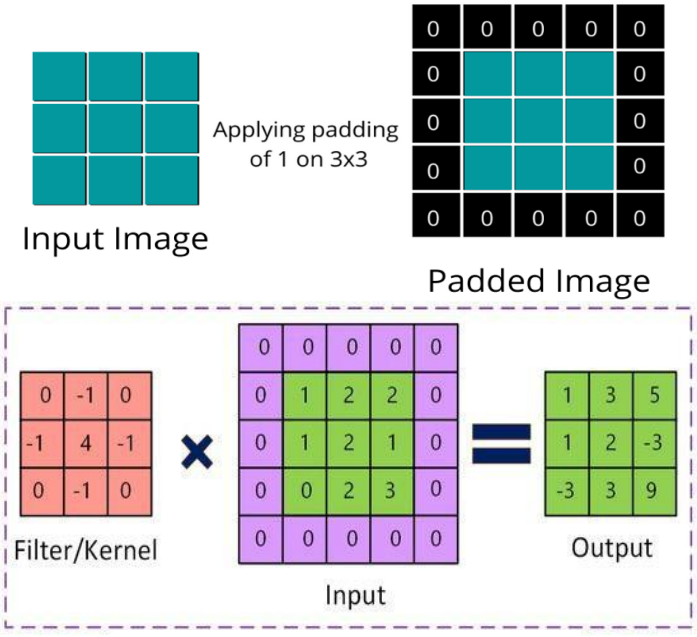


Figure 3-8: ZeroPadding and Convolution

3.4.1.3 Activation Function

The activation function is an essential component of CNN layers. After convolution, the filter output is passed to an activation function, which introduces non-linearity into the network.

An activation function is a function applied to neurons in a neural network to determine their output. It enables the network to retain and transform important features, enhances the model's expressive power, and allows the neural network to learn and solve complex problems, giving it intelligent behavior [50].

Linear Activation function: It can be written as : **F(x)=C*Y,** the input values are multiplied with the constant parameter C, which is weight of each neuron.

Non-Linear Activation functions: are used in modern neural networks. They allow the model to design a complicated mapping between the network's input and output, which is essential for complex learning and modeling systems [51].

ReLU (Rectified Linear Unit): It is the most commonly used activation function in CNNs for feature extraction . The main purpose of an activation function is to determine how neurons respond to input signals. Depending on the chosen function, the output values are mapped to specific ranges, such as 0 to 1 or −1 to 1.

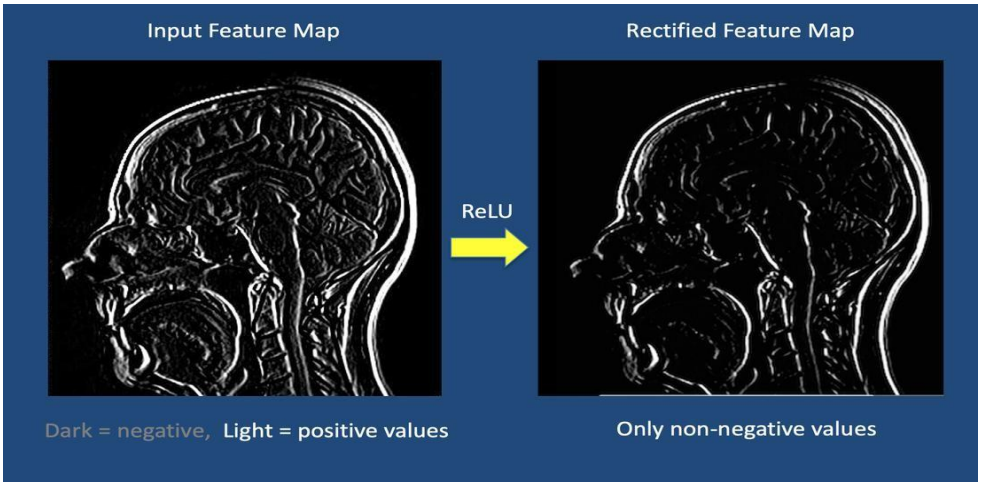
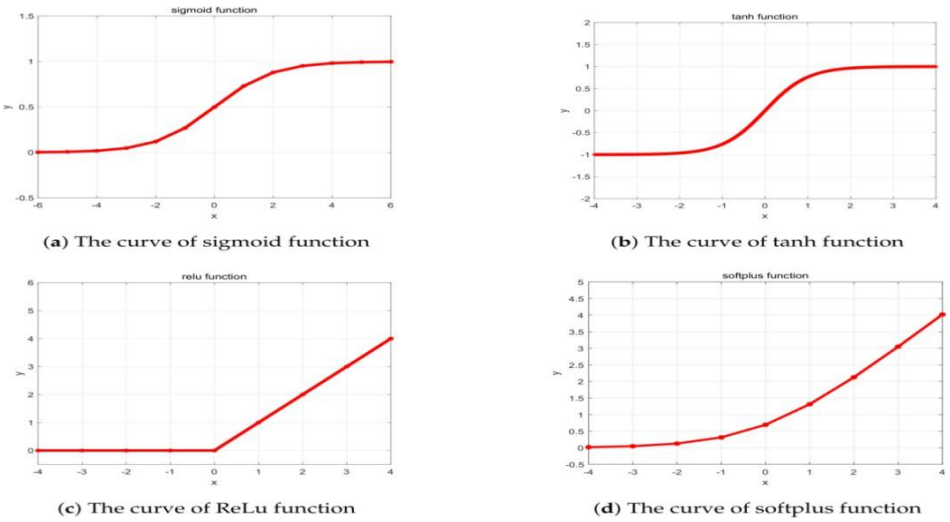


Figure 3-9: The impact of ReLU function in brain MRI image.

The activation function is the core of a neural network's structure, and common activation functions include: Sigmoid , Tanh , ReLu and Softplus, which can be seen in **figure 10**[50].



(a) The curve of sigmoid function
(b) The curve of tanh function
(c) The curve of ReLu function
(d) The curve of softplus function

Figure 3-10: Graphs of Sigmoid function, Tanh function, ReLu function, and softplus function.

Here is an explanation table of most common Activation function:

Table 3-2 : Comparative table between non-linear activation functions [52, 53].

Activation function	Explanation
Sigmoid function	It maps any real-valued input to an output **between** 0 **and1.** It has **sigmoid al characteristic curve**, similar to biological neurons. It is **defined or all real input values** and has a **non-negative derivative** (easy to calculate) **Function**: $F(x) = \frac{1}{1+e^{-x}}$
Tanh function	It is an improved version of sigmoid it is **symmetric** function **centeredat0** The outputs between -1 and 1 **Function**: $F(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
ReLU(Rectified	It is **defined** as **zero for non-positive inputs** and **equal to the input for**

Linear Unit)	**positive values**. It helps mitigate the **vanishing gradient problem** and allows **faster convergence** during training. **Function:** $F(x) = \begin{cases} 0x \leq 0 \\ 1 \quad x > 0 \end{cases}$
Softplus function	It is a differentiable approximation of ReLU For **negative inputs**, it outputs a small positive value instead of zero. For **large positive inputs**, it behaves almost like a linear function. **Function**: $f(x) = ln(1 + e^x)$

3.4.1.4 Pooling Layer

This layer reduces the spatial dimensionality of the feature maps, while keeping the most important informations. We have three types of pooling: **Max pooling**, **Average pooling** and **Min pooling** [54].

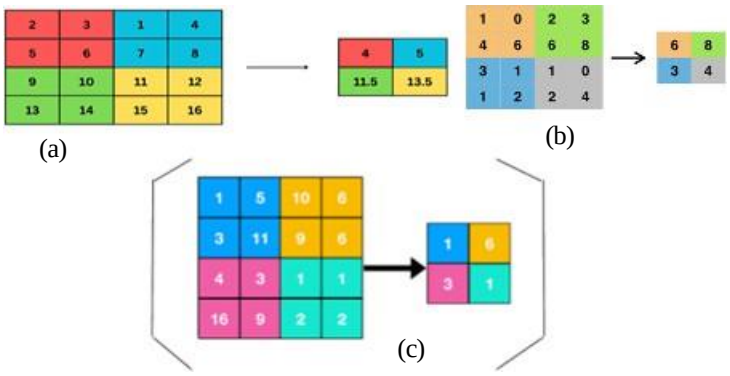


Figure 3-11: Average , Max , and Min Pooling

The **goal of pooling layer** is to produce a summary statistic of the inputs and to reduce the dimensions of the feature map [54].

3.4.1.5 Fully-Connected Layer (Dense Layer)

This layer in a CNN connects every input from the previous layer to every output neuron through learnable weights. It takes the features extracted by convolution and pooling layers and maps them to the network's final outputs, such as class probabilities in classification tasks.

This layer work by taking an input vector, multiplying it by a weight matrix, adding a bias, and then applying an activation function. It is defined by the formula [53]:

$$FC(x) = f(W \times x + b) \tag{3.2}$$

W is the weight matrix that contains learnable parameters which control how strongly each input affects each neuron

b is the bias vector, The bias allows the neuron output to be shifted independently of the input, improving the model's flexibility, and **f** is the Activation function.

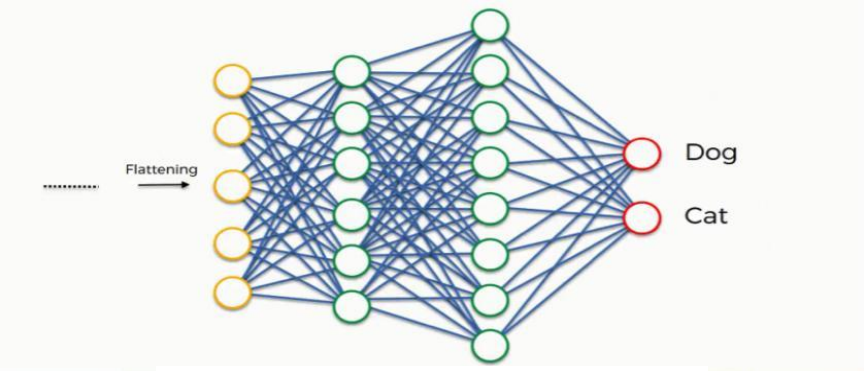


Figure 3-12: The Dense Layer in CNNs.

Before entering a fully connected layer the **feature maps are 'flattened** ' transformed from 2D matrix into a **one-dimensional (1D) vector,** so that each neuron can receive input from every element of the vector. This flattening step ensures that the dense layer can process all the extracted features simultaneously and produce the final predictions [47-51].

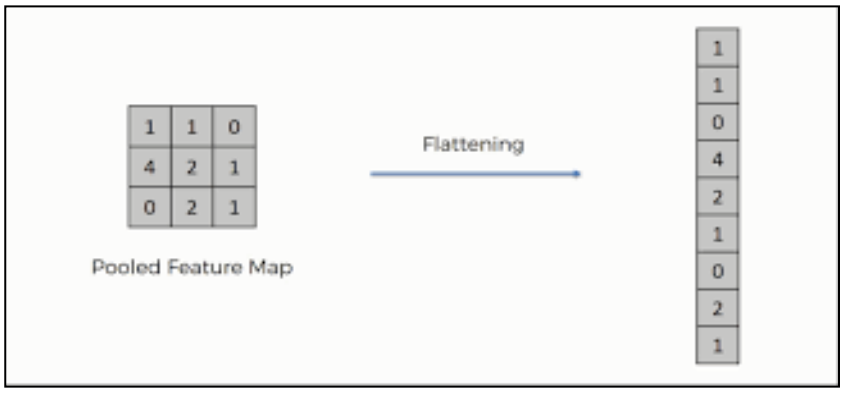


Figure 3-13: The Flattening Process in CNNs.

3.4.1.6 Softmax Layer

It is typically used as the final layer in a convolutional neural network for classification tasks. Its main role is to transform the raw output values (logits) produced by the previous fully connected layer into a probability distribution over the target classes. Each output value represents the probability that the input belongs to a specific class, and the sum of all probabilities equals one. By assigning higher probabilities to the most likely classes.

The Softmax layer enables the model to make a clear and interpretable classification decision by selecting the class with the highest probability.

3.4.2 Top CNN Architecture for Medical Imaging

The following established architectures have demonstrated robust performance across a wide range of medical imaging tasks.

3.4.2.1 Residual Network (ResNet)

The deep residual network **(ResNet)**, introduced by He et al. in 2016, is a deep learning architecture designed to address challenges associated with training very deep neural networks. In general, deep learning models require substantial training time and are often constrained by the number of layers that can be effectively trained (can reach 152 layers).

ResNet overcomes these limitations through the use of skip connections, also known as shortcut connections. A key advantage of this architecture is that its performance does not degrade as network depth increases. In addition, the computational complexity is reduced, and network training becomes more efficient. The **ResNet** architecture incorporates skip connections across two to three layers that typically include **ReLU** activation and batch normalization. He et al. demonstrated that **ResNet** outperforms other architectures in image classification tasks, confirming its strong capability for effective image feature extraction [18].

ResNet Architecture and Description

Residual block structure:

ResNet uses skip connection (shortcut) to allow the network to learn residual functions

The residual block is defined as:

$$y = F(x, W) + x \tag{3.3}$$

x: the input.

y: the output.

F: the residual mapping.

ResNet blocks consist of two layers (for ResNet-18 and ResNet-34) or three layers (for ResNet-50 and ResNet-101).

The first two layers of ResNet use a **7×7** convolution and a **3×3** max-pooling layer.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 3-14: ResNet Architecture.

The above table of **Figure 14** provides a **structured overview of how different ResNet models are organized and scaled** as network depth increases. It shows that all ResNet variants share the **same overall architecture**, but differ in **depth, internal block composition, and computational cost**.

First, the table highlights that **all models follow the same five-stage design** (**conv1**to **conv5_x**). Across these stages, the **spatial resolution of feature maps decreases progressively** (from 112×112 to 7×7), while the **feature representation becomes richer**, as indicated by the increasing number of channels in deeper stages.

Second, the table reveals a **clear distinction between shallow and deep ResNet variants**.

ResNet-18 and **ResNet-34** rely on simpler blocks composed of **two** 3×3 **convolution layers**.

ResNet-50, ResNet-101, and ResNet-152 adopt a more complex block structure consisting of **three convolution layers (1×1, 3×3, 1×1)**.

This indicates that deeper networks do not merely add more layers, but **change the internal structure of their building blocks** to support increased depth.

Third, the table shows that **model depth is increased by repeating blocks within the same stages** rather than by introducing new stages. For example, the depth difference between **ResNet-50**, **ResNet-101**, and **ResNet-152** is achieved mainly by increasing the number of block repetitions in the conv3_x and conv4_x stages, while keeping the overall stage layout unchanged. This demonstrates a **systematic and modular design strategy**.

Finally, the FLOPs row provides an interpretation of the **computational implications** of increasing depth. As the number of layers and repetitions increases, the computational cost rises significantly from 1.8×10^{9} for ResNet-18 to 11.3×10^{9} for ResNet-152 indicating a **trade-off between model depth and computational efficiency**(for more details see Appendix A).

3.4.2.2 VGG-16 (Visual Geometry Group)

VGG-16 is a convolutional neural network (CNN) architecture introduced in 2014 by the Visual Geometry Group (VGG) at the University of Oxford. It represents a particular version of the VGG family developed for image classification tasks. The model is distinguished by its use of small 3×3 convolutional filters and its relatively deep structure. VGG16 is composed of 16 layers in total, including 13 convolutional layers followed by 3 fully connected layers. Moreover, it incorporates a technique known as “very deep supervision”, which applies additional supervision to intermediate layers, leading to improved accuracy and more stable optimization during training (for more details see Appendix A).

VGG-16 was trained on the ImageNet dataset, which contains approximately 14 million images categorized into 1000 classes. In 2014, it reached state-of-the-art results on the

ImageNet classification challenge. Since then, it has been extensively adopted as a foundational model for various computer vision applications, including object detection and semantic segmentation. The model's pre-trained weights are publicly accessible and can be easily integrated into other architectures, making VGG16 a widely preferred option among researchers and developers, as it offers a strong starting point without requiring training from scratch [56].

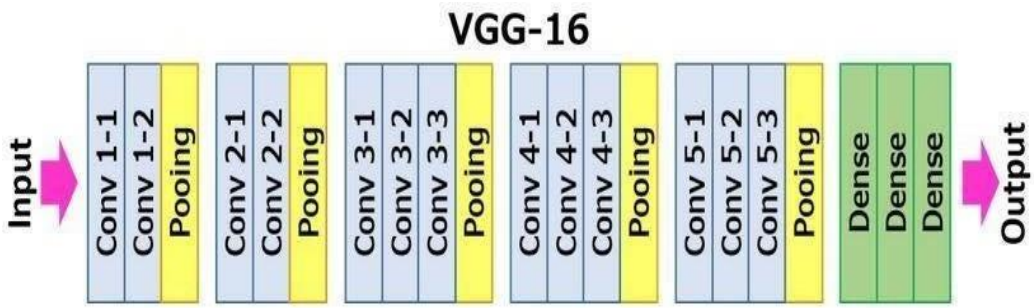


Figure 3-15: The VGG-16 Architecture.

3.4.2.3 VGG-19

VGG19 is a convolutional neural network architecture closely related to VGG16, but it consists of 19 layers. It was introduced in 2014 by the Visual Geometry Group (VGG) at the University of Oxford with the primary goal of performing image classification tasks. Like VGG-16, VGG-19 is characterized by the use of small 3×3 convolutional filters and a deep network structure(for more details see Appendix A).

The architecture is organized into five convolutional blocks, each containing several convolutional layers. The first two blocks include two convolutional layers each, while the last three blocks contain four convolutional layers each. These convolutional blocks are followed by three fully connected layers. In addition, VGG-19 applies the "very deep supervision" technique, similar to that used in VGG-16[55].

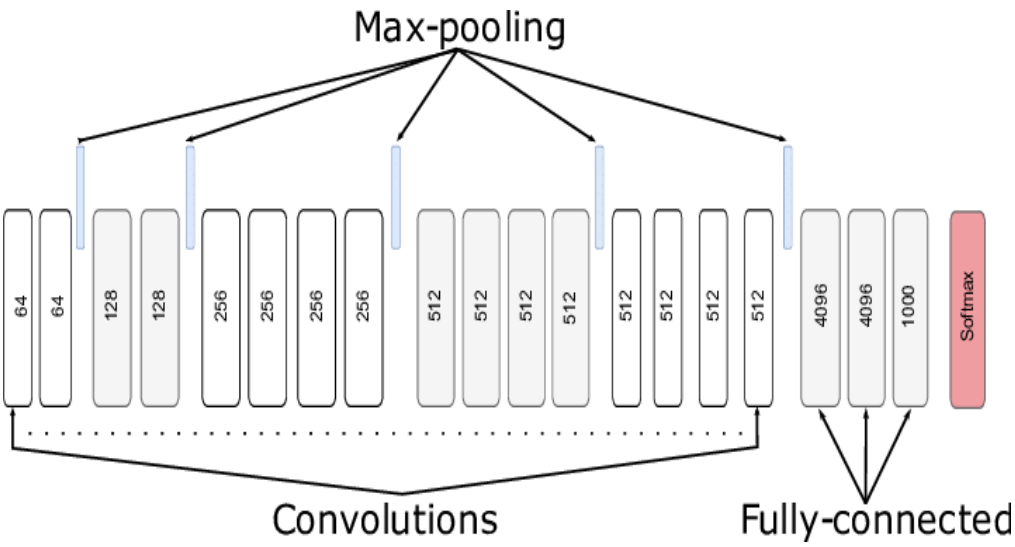


Figure 3-16: The VGG-19 Architecture.

3.4.2.4 AlexNet

AlexNet is a landmark convolutional neural network (CNN) architecture that fundamentally transformed the field of computer vision. As one of the earliest deep learning models to achieve groundbreaking success in large-scale image classification, it served as the catalyst that propelled deep neural networks into the mainstream. **AlexNet** introduced or popularized several key innovations including ReLU activation functions, extensive data augmentation, and efficient GPU-based training that are now standard in modern CNN design. Despite its comparatively large and simple architecture, it remains computationally feasible for resource-constrained environments and is widely available in pre-trained form, making it highly effective for transfer learning. Beyond its technical specifications, **AlexNet** holds enduring historical significance as the model that demonstrated the true potential of deep learning for visual recognition tasks, influencing nearly every architecture that followed [57].

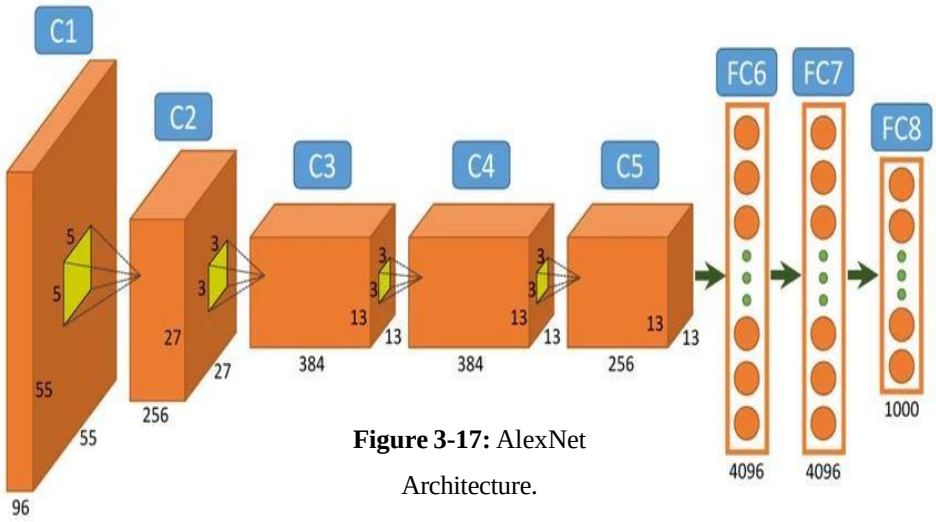


Figure 3-17: AlexNet Architecture.

3.5 Autoencoders

An **autoencoder** is an **unsupervised neural network architecture** designed to learn a compressed representation of input data and reconstruct the original input as accurately as possible. It consists of two main components: an **encoder**, which maps the input data x into a lower-dimensional latent representation ***h=f(x)*** which is a set of features called "latent space" or "Bottleneck", and a **decoder**, which reconstructs the input from this representation ***r=g(h)***.The main goal is to **capture the most important information** from the input in a smaller representation[58,59,60].

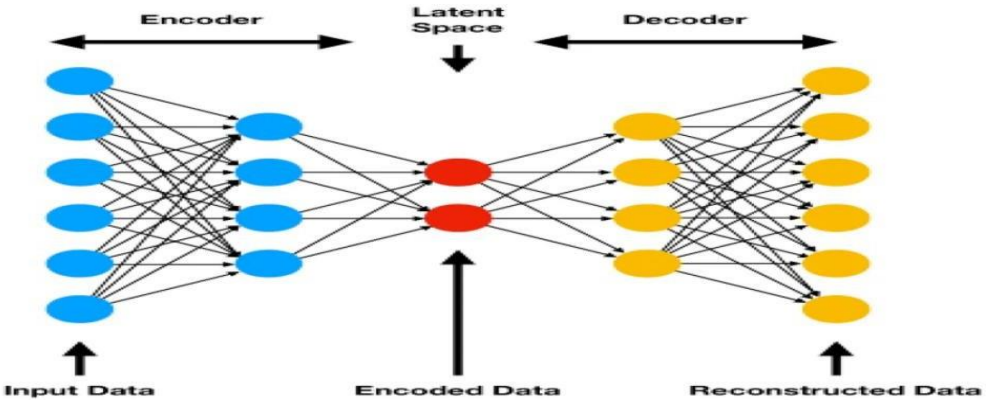


Figure 3-18: The Representation of an Auto-Encoder.

The encoder formula:

$$h(x) = x \times w + b \tag{3.4}$$

The decoder formula:

$$x' = h(x) \times w' + b' \tag{3.5}$$

Where **w** and **w`** represent the weight matrixes, **b** and **b`** represent the bias of encoder and decoder respectively (for more details see Appendix B).

An auto-encoder consists of three main components

The encoder: transforms the input data into a compressed representation, usually smaller than the original input, producing what is called the latent space.

The bottleneck (the latent space): a module that contains the compressed data(important features).

The decoder: A component that reconstructs the data from its encoded form, essentially "decompressing" it. The reconstructed output is then compared to the original input.

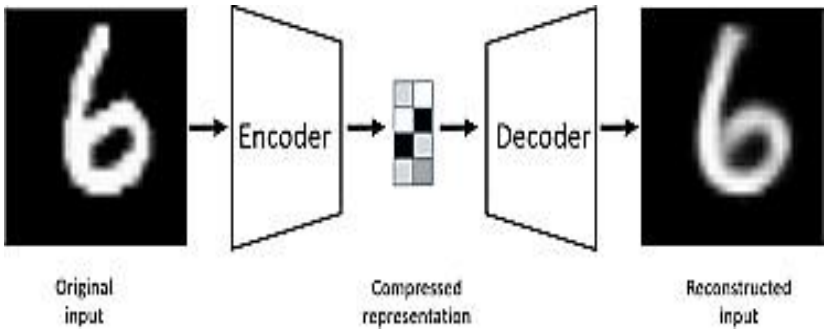


Figure 3-19: An example of Auto-Encoding process.

3.6 Different Autoencoder Types

3.6.1 Under-complete Autoencoder

It's one of the simplest types of autoencoders, where the **hidden layer is smaller than the input d < n**, forcing the network to compress the data and learn only the most important features.

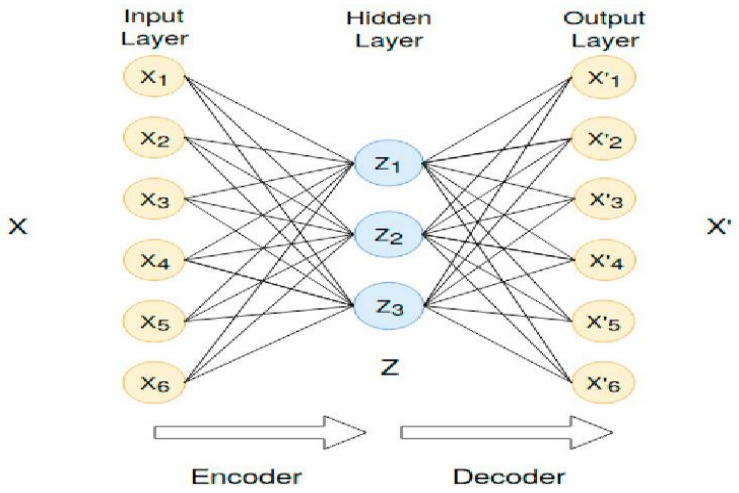


Figure 3-20: Architecture of Under-Complete Auto-Encoder.

3.6.2 Sparce Autoencoder (SAE)

It is a type of autoencoder that learns useful features by forcing most of its **hidden neurons** to remain inactive. The goal is to accurately reconstruct the input data while using only a small number of active neurons in the hidden layer. The sparsity here helps the model learn more meaningful and distinctive features, similar to how neurons in the human visual system respond selectively to specific patterns [62, 63].

Sparse autoencoders have a sparsity penalty (h), a value close to zero but not zero.

Sparsity penalty is applied on the hidden layer in addition to the reconstruction error. This prevents overfitting.

The sparsity is taking the highest activation values(for active neurons), and zero out the rest (for inactive neurons).

$$L = |x - g(f(x))| + \Omega(h) \tag{3.6}$$

$\mathbf{x - g(f(x))}$: 'reconstruction error' the difference between the original input and the reconstructed output.

$\mathbf{\Omega(h)}$: is the sparsity penalty which forces most neurons to be zero (inactive).

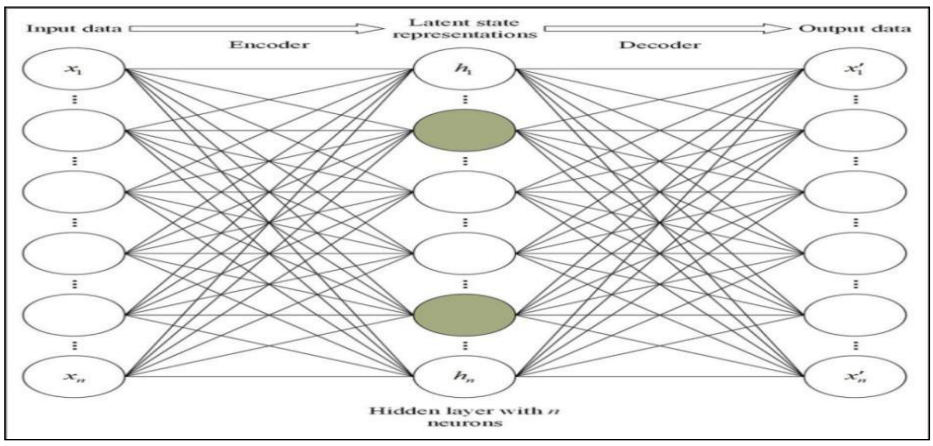


Figure 3-21: The Sparse Auto-Encoder architecture.

In the above figure, we can see three layers: **input**, **hidden (latent)** and **output layers**.

The sparsity forces most **hn≈ 0**, the decoder uses only the active neurons to reconstruct data.

3.6.3 Covolutional Autoencoder (CAE)

CAEs are unsupervised neural networks that learn compact and informative representations of data by reconstructing the original input. They consist of an encoder that compresses the input into a low-dimensional feature map and a decoder that reconstructs the input by minimizing reconstruction error. Unlike traditional auto-encoders based on fully connected layers, CAEs use convolutional operations, which preserve spatial relationships between pixels, making them especially effective for image and video processing tasks [64].

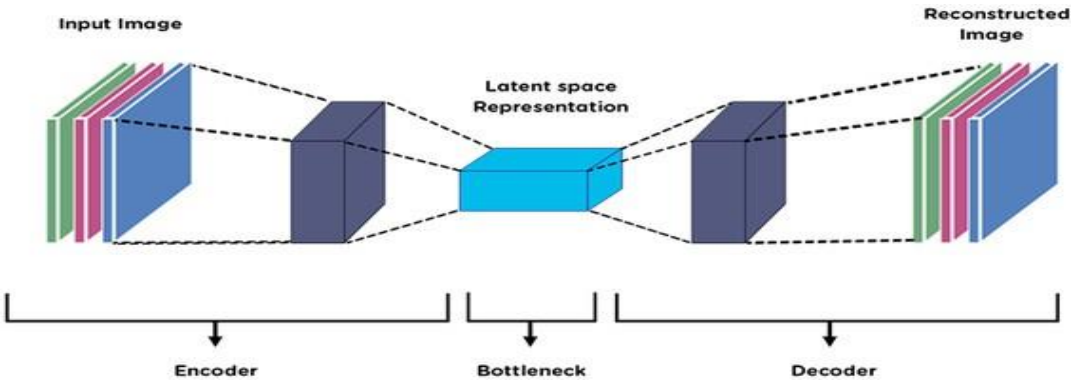


Figure 3-22: Convolutional Auto-Encoder Architecture.

A convolutional autoencoder (CAE) consists of an encoder and a decoder built with convolutional and transposed convolutional layers. The encoder extracts mid- and high-level features from input data using convolution, pooling, and ReLU activation, where pooling reduces dimensionality and the bottleneck layer captures the most important representations. The decoder then reconstructs the original input by progressively up sampling the encoded feature maps using transposed convolutions[65][66].

By preserving spatial relationships through convolutional operations, CAEs enable effective hierarchical feature learning, and fully connected layers may be included to support global feature representation and dimensional consistency [67].

A Convolutional Auto-Encoder is an unsupervised machine learning method that learns to reproduce its input, so the output is similar to the input. They compress the input into a lower-dimensional code and then reconstruct the output from this representation. the code is a compression of the input, also called the latent space.

It combines the strengths of autoencoders and CNNs. While a CNN is trained end-to-end for classification, a CAE is trained only to learn meaningful feature representations [62].

The CAE encoder transforms the input image into a compressed representation using convolutional and pooling layers. the feature maps of convolutional layer in encoder block is expressed as [66]:

$$F_k = f(X \times W_k + b_k) \tag{3.7}$$

F_k: the output feature map; ***W_k:*** the weight vector; ***X:*** the input data; ***b_k:*** the bias term and ***f***: the ReLU activation function

The CAE decoder: rebuilds the input image from the encoder's low-dimensional feature maps using deconvolution and up-sampling layers, and this is its formula:

$$X = g(F \times W^T + b) \tag{3.8}$$

3.6.4 Variational Autoencoder (VAE)

3.6.4.1 Definition

A **Variational Autoencoder (VAE)** is an unsupervised generative model and a probabilistic version of the standard autoencoder. Its main capability is generating data whose probability distribution is learned to be identical to the training dataset. The core idea is being able to generate data with some variations from a latent vector.

3.6.4.2 Mathematical Foundations

The generative process is expressed as:

$$p_\theta(x,z) = p_\theta(x|z)p_\theta(z) \tag{3.9}$$

Where $p\theta(x \mid z)$ is the generative model,

$p_\theta(z)$ is the prior distribution, and θ are the model parameters.

Training is done by minimizing the likelihood

$$p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz \tag{3.10}$$

But this integral is intractable, making the posterior distribution $p_\theta(z \mid x)$ impossible to estimate directly.

To solve this, VAE uses **variational inference** to approximate the true posterior $p_\theta(z \mid x)$ with a tractable one $q_\phi(z \mid x)$ by minimizing the Kullback-Leibler Divergence (KLD).

This leads to the maximization of the **variational lower bound** $L(\phi,\theta,x)$, which is an association of two terms:

$$L(\varphi, \theta, x) = KLD(q_{\varphi}(z|x)||p_{\theta}(z|x)[\log p_{\theta}(x|z)] \tag{3.11}$$

Regularization: $-KLD(q\phi(z|x)||p\theta(z))$, which acts as a regularizer by approximating the posterior $q\phi(z|x)$ to a prior distribution $p_{\theta}(z)$.

Reconstruction: $Eq\phi(z|x)[\log p\theta(x|z)]$, which represents the reconstruction error.

3.6.4.3 VAE Architecture

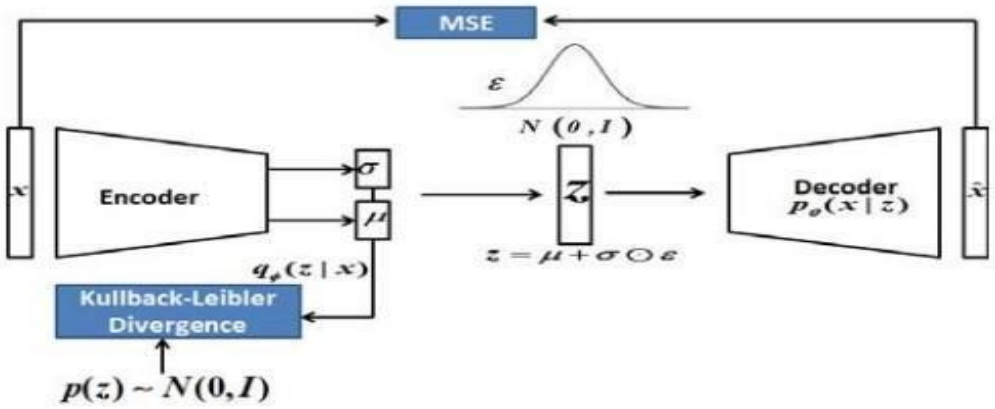


Figure 3-23: Variational Autoencoder architecture.

The VAE is modeled with an artificial neural network architecture comprising an encoder and a decoder.

Encoder (Inference network): Approximates the posterior distribution $q_{\phi}(z|x)$. It extracts the **mean (μ)** and **variance (σ)** vectors to define a Gaussian distribution from which the latent vector z is sampled.

$$q_{\varphi}(z|x) = N(z|\mu(x, \varphi), \sigma(x, \varphi)) \tag{3.12}$$

Input Processing:

Input: The encoder receives a high-dimensional input vector *x*,

Input Dimension: 513 points.

Forward Propagation through Hidden Layers

The input passes through deep neural network architecture:

Layer 1: 513 → 256 neurons (with activation function)

Layer 2: 256 → 128 neurons (with activation function)

The activation functions tested include Tanh, ReLU, or Sigmoid for these hidden layers.

Latent SpaceParameter Computation

Unlike a standard autoencoder that outputs a single latent vector directly, the VAE encoder outputs **parameters of a probability distribution** (3,12)

Specifically, the encoder computes:

Mean vector (μ): The center of the latent distribution

Variance vector (σ): The spread/uncertainty of the latent distribution

Decoder (Generator network): Takes the latent vector z as input and reconstructs the output vector $x^{\wedge}$. It has a structure similar to the encoder:

$$p_{\varphi}(x|z) = N(x|\mu(z,\theta), \sigma(z,\theta)) \tag{3.13}$$

Reparameterization Trick: Because sampling z directly from the Gaussian distribution makes backpropagation impossible, a reparameterization trick is performed. This involves sampling ϵ from $N(0,I)$ and computing $z=\mu+\sigma\odot\epsilon$[30].

Hyperparameters of the VAE

1-code size (size of bottleneck): decides how much the data has to be compressed.

2-**number of layers**: the depth of the encoder and decoder is the most important hyperparameter, while a higher depth increases complexity than a lower depth (faster to process).

3-**number of kernels**: which determine how much information that layer can represent.

More neurons: The layer can capture more details, making the reconstruction closer to the input.

Fewer neurons: Forces the network to compress information, which can help extract only the most important features but may lose some details.

4-**the activation function**: Choosing the right **activation function** for the bottleneck layer is essential for a high-performing autoencoder. Because these functions introduce nonlinearity, they dictate how well the network learns complex patterns.

3.7 Conclusion

This chapter presented the theoretical foundations required to understand the use of artificial intelligence, machine learning, and deep learning in medical image analysis. It first clarified the relationship between AI, ML, and DL, showing how deep learning extends classical machine learning by learning hierarchical representations directly from data, without relying on manual feature engineering. It also reviewed the main learning paradigms supervised, unsupervised, semi-supervised, and reinforcement learning and highlighted their respective roles in solving prediction and classification problems in different settings.

In addition, the chapter examined the principal models that support modern image-based diagnosis, with particular emphasis on convolutional neural networks and autoencoders. CNNs were presented as a fundamental architecture for image understanding because of their ability to automatically extract discriminative visual features through convolution, pooling, and fully connected layers. The chapter also discussed representative architectures such as AlexNet, VGG, and ResNet, as well as autoencoder variants including sparse autoencoders, convolutional autoencoders, and variational autoencoders, which are especially useful for representation learning, dimensionality reduction, reconstruction, and feature extraction.

Overall, this theoretical chapter provides the conceptual and technical basis for the methodological work developed in this Master thesis. By reviewing both classical machine learning methods and deep learning architectures, it establishes the foundation for selecting appropriate models for medical image classification and related tasks, while also underscoring the importance of architecture choice, activation functions, depth, and feature learning strategy in achieving reliable performance.

Chapter 4

Experimental Results and Discussions

4.1 Performance Analysis

The quantitative evaluation of deep learning models across medical imaging modalities reveals critical characteristics regarding representation learning, generalization, and structural suitability. This chapter systematically evaluates two distinct clinical studies: Study 1, a multiclass classification task segmenting brain lesions into four distinct pathological and physiological categories using a public Kaggle dataset 1; and Study 2, a binary classification task identifying the presence of pulmonary consolidation and pneumonia anomalies from chest X-rays.

To establish a comparative baseline, the overall diagnostic performance of pre-trained convolutional neural networks (CNNs), generative variational autoencoders (VAEs), and reconstructive convolutional autoencoders (CAEs) has been tabulated.

In study 1, VGG-19 and VGG-16 demonstrate superior classification performance, achieving accuracies of 94% and 93% respectively, with an outstanding Area Under the ROC Curve (AUC) of 98% (**Table 1**). This suggests that the hierarchical, small-kernel convolutions of VGG architectures are highly capable of preserving localized, high-contrast structural margins characteristic of neuro-oncological lesions[69]. Conversely, the parameter-efficient EfficientNet-B0 exhibits a substantial degradation in performance, collapsing to 63% accuracy. Unsupervised learning paradigms display diverging capabilities ; the CAE performs commendably at 79.06% accuracy, while the generative VAE struggles, yielding a near-random classification accuracy of 55%. This dichotomy highlights how regularized latent bottlenecks affect downstream medical classification tasks.

Table 4-1 : Comparative table that shows the performance of CNN and AE models in brain tumor classification.

CNN models	Accuracy	Precision	Recall	F1-Score	AUC	Confusion matrix
VGG16	93%	94%	93%	93%	98%	$\begin{bmatrix} 315 & 37 & 43 & 5 \\ 1 & 384 & 7 & 8 \\ 0 & 0 & 400 & 0 \\ 0 & 5 & 0 & 395 \end{bmatrix}$
VGG19	94%	95%	95%	94%	98%	$\begin{bmatrix} 323 & 44 & 33 & 0 \\ 3 & 394 & 0 & 3 \\ 0 & 0 & 400 & 0 \\ 1 & 4 & 0 & 395 \end{bmatrix}$
EfficientNetB0	63%	74%	63%	61%	87%	$\begin{bmatrix} 323 & 14 & 57 & 6 \\ 210 & 145 & 39 & 6 \\ 1 & 0 & 399 & 0 \\ 140 & 0 & 115 & 145 \end{bmatrix}$
AE models						
VAE	55%	55%	55%	54%	78%	$\begin{bmatrix} 183 & 97 & 56 & 64 \\ 64 & 142 & 128 & 66 \\ 30 & 48 & 281 & 41 \\ 22 & 47 & 60 & 271 \end{bmatrix}$
CAE	79.06%	79.55%	79.06%	77.91%	93.6%	$\begin{bmatrix} 256 & 74 & 57 & 13 \\ 32 & 220 & 85 & 63 \\ 0 & 0 & 400 & 0 \\ 4 & 7 & 0 & 389 \end{bmatrix}$

In Study 2, the performance profiles show unique characteristics. Inception and CAE architectures lead the binary pneumonia cohort with identical accuracies of 86.50%, yet they demonstrate completely inverted clinical priorities. Inception achieves an exceptional Recall (Sensitivity) of 97.50% but a modest Precision of 85.26%, whereas the CAE achieves a highly conservative Precision of 95.11% and a Recall of 85.02%.

Notably, both autoencoder models (CAE and VAE) outperform several heavily parameterized supervised transfer learning models, including VGG-16, ResNet-50, and

DenseNet-12. DenseNet-121 exhibits a severe performance bottleneck, falling to 43.11% accuracy, although it maintains a high Precision of 98.13%. This systemic diagnostic divergence indicates that deeply nested skip-connections and dense feature reuse can inadvertently overfit to the high-frequency structural elements of chest X-rays, such as bone interfaces, rather than clinical consolidations [70].

Table 4-2 : Comparative table that shows the performance of CNN and AE models in pneumonia detection.

CNN models	Accuracy	Precision	Recall	F1-Score	AUC	Confusion matrix
ResNet50	62.13%	98.99%	46.17%	62.97%	93%	[278 0] [639 2]
VGG16	69.85%	96.19%	59.12%	73.23%	91%	[263 1 [262 37
VGG19	74.53%	92.30%	69.26%	79.14%	87.8%	[241 3 [197 44
EfficientNetB0	82.15%	95.08%	78.98%	85.98%	94.7%	[252 2 [138 50
DenseNet121	43.11%	98.13%	24.64%	39.40%	90%	[275 3 [483 15
Inception	86.50%	85.26%	97.50%	90.97%	92%	[170 10 [16 62
AE models						
VAE	84.98%	94.51%	83.30%	88.55%	**94.8%**	[247 3 [107 53
CAE	86.50%	95.11%	85.02%	89.78%	94%	[250 2 [96 54

4.1.1 Performance Evaluation Metrics

To evaluate the performance of the proposed deep learning models, several standard metrics are used. Accuracy measures the overall correctness of predictions, while precision reflects the reliability of positive predictions. Recall evaluates the ability to correctly detect positive cases, which is particularly important in medical diagnosis. The F1-score provides a balance between precision and recall, especially in imbalanced datasets. In addition, the AUC-ROC measures the model's ability to distinguish between classes across different thresholds, while the confusion matrix provides a detailed breakdown of correct and incorrect predictions.

Table 4-3 : Descriptive table of Evaluation metrics.

Metric	Explanation
Precision	It is the percentage of predictions that are correct. It is mathematically defined as : $$Precision = \frac{correctlyclassifiedactualpositives}{everythingclassifiedaspositive} = \frac{TP}{TP+FP}$$
Accuracy	It is the percentage of total predictions that are correct, including both positive and negative classifications. It is mathematically defined as: $$Accuracy = \frac{correctclassification}{totalclassification} = \frac{TP+TN}{TP+TN+FP+FN}$$
Recall	It is the proportion of all actual positives that were classified correctly as positives, is also known as the **True Positive Rate**. it is mathematically defined as: $$Recall = \frac{correctlyclassifiedactualpositives}{allactualpositives} = \frac{}{TP}$$
F1-Score	It is a metric that combines precision and recall into a single value. It measures the balance between correctly identifying positive cases and minimizing missed or incorrect positive predictions. It is mathematically defined as :

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

Area under the ROC curve	It measures how well a model can separate positive and negative classes.
Confision matrix	It is a table that shows the number of correct and incorrect predictions made by a classification model.

The elements of confusion matrix

It is a simple table used to measure how well a classification model is performing. It compares the predictions made by the model with the actual results and shows where the model was right or wrong. This helps us understand where the model is making mistakes so you can improve it. It breaks down the predictions into four categories:

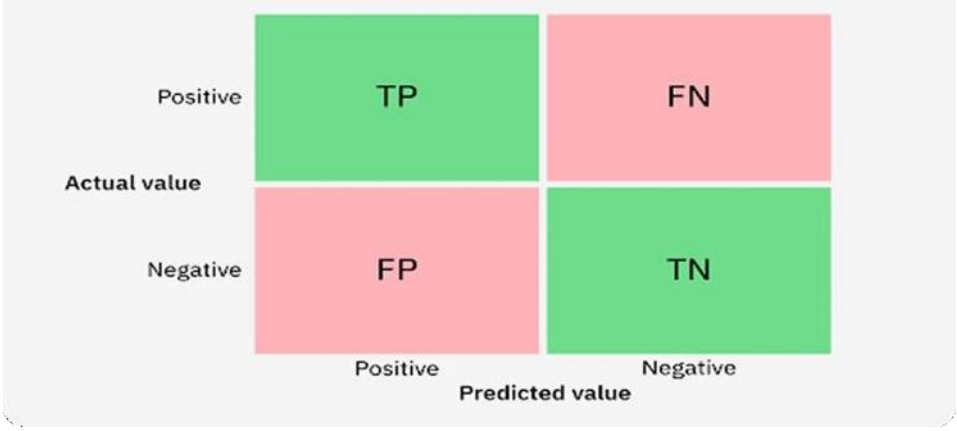


Figure 4-1 : Elements of Confusion matrix

1. **True Positive (TP):** The model predicts positive, and the actual class is positive.
2. **True Negative (TN):** The model predicts negative, and the actual class is negative.
3. **False Positive (FP):** The model predicts positive, but the actual class is negative.
4. **False Negative (FN):** The model predicts negative, but the actual class is positive.

4.1.2 Study 1: Brain Tumor Classification in MRI Images using CNNs, AEs, and Transfer Learning

A brain tumor is a mass or cluster of abnormal cells in the brain, which has the possibility of becoming life-threatening because of its ability to invade neighboring tissues and also form metastases. Brain tumor classification plays a crucial role in early diagnosis and treatment planning,

This project implements an automated brain tumor detection system using deep learning. Various models are built based on CNNs and AEs architectures (VGG16, VGG19, ResNet50, DenseNet, Inception, EfficientNet-B0, and VAE and CAE), to classify MRI brain scans into tumor and non-tumor categories.

Dataset 01

The dataset used consists of MRI images categorized into different classes: **Glioma Tumor, Meningioma Tumor, No Tumor, Pituitary Tumor.**

Glioma Tumor: A glioma is a type of tumor that starts in the glial cells of the brain or the spine. Gliomas comprise about 30 percent of all brain tumors and central nervous system tumors, and 80 percent of all malignant brain tumors.

Malignant Tumor: A malignant tumor contrasts with a non-cancerous benign tumor in that a malignancy is not self-limited in its growth, is capable of invading into adjacent tissues, and may be capable of spreading to distant tissues.

Pituitary Tumor: Pituitary adenomas are tumors that occur in the pituitary gland. Pituitary adenomas are generally divided into three categories dependent upon their biological functioning: benign adenoma, invasive adenoma, and carcinomas.

No Tumor: The Brain scan is normal. No Tumor is detected.

Here is the link of dataset:https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection

Here is the table of dataset distribution.

Table 4-4 : Distribution of images used in this study

Class	Training Images	Testing Images
Glioma	1400	400
Meningioma	1400	400
Pituitary	1400	400
No Tumor	1400	400
Total	**5600**	**1600**

4.1.2.1 Classification Errors, Class Confusions, and Pathological Overlap

Analyzing confusion matrices (**Tableau 4-1**) reveals that the "No Tumor" control cohort is perfectly classified by VGG-16, VGG-19, and the CAE (each achieving 400/400 True Negatives).Clinically, normal brain tissue is highly homogenous, lacking mass effect, dural tails, ventricular compression, or localized pathological vascularization. This homogeneity allows deep kernels to easily separate normal anatomy from abnormal structures.

Similarly, **Pituitary Tumors** are consistently classified with high accuracy (VGG-16:395/400, VGG-19: 395/400 and CAE: 389/400). Pituitary adenomas possess a distinct anatomical profile, consistently localizing to the sella turcica inferior to the optic chiasm. This anatomical consistency provides invariant spatial priors that the models exploit.

The most challenging classification boundary is between **Gliomas** and **Meningiomas**. Under VGG-19, 44 true Gliomas are misclassified as Meningiomas; under VGG-16, 37 Gliomas are misclassified as Meningiomas, and 43 are misclassified as No Tumor. This confusion stems directly from overlapping radiological features:

Infiltrative vs. Well-Circumscribed Margins: High-grade gliomas often present with dural contact or mimic the peripheral contrast-enhancement patterns of meningiomas [71].

Tissue Heterogeneity: Gliomas are highly heterogeneous malignant tumors with indistinct boundaries that merge into normal brain tissue, leading to confusion with normal scans (e.g., 43 VGG-16 misclassifications).

Mixed Sequences: The dataset blends T1-weighted, T2-weighted, and Fluid-Attenuated Inversion Recovery (FLAIR) sequences without explicit channel separation[3]. Gliomas appear hypointense on T1 and hyperintense on T2/FLAIR, while meningiomas display variable signal intensities depending on calcification levels. This sequence mixing creates substantial intra-class variance, making it difficult for the models to learn consistent feature mappings.

4.1.2.2 Discriminative Superiority of VGG Models and Transfer Learning

VGG architectures outperform the other models by utilizing small, stacked 3×3 convolutional kernels rather than larger ones (e.g., 11×11 or 5×5)[69].

Stacking two 3×3 convolutional layers achieves an effective receptive field of a 5×5 layer, while stacking three achieves a 7×7 field. This design provides two main advantages:

Incorporating multiple non-linear activation functions (ReLU) increases the network's capacity to learn complex, non-linear pathological boundaries [69].

It significantly reduces the number of parameters, lowering the risk of overfitting on relatively small medical image datasets [69].

Using pre-trained ImageNet weights enables the VGG models to leverage established spatial hierarchies [71]. The early layers act as robust edge and texture detectors, while the deeper, fine-tuned layers adapt to specific clinical features like tumor margins, necrosis, and vasogenic edema, accelerating convergence and improving generalization [72].

4.1.2.3 EfficientNetB0 Failure Mode

The performance collapse of EfficientNet-B0 to 63% accuracy is illustrated by its confusion matrix: it misclassifies 210 Meningiomas as Gliomas, and 140 Pituitary Tumors as Gliomas. EfficientNet's backbone relies heavily on **Depthwise Separable Convolutions** to reduce parameter density [5].While computationally efficient, depthwise separable convolutions decouple spatial filtering from channel-wise correlation learning [73].

In natural image classification (e.g., ImageNet), this separation is highly effective. However, in medical imaging, where pathological markers are defined by co-occurring spatial configurations and intensity variations across channels, decoupling these dimensions can lead to a loss of fine-grained texture representations [73].

Additionally, the compound scaling of EfficientNet-B0 dynamically scales depth, width, and resolution using fixed coefficients [74]. This scaling can compromise spatial feature map retention when processing grayscale medical inputs that differ significantly from the ImageNet target domain, resulting in high-level feature extraction failure.

4.1.2.4 Autoencoder Feature Space (VAE vs. CAE)

The CAE demonstrates a robust classification capability (79.06% accuracy) compared to the VAE (55% accuracy). In a CAE, representation learning is unregularized; the network is incentivized to minimize a reconstruction loss, such as Mean Squared Error (MSE), over spatial feature maps:

$$L_{CAE}(\theta, \varphi) = \frac{1}{N}\sum_{i=1}^{N} \left\| x_i - f_\theta(g_\varphi(x_i) \right\|^2 \tag{4.1}$$

This formulation forces the latent bottleneck to preserve high-frequency spatial structures, edge parameters, and localized density maps, which are essential for identifying distinct tumor boundaries. Conversely, the VAE regularizes its latent space by adding a Kullback-Leibler (KL) divergence term to force the latent distribution $\boldsymbol{q_\varphi(z\backslash x)}$ toward a standard multivariate Gaussian prior $\boldsymbol{p(z) \sim N(0,1)}$:

$$L_{VAE}(\theta, \varphi) = E_{q_\varphi(z|x)}[\log p_\theta(x|z)] \tag{4.2}$$

This KL regularization acts as an aggressive smoothing operator, penalizing complex, highly localized configurations in the latent space. Consequently, the generative decoder tends

to produce blurred reconstructions, and the classifier head receives flattened, low-frequency latent representations. This smoothing discard the fine-grained pathological variations needed to distinguish between Glioma, Meningioma, and normal tissue, as shown by the high error rates in the VAE confusion matrix (**Table 4-1**).

4.1.3 Study 2: Pneumonia Detection in Chest X-ray Images using CNNs, AEs, and Transfer Learning

Pneumonia is a severe respiratory infection affecting millions globally, leading to high mortality rates, particularly among young children and the elderly. Early diagnosis is crucial for effective treatment.

The main idea of this part of project was to build a various CNN and AE models for the image classification problem of pneumonia detection in chest X-ray images, then to perform a comparative study between different models. For this purpose, the CNN and AE deep learning model were trained and tested on the Chest X-ray Images dataset and transfer learning was used to speed up the training and further improve the performance of the model.

The problem includes detecting whether or not pneumonia shows up on X-ray images, i.e. binary classification problem, with the goal of providing additional help in medical diagnosis. To solve the problem, multiple different technologies and tools were used, mainly including Python and Keras.

Dataset 02

The Chest X-Ray Images (Pneumonia) dataset is utilized, comprising 5856 chest X-ray images, distributed in **two classes**:

- normal
- pneumonia

The train and test sets are pre-defined and contain 5216 and 624 images, respectively. The distribution of the images in both sets is imbalanced, with the majority of cases being pneumonia, as shown in **Figure a**. The imbalance is expected when it comes to medical data and quite common in practice as well.

Here is the link of dataset:https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

Here is a table that shows the distribution of dataset used in pneumonia detection:

Table 4-5 : Distribution of images used in this study

Class	Training	Testing	Validation	Total
Normal	1,224	278	81	1,583
Pneumonia	3,418	641	215	4,274
Total	**4,642**	**919**	**296**	**5,857**

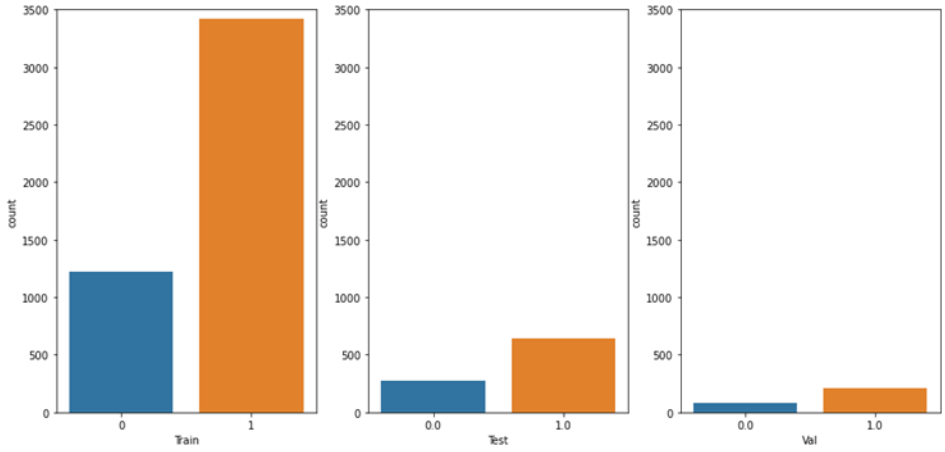


Figure 4-2 : Class distribution of train, test and validation sets in this study

4.1.3.1 Clinical Diagnosis Metrics

In medical radiodiagnostics, a model's clinical utility is determined by the balance between Sensitivity (Recall) and Specificity, rather than raw classification accuracy alone.

$$Sensitivity = \frac{TP}{TP + FN} \quad , Specificity = \frac{TN}{TN + FP} \tag{4.3}$$

For a clinical screening tool used to detect acute pulmonary infections, **False Negatives (FN)** present the highest risk. An FN represents a patient with active pneumonia who is

misclassified as normal and discharged without receiving antibiotics. This diagnostic failure can lead to severe clinical complications, sepsis, or death.

Conversely, **False Positives (FP)** result in healthy patients being flagged for pneumonia. While FPs incur secondary clinical costs, such as follow-up CT scans and unnecessary antibiotic administration (which can contribute to antimicrobial resistance), they are not immediately life-threatening.

Inception displays a clinical profile highly suited for screening, achieving an outstanding Sensitivity of 97.50% (FN = 16/641). However, this sensitivity comes at the cost of a high False Positive rate, resulting in a Specificity of only 61.15% (FP = 108/278).

In contrast, ResNet-50 and DenseNet-121 display clinically hazardous profiles. ResNet-50 achieves a Specificity of 98.92% (FP = 3), but its Sensitivity drops to 46.17% (FN = 345/641).[1] DenseNet-121 is even more conservative, with a Specificity of 98.92% (FP = 3) but a clinically unacceptable Sensitivity of 24.64% (FN = 483/641).

These models fail as screening tools because they miss the majority of active cases. This conservative behavior suggests that residual skip-connections and dense blocks can overfit to normal anatomical structures, causing the network to ignore low-contrast parenchymal infiltrates.

4.1.3.2 Pathological and Radiographic Challengies

The detection of pneumonia on chest X-rays is complicated by several factors such as:

Infiltration Overlap: Pneumonia manifests as areas of increased opacity (consolidation or interstitial infiltrates) that can be difficult to distinguish from pleural effusion, atelectasis, or pulmonary edema.

Inspiration Depth: Shallow inspiration can crowd the bronchovascular structures, mimicking pathological consolidations in the lower lung fields.

Geometrical and Scale Variations: Differences in patient positioning relative to the X-ray detector can alter the projected size and orientation of the lungs.

To address these variations, the data augmentation pipeline incorporates **Shear** (to simulate minor positioning skew) and **Zoom** (to simulate patient-to-detector distance

variations). These transformations help regularize the networks, training them to recognize pathologic consolidations independently of patient positioning.

4.1.3.3 Transfer Learning and Feature Extraction

Transfer learning leverages deep spatial hierarchies pre-trained on ImageNet[6].In pneumonia detection, models utilize frozen early layers for basic feature extraction, while deeper layers are fine-tuned to capture diffuse opacities and infiltrates [6]. This approach significantly improves training efficiency and performance on the relatively small chest X-ray dataset.

4.1.3.4 Autoencoder Performance in Binary Envirement

A notable finding is the high performance of the unsupervised architectures: the CAE and VAE achieve accuracies of 86.50% and 84.98%, respectively, outperforming VGG-16, ResNet-50, and DenseNet-121.

In a binary classification setting, the autoencoders' reconstructive pre-training acts as a strong regularizer. The global reconstruction objective forces the encoder to capture overall lung structures, pleural boundaries, and major density variations.

For the CAE, the reconstructed latent space is defined by a dense feature representation *z*:

$$z = \sigma(W * x + b) \tag{4.4}$$

This spatial representation preserves large-scale structural patterns. When a dense classification layer with a sigmoid activation is appended:

$$\hat{y} = sigmoid(W_c z + b) \tag{4.5}$$

The model utilizes these robust spatial features. Because the autoencoder's unsupervised phase does not rely on binary labels, it avoids overfitting to high-frequency background features, allowing it to generalize well to unseen cases.

4.2 Learning Curves Analysis

4.2.1 Training Dynamics and Convergence Profiles

Analyzing the training and validation learning curves (accuracy and loss) reveals the dynamics of convergence, generalization, and stability across the models.

VGG-16 and VGG-19: Show stable, monotonic convergence in both studies. The validation loss tracks closely with training loss, indicating robust generalization and a well-regularized optimization trajectory. This stability is supported by early stopping and dynamic learning rate decay (ReduceLROnPlateau), which adjust the learning rate by a factor of 0.1 upon validation loss stagnation.

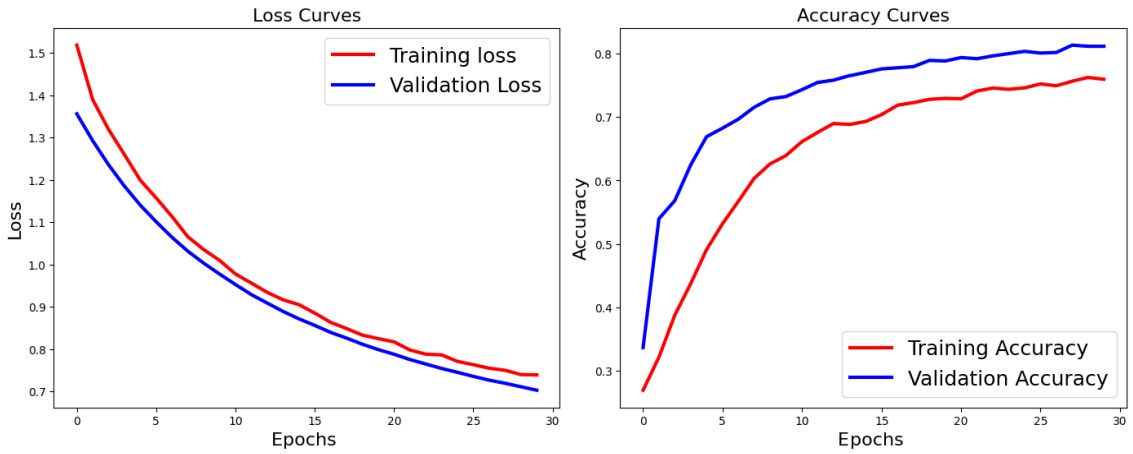


Figure 4-3 : Loss and Accuracy curves for VGG-16 models in brain tumor classification

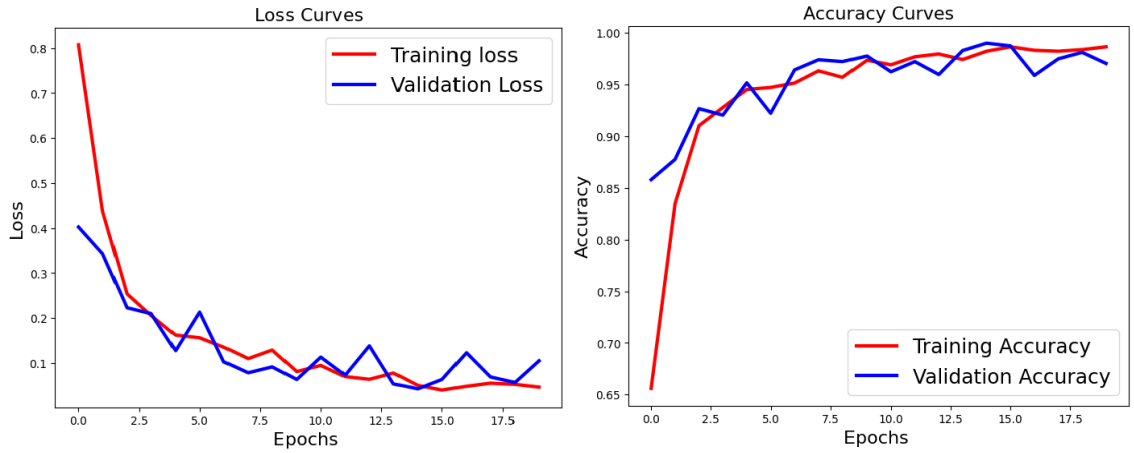


Figure 4-4 : Loss and Accuracy curves for VGG-19 in brain tumor classification

EfficientNet-B0: Displays unstable convergence, with highly fluctuating validation loss and accuracy curves. This volatility suggests that depthwise separable convolutions struggle to map grayscale medical structures consistently, leading to gradient variance during backpropagation.

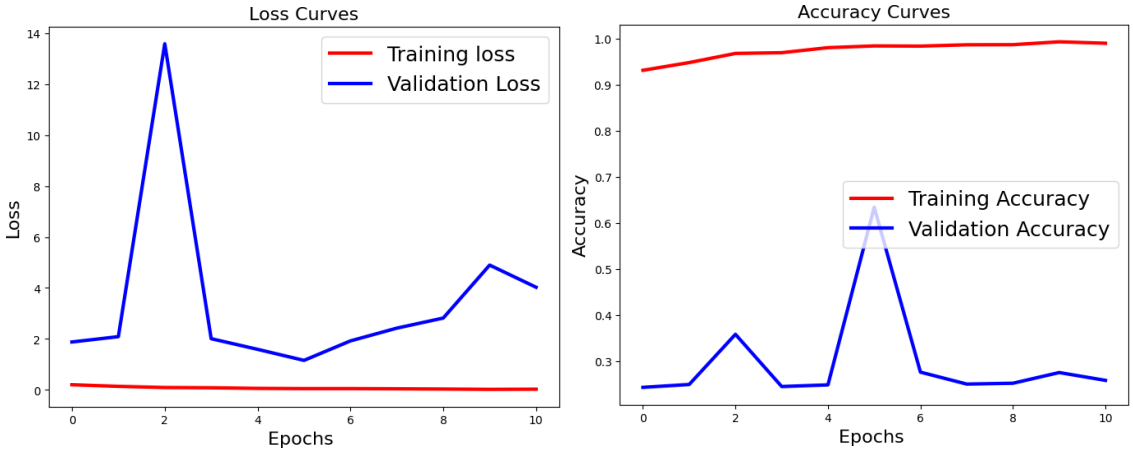


Figure 4-5 : Loss and Accuracy curves for EfficientNet-B0 in brain tumor classification

Inception: Demonstrates rapid convergence, reaching high validation accuracy early in training. The multi-scale inception blocks capture features at different resolutions, promoting stable convergence.



Figure 4-6 : Loss and Accuracy curves for InceptionV3 in pneumonia detection

ResNet-50 and DenseNet-121: Exhibit severe overfitting. Their training curves show rapid convergence, with training loss approaching zero and accuracy reaching nearly 100.00%. However, their validation losses diverge early in the training process, accompanied by fluctuating validation accuracies. This behavior indicates that deep, complex backbones can overfit to dataset-specific noise and high-frequency structures when trained on small datasets without extensive fine-tuning.

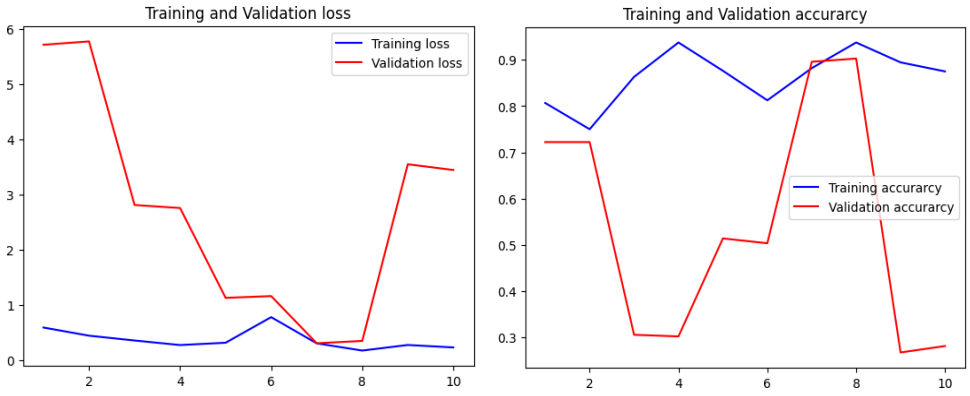


Figure 4-7 : Loss and Accuracy curves for ResNet50 in pneumonia detection

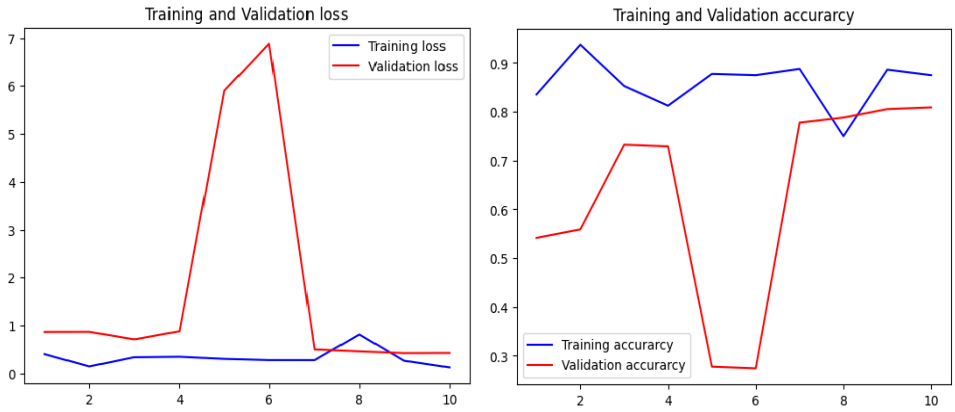


Figure 4-8 : Loss and Accuracy curves for DenseNet121 in pneumonia detection

CAE and VAE: Show stable reconstruction loss curves. The CAE converges smoothly, indicating that the reconstructive objective is well-suited for representation learning. The VAE displays slightly higher variance in its loss curves, reflecting the optimization trade-off between the reconstruction loss and the KL divergence penalty.

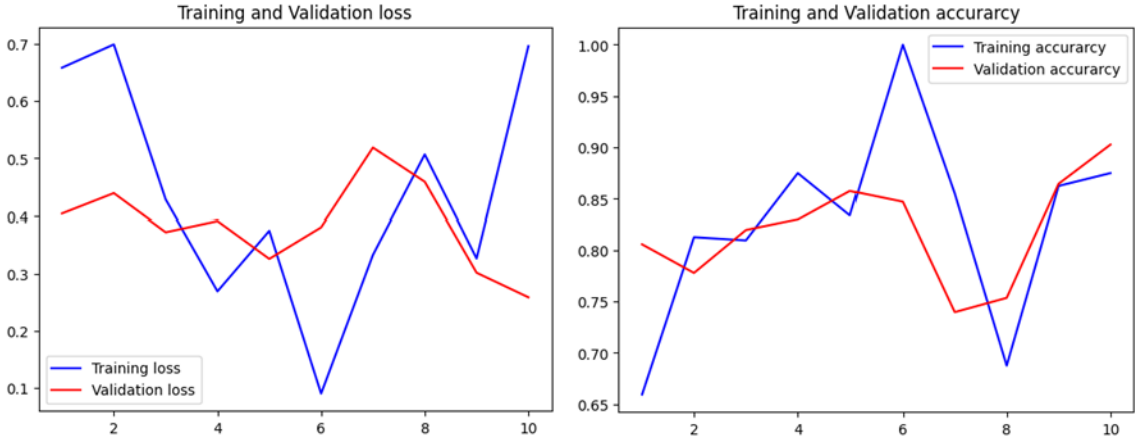


Figure 4-9 : Loss and Accuracy curves for VAE in Pneumonia detection



Figure 4-10 : Loss and Accuracy curves for CAE in pneumonia detection

4.2.2 Effectiveness of Data Augmentation and Fine-tunning

Applying data augmentation (rotation, translation, shear, and zoom) helps mitigate overfitting. These transformations introduce controlled spatial variance, preventing the networks from memorizing specific pixel configurations.

Fine-tuning deep layers of pre-trained models while keeping early layers frozen allows the networks to adapt to medical-specific features (such as lung consolidations or brain lesions) while preserving general spatial extractors, which improves generalization.

4.3 Comparative Analysis of CNNs, AEs, and Transfer Learning Approaches

Evaluating the relative merits of these deep learning paradigms requires comparing them across several key dimensions.

Table 4-6 : Comparative Assessment of Deep Learning Paradigms in Medical Image Classification.

Evaluation Dimension	Supervised Transfer Learning (VGG/Inception)	Unsupervised Autoencoders (CAE/VAE)	Custom CNNs (Built from Scratch)
Quantitative Performance	Superior in multiclass and high-complexity classification.	Excellent in binary tasks; moderate in multiclass settings.	Poor to moderate; limited by dataset size and depth.
Generalization	High; leverages pre-trained features.	High; learns general anatomical representations.	Low; prone to overfitting on small datasets.
Robustness to Noise	Sensitive to out-of-domain noise and modality shifts.	High; reconstruction loss regularizes latent space.	Highly sensitive to noise and artifacts.
Data Requirements	Low; pre-training reduces label volume needs.	Low to moderate; can utilize unlabeled images.	High; requires large labeled datasets for training.
Computational Complexity	High during fine-tuning; fast inference.	High reconstruction overhead; lightweight classification head.	Low to moderate; simple structural configurations.

Clinical Relevance	High sensitivity; excellent for screening.	Balanced metrics; robust to patient variations.	Low clinical utility due to generalization limits.

4.3.1 Quantitative Performance and Generalization

Supervised transfer learning models (such as VGG-19 and Inception) achieve high quantitative performance on complex datasets because they can leverage features learned from ImageNet [69].

This pre-training acts as a strong regularizer, allowing the networks to generalize well to unseen clinical cases. In contrast, custom CNNs trained from scratch are prone to overfitting on small datasets, as they must learn spatial representations without prior knowledge [74].

4.3.2 Robustness and Data Requirements

Unsupervised autoencoders offer a robust alternative when labeled data is limited. By optimizing for reconstruction rather than class labels, these networks learn to represent general anatomical structures.

This self-supervised learning paradigm is highly robust to noise and scan-to-scan variations, as the global reconstruction objective prevents the network from focusing on localized artifacts or class-specific noise. This allows the autoencoders to perform well on binary classification tasks even with limited supervision.

4.4 Clinical Relevance and Practical Implementations

Evaluating deep learning models for clinical deployment requires assessing their alignment with radiodiagnostic workflows and patient safety.

4.4.1 Computer-Aided Detection (CADe) vs. Computer-Aided Diagnosis (CADx)

These models serve distinct clinical functions:

CADe (Screening): High-sensitivity models like Inception (97.50% Recall) are well-suited for automated screening. Appended to a Picture Archiving and Communication System

(PACS), they can flag abnormal scans in high-volume environments (such as emergency departments), prioritizing them for radiologist review to reduce diagnostic delays.

CADx (Diagnostic Support): High-precision models like the CAE (95.11% Precision) are better suited for diagnostic confirmation and surgical planning. By minimizing False Positives, these networks provide reliable confirmation of pathology, reducing unnecessary interventions.

4.4.2 Multi-Parametric Integration and Real-World Neuro-Oncology

Deploying these models in clinical settings requires addressing a key limitation: the reliance on single-modality or mixed-sequence datasets. For clinical brain tumor diagnosis, radiologists do not rely on a single mixed sequence; they integrate multi-parametric MRI protocols, analyzing T1-weighted, T2-weighted, FLAIR, and contrast-enhanced scans in parallel to evaluate vascularity and margins [71]. Future clinical models must adopt multi-input, channel-stratified architectures to mimic this multi-parametric analysis before being integrated into diagnostic workflows.

4.5 Comparison with Existing Literature

Comparing these results with the literature highlights the impact of dataset design on reported model performance.

4.5.1 Multiclass Neuro-Oncology Classifications

In the literature, studies using the Sartaj or Figshare brain tumor datasets report accuracies ranging from 96% to 99% when using fine-tuned transfer learning models (such as VGG-16 or ResNet-50) [69]. Our study's VGG-19 and VGG-16 models achieve slightly lower, yet highly competitive, accuracies of 94% and 93%.

This minor performance difference is explained by dataset curation. Many early studies evaluated their models on datasets with significant image overlap, duplicate scans, or shared patient slices across the training and testing sets, resulting in data leakage. By using a cleaned, balanced dataset version where training and testing sets are strictly separated to prevent leakage, our evaluation provides a more realistic and clinically honest assessment of the models' generalization performance.

4.5.2 Unsupervised Benchmarks in Radiodignostics

The high performance of our CAE (86.50% accuracy) and VAE (84.98% accuracy) in binary pneumonia detection is consistent with recent research in self-supervised medical image analysis. While traditional literature focus heavily on fully supervised networks, recent studies demonstrate that self-supervised pre-training (using reconstruction or contrastive learning) yields features that are highly robust to scanner variations and domain shifts, often outperforming supervised models on out-of-distribution test sets.

4.6 Study Limitations

Despite the strong performance of some architectures, several methodological and algorithmic limitations affect their clinical generalizability.

4.6.1 Methodological and Dataset-Related Limitations

Single-Center Scans and Domain Bias: The datasets are sourced from public Kaggle repositories, which often aggregate scans from limited centers. This introduces domain bias, as the models adapt to specific scanner types, slice thicknesses, and acquisition protocols, which limits their performance on external clinical data.

Mixed-Sequence Confounding: The brain MRI dataset mixes T1-weighted, T2-weighted, and FLAIR sequences under a single set of labels [71]. This sequence mixing forces the models to learn features across completely different contrast mechanisms, which can lead to unstable feature representations.

Class Imbalance in Pneumonia Cohort: Although class weights were applied during training, the underlying class imbalance in the pneumonia dataset can still bias the models' decision thresholds, as shown by the high false negative rates in ResNet-50 and DenseNet-121.

4.6.2 Algorithmic and Statistical Limitations

Black-Box Nature and Explainability: The models lack integrated explainability tools, such as Grad-CAM or SHAP [75]. Without visual heatmaps to verify where the network is focusing, we cannot ensure that the classification is based on actual pathology rather than background artifacts, skull structures, or scanning boundaries.

Absence of Confidence Intervals: The performance metrics are reported as single scalar values without confidence intervals (such as bootstrapping or cross-validation standard deviations). This limits the ability to assess the statistical significance of the performance differences between architectures.

4.7 Conclusion

In this part some deep learning architectures were evaluated across multiclass neuro-oncological and binary thoracic radiodiagnostic tasks, revealing key insights for clinical applications.

In Study 1, fine-tuned transfer learning models (VGG-19 and VGG-16) demonstrated strong capabilities in multiclass tumor classification, achieving accuracies up to 94.00%. This performance indicates that simple, stacked convolutional layers are highly effective at preserving localized pathological details.

In Study 2, the high performance of unsupervised autoencoders (CAE: 86.50% accuracy; VAE: 84.98% accuracy) highlights the clinical potential of self-supervised reconstruction pre-training, which achieved robust binary classification without relying on large labeled datasets.

The evaluation also revealed notable performance trade-offs, such as Inception's high sensitivity (97.50% Recall) which is well-suited for automated screening, and the CAE's high precision (95.11%) which is better suited for diagnostic confirmation.

To move these systems closer to clinical deployment, future research must prioritize integrating explainability tools (e.g., Grad-CAM), validating models on multi-center datasets, and developing multi-parametric input architectures. Addressing these areas will help build trust and ensure these tools can be safely integrated into clinical radiodiognostic workflow.

Conclusion

The rapid advancement of Artificial Intelligence (AI) and Deep Learning technologies has significantly transformed the field of medical image analysis, providing powerful tools capable of assisting healthcare professionals in disease diagnosis and clinical decision-making. Medical imaging modalities such as chest X-rays and brain imaging generate vast amounts of visual information that require accurate interpretation. However, manual analysis of these images is often time-consuming and subject to inter-observer variability. Consequently, the development of intelligent computer-aided diagnostic systems has become an important research area aimed at improving diagnostic accuracy, efficiency, and reliability.

This Master's thesis has presented a systematic, comparative evaluation of supervised deep learning models and unsupervised autoencoder frameworks applied to diagnostic medical image analysis. By examining the distinct clinical tasks of multiclass brain tumor classification on MRI scans and binary pneumonia detection on chest X-rays, this research highlighted how architectural choices, latent space regularizations, and data properties influence automated diagnostic support systems. The investigation traced the evolution of computer-aided diagnosis from early handcrafted feature extraction pipelines to modern end-to-end deep representation networks, demonstrating that deep architectures can autonomously isolate complex pathological markers without manual intervention.

The experimental results presented in Chapter 4 reveal important insights regarding model selection and task suitability.

In Study 1 (multiclass brain tumor categorization), supervised small-kernel topologies demonstrated strong performance, with VGG-19 and VGG-16 achieving diagnostic accuracies of 94% and 93% respectively, backed by an outstanding AUC-ROC of **98 %**. This performance stems from the use of stacked ***3×3*** convolutions that expand the model's effective receptive field while minimizing parameter overhead and incorporating multiple non-linear activation functions. This architecture proved highly effective at distinguishing subtle structural margins in neuro-oncological lesions.

Conversely, compound-scaled models like **EfficientNet-B0** showed a notable performance drop, collapsing to **63%** accuracy. The confusion matrix analysis indicated a systemic failure in high-level feature extraction, driven by the model's reliance on depthwise

separable convolutions that decouple spatial filtering from channel-wise correlations, which filters out critical grayscale texture variations.

The unsupervised paradigms in Study 1 displayed a clear dichotomy: the Convolutional Autoencoder (CAE) achieved a reliable classification accuracy of **79.06 %**, whereas the Variational Autoencoder (VAE) fell to a near-random accuracy of **55 %**. This variance is directly explained by their mathematical foundations. The CAE optimization is completely unregularized, focusing strictly on minimizing spatial reconstruction errors via Mean Squared Error (MSE), which forces the bottleneck layer to preserve high-frequency structures and lesion boundaries. In contrast, the VAE incorporates a Kullback-Leibler (KL) divergence penalty to align its latent space with a standard Gaussian prior. This regularization acts as an aggressive smoothing operator, penalizing highly localized configurations and producing blurred reconstructions that discard the subtle variations needed to separate active gliomas from healthy tissue.

In Study 2 (binary pneumonia detection), the experimental outcomes shifted, with the **Inception** network and the CAE tying for the highest overall accuracy at **86.50%**. However, their underlying clinical profiles were completely inverted. The supervised Inception model achieved an exceptional Recall (Sensitivity) of **97.50%** but a moderate Precision of **85.26 %**, making it highly effective as a screening tool where minimizing missed cases (False Negatives) is the priority.

Conversely, the CAE model demonstrated a highly conservative Precision of **95.11 %** balanced by a Recall of **85.02%**, positioning it as a precise diagnostic verification tool that minimizes false alarms (False Positives). Notably, both autoencoders outperformed several heavily parameterized supervised networks, such as **ResNet-50** (**62.13%** accuracy) and **DenseNet-121** (**43.11%** accuracy). This performance bottleneck suggests that deeply nested skip connections and aggressive feature reuse can cause models to overfit to high-frequency structural elements like bone interfaces on chest X-rays, rather than identifying soft-tissue pulmonary consolidations.

A major strength of this study lies in its commitment to strict, discipline-specific validation protocols. As shown in Chapter 2, medical imaging datasets exhibit strong spatial correlations across adjacent slices from the same subject. Splitting data at the slice level introduces data leakage and identity confounding, where models memorize patient-specific

anatomical signatures rather than learning generalizable pathology, leading to artificially inflated performance metrics. This research implemented strict patient-level splitting, ensuring that all derived slices and augmented variants from a single subject were isolated within a single partition, providing a reliable estimate of clinical performance.

The study has several limitations that must be considered. The datasets were obtained from public Kaggle repositories, which may introduce domain bias related to scanner type, acquisition protocol, and source population. The brain MRI dataset mixes T1-weighted, T2-weighted, and FLAIR sequences under shared labels, which may create sequence-related confounding. The pneumonia dataset presents class imbalance, which can affect decision thresholds even when class weights are applied. In addition, the models were evaluated using scalar metrics without confidence intervals, and explainability methods such as Grad-CAM or SHAP were not integrated into the experimental workflow. These limitations restrict direct clinical generalizability and indicate that further validation is necessary.

Additionally, the chest X-ray cohort exhibits significant data imbalance, reflecting real-world clinical data distribution but inherently complicating model optimization under standard cross-entropy formulations.

Future work should focus on several key optimization pathways to build upon these findings:

1) **Multi-Channel Sequence Separation**: Designing multi-input network paths that explicitly separate T1, T2, and FLAIR modalities to resolve boundary confusion between infiltrative gliomas and well-circumscribed meningiomas.
2) **Advanced Loss Formulations**: Integrating regional overlap losses, such as Dice Loss, combined with confidence-weighted Focal Loss terms to prevent background pixels from dominating gradient updates during training on imbalanced cohorts.
3) **External Multicenter Validation**: Testing the trained models on data from external clinics to evaluate generalizability across different scanner manufacturers and protocol variations.
4) **High-Fidelity Explainability**: Advancing beyond coarse Grad-CAM visual heatmaps toward high-resolution pixel-level explainability tools, giving clinicians clear insight into the exact features driving model recommendations.

In conclusion, this thesis provides a coherent study of deep learning for diagnostic medical image classification, moving from physical imaging foundations to theoretical AI concepts and finally to experimental evaluation. The results show that while supervised frameworks like VGG achieve high performance on well-structured multiclass classification tasks, unsupervised convolutional autoencoders offer a robust, parameter-efficient alternative for binary detection pipelines. By balancing precision and sensitivity, these models show great promise as reliable, automated "second opinions" that can seamlessly integrate into modern clinical workflows to reduce diagnostic errors and support evidence-based healthcare decision-making.

References

[1] S. G. Armato III, G. McLennan, L. Bidaut, et al., “The lung image database consortium (LIDC) and image database resource initiative (IDRI): A completed reference database of lung nodules on CT scans,” Med. Phys., vol. 38, no. 2, pp. 915–931, Feb. 2011.

[2] S. H. Kim, “Basic Physical Principles and Clinical Applications of Computed Tomography,” Prog. Med. Phys., vol. 32, no. 1, pp. 1–15, 2021.

[3] W. Ahmed and M. Hammad, “Fundamentals of Medical Image Analysis,” in AI for Medical Image Analysis, N. Ben Aoun, S. Ahmad, and M. Hammad, Eds. Cham, Switzerland: Springer, 2026, ch. 2, doi: 10.1007/978-3-032-02963-8_2.

[4] “Radon Transform,” Wikipedia.

[5] G. Jerusalem, R. Hustinx, Y. Beguin, and G. Fillet, “PET scan imaging in oncology,” Eur. J. Cancer, vol. 39, no. 11, pp. 1525–1534, 2003.

[6] P. M. Parizel and M. A. Smits, “The Physics of MRI,” in Clinical Emergency Radiology. Cambridge, U.K.: Cambridge Univ. Press.

[7] MRI Master, “TR and TE in MRI | TR (Repetition Time), TE (Echo Time) and Image Contrast.”

[8] University of Minnesota, “The Bloch Equations.”

[9] UCSF Larson Advanced Imaging Group, “Bloch Equation — Principles of MRI.”

[10] A. J. Scherzinger et al., “The clinician-AI interface: Intended use and explainability in FDA-cleared AI devices for medical image interpretation,” NPJ Digit. Med., 2024.

[11] AnonJC, “Image Classification Technology in Medical Imaging: From CAD to Multimodal Foundation Models,” Medium, 2024.

[12] Y. Li et al., “UNet and Variants for Medical Image Segmentation,” Int. J. Netw. Dyn. Intell., 2025.

[13] Y. Zhang et al., “MetaSwin: A unified meta vision transformer model for medical image segmentation,” 2024.

[14] M. Hatamizadeh et al., “Swin Transformers for Semantic Segmentation of Brain Tumors in MRI Images,” arXiv:2201.01266, 2022.

[15] D. A. Taha and A. Hanbury, “Towards a guideline for evaluation metrics in medical image segmentation,” BMC Med. Imaging, vol. 22, no. 1, 2022.

[16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal Loss for Dense Object Detection,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2017, pp. 2980–2988.

[17] X. Zhang et al., “Partial Decoder Attention Network with Contour Weighted Loss Function for Data-Imbalance Medical Image Segmentation,” arXiv:2601.14338, 2026.

[18] X. Wang et al., “3D-MRI Brain Glioma Intelligent Segmentation Based on Improved 3D U-Net Network,” PLoS One, vol. 20, no. 6, 2025.

[19] VeraDP, “Dice Coefficient vs. IoU in Medical Image Segmentation.”

[20] N. Huynh, “Understanding Evaluation Metrics in Medical Image Segmentation,” Medium, 2023.

[21] R. Alzubaidi et al., “Effect of Data Leakage in Brain MRI Classification Using 2D Convolutional Neural Networks,” Sci. Rep., vol. 11, 2021

[22] A. Smith et al., “Data Leakage in Deep Learning for Alzheimer's Disease Diagnosis: A Scoping Review of Methodological Rigor and Performance Inflation,” Diagnostics, vol. 15, no. 18, 2025.

[23] M. Roy et al., “Data Leakage in Deep Learning Studies of Translational EEG,” Front. Neurosci., 2024.

[24] Rigaku, “How Does CT Reconstruction Work?”

[25] R. K. Knipe and M. Sharma, “Magnetic Resonance Imaging Physics,” StatPearls Publishing, 2025.

[26] E. Mark Haacke, “Bloch Equations,” Questions and Answers in MRI. [27] M. A. Khan et al., “Machine Learning and Deep Learning Approach for Medical Image Analysis: Diagnosis to Detection,” Diagnostics, vol. 12, no. 12, 2022.

[28] H. Zhang et al., “Enhancing Lung Cancer Classification Through a Double Attention Hybrid CNN-HiFuse Approach,” 2025.

[29] M. H. Ali et al., “Grad-CAM-Based Explainable Artificial Intelligence Related to Medical Text Processing,” Healthcare, vol. 11, no. 19, 2023.

[30] A. Sharma et al., “Advancing AI Interpretability in Medical Imaging: A Comparative Analysis of Pixel-Level Interpretability and Grad-CAM Models,” AI, vol. 7, no. 1, 2025.

[31] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in Med. Image Comput. Comput.-Assist. Intervent. (MICCAI), 2015, pp. 234–241.

[32] Y. Li et al., “RTAS-Net: A ResNet-Transformer-ASPP Semantic Segmentation Network for Remote Sensing Images,” PLoS One, 2026.

[33] Y. Li et al., “Patient-Level Classification of Rotator Cuff Tears on Shoulder MRI Using an Explainable Vision Transformer Framework,” 2025.

[34] A. Kumar et al., “Lite Swin UNETR: A Lightweight Version of Swin UNETR for Efficient 3D Medical Image Segmentation,” 2025.

[35] M. A. Kohli et al., “Rethinking Privacy in Medical Imaging AI: From Metadata and Pixel-Level Identification Risks to Federated Learning and Synthetic Data Challenges,” Radiol. Artif. Intell., 2025.

[36] DCM Services, “What is De-Identification, and Why is it Essential for Safeguarding Healthcare Imaging Data?”

[37] Early Detection Research Network, “EDRN DICOM De-identification Process.”

[38] C. M. Bishop, Pattern Recognition and Machine Learning, vol. 1. New York, NY, USA: Springer,2006.

[39] A. Coates, "Demystifying Unsupervised Feature Learning," Ph.D. dissertation, Stanford University, Stanford, CA, USA, 2012.

[40] :J.Zou, Y.han , and S.-S. So ,” Overview of artificial neural networks,” in Artificial Neural Networks : Methods and Applications , pp.14-22 , 2009.

[41] .Algorithms that Learn: Foundations of AI and Machine Learning. Chyren Publication, 2025.

[42]N. K. Mishra, P. Singh, A. Gupta, et al., "PP-CNN: probabilistic pooling CNN for enhanced image classification," Neural Comput&Applic, vol. 37, pp. 4345–4361, 2025, doi: 10.1007/s00521-024-10862-3.

[43] M. M. Taye, “Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions,” Computation, vol. 11, no. 3, p. 52, 2023, doi: 10.3390/computation11030052.

[44] K. O’Shea and R. Nash, “An introduction to convolutional neural networks,” *arXiv preprint arXiv:1511.08458 [cs.NE]*, Dec. 2015. doi: 10.48550/arXiv.1511.08458.

[45] Phung, V. and Rhee, E., “A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets,” *Applied Sciences*, vol. 9, no. 21, p. 4500, Oct. 2019, doi: 10.3390/app9214500.

[46] Aakanksha, G. Singh, J. Singh, and D. Verma, "A Systematic and Bibliometric Review on Face Recognition: Convolutional Neural Network," in *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)*, 2022, pp. 791–796, doi: 10.1109/ICACRS55517.2022.10029115.

[47] E. Pap, Ed., *Artificial Intelligence: Theory and Applications*, Studies in Computational Intelligence, vol. 973. Cham, Switzerland: Springer, 2021, ISBN: 978-3-030-72710-9, doi: 10.1007/978-3-030-72711-6.

[48] Zhao, X., L. Wang, Y. Zhang, *et al.*, "A review of convolutional neural networks in computer vision," *Artificial Intelligence Review*, vol. 57, Art. no. 99, 2024, doi: 10.1007/s10462-024-10721-6

[49] Yamashita, R., M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," Insights into Imaging, vol. 9, no. 4, pp. 611–629, Aug. 2018, doi: 10.1007/s13244-018-0639-9.

[50] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat, "CNN variants for computer vision: History, architecture, application, challenges and future scope," *Electronics*, vol. 10, no. 20, p. 2470, 2021, doi: 10.3390/electronics10202470.

[51] Purwono, I., A. Ma'arif, W. Rahmaniar, H. Imam, H. I. K. Fathurrahman, A. Frisky, and Q. M. U.Haq, "Understanding of convolutional neural network (CNN): A review," *International Journal of Robotics and Control Systems*, vol. 2, no. 4, pp. 739–748, Jan. 2023, doi: 10.31763/ijrcs.v2i4.888.

[52] Y. Wang, Y. Li, Y. Song, and X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition," *Applied Sciences*, vol. 10, no. 5, p. 1897, 2020, doi: 10.3390/app10051897.

[53] Z. A. Haq and Z. A. Jaffery, "Impact of activation functions and number of layers on the classification of fruits using CNN," in *Proc. 8th Int. Conf. Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 2021, pp. 227–231.

[54] J.Teuwen and N. Moriakov, "Convolutional neural networks," in *Handbook of Medical Image Computing and Computer Assisted Intervention*, S. K. Zhou, D. Rueckert, and G. Fichtinger, Eds.

Elsevier and MICCAI Society Book Series. Academic Press, 2020, pp. 481–501, ISBN: 978-0-12-816176-0, doi: 10.1016/B978-0-12-816176-0.00020-6.

[55] D. Sarwinda, R. H. Paradisa, A. Bustamam, and P. Anggia, "Deep learning in image classification using residual network (ResNet) variants for detection of colorectal cancer," *Procedia Computer Science*, vol. 179, pp. 423–431, 2021, doi: 10.1016/j.procs.2021.01.025.

[56]J. Jain, M. Bora, S. Chandnani, S. Grover, and S. Sadwal, "Comparison of VGG-16, VGG-19, and ResNet-101 CNN Models for the Purpose of Suspicious Activity Detection," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 121–130, 2023, doi: 10.32628/CSEIT2390124.

[57]A. T. Adeleke, U. Iwok, E. Ogungbemi, and K. Udofia, "Comparing the Performance of MobileNet and AlexNet Deep Learning-Based Models for Facial Recognition," *Journal of Engineering Research and Reports*, vol. 27, pp. 110–134, 2025, doi: 10.9734/jerr/2025/v27i121730.

[58] *Advances in Computer Communication and Computational Sciences: Proceedings of IC4S 2017, Volume 1*, Springer Nature Singapore, Allemagne, 2018.

[59] P. Torío, M. G. Sánchez, Í. Cuiñas, *et al.*, "Unsupervised detection of offending radio transmissions by means of a deep learning autoencoder," **J. Wireless Com Network**, vol. 2025, no. 104, 2025, doi: 10.1186/s13638-025-02496-3.

[60] P. Riti, *Autoencoders*, in *Introduction to Generative AI with Julia and Python*. Berkeley, CA, USA: Apress, 2026. doi: 10.1007/979-8-8688-2329-9_6

[61]**I. D. Mienye and T. G. Swart**, "Deep autoencoder neural networks: A comprehensive review and new perspectives," *Archives of Computational Methods in Engineering*, vol. 32, pp. 3981–4000, 2025, doi: 10.1007/s11831-025-10260-5.

[62] K. Berahmand, F. Daneshfar, E. S. Salehi, *et al.*, "Autoencoders and their applications in machine learning: A survey," **Artificial Intelligence Review**, vol. 57, no. 28, 2024, doi: 10.100/s10462-023-10662-6.

[63] N. Rathour, R. Singh, A. Gehlot, S. V. Akram, A. K. Thakur, and A. Kumar, “The decadal perspective of facial emotion processing and recognition: A survey,” *Displays*, vol. 75, Art. no. 102330, 2022, doi: 10.1016/j.displa.2022.102330.

[64]Q. Ning, Q. Dong, H. Huang, H. Wang, S. Zhao, X. Hu, Q. Li, W. Zhang, Y. Liu, M. He, B. Ge, L. Zhao, Z. Wu, L. Zhang, S. Xu, D. Zhu, X. Jiang, and T. Liu, "Chapter 14 - Deep learning in functional brain mapping and associated applications," in *Deep Learning for Medical Image Analysis (Second Edition)*, S. K. Zhou, H. Greenspan, and D. Shen, Eds. Academic Press, 2024, pp. 395–423. doi: 10.1016/B978-0-32-385124-4.00025-8.

[65] B. Nageswara Rao, U. R. Acharya, R. S. Tan, *et al.*, “Convolutional autoencoder-based deep learning for intracerebral hemorrhage classification using brain CT images,” **Cogn. Neurodyn.**, vol. 19, no. 77, 2025, doi: 10.1007/s11571-025-10259-5.

[66]P.Upretee and M. E. Yüksel, "Chapter 13 - Accurate classification of heart sounds for disease diagnosis by using spectral analysis and deep learning methods," in *Data Analytics in Biomedical Engineering and Healthcare*, K. C. Lee, S. S. Roy, P. Samui, and V. Kumar, Eds. Academic Press, 2021, pp. 215–232. doi: 10.1016/B978-0-12-819314-3.00014-8.

[67]N. Mansouri and Z. Lachiri, "Laughter synthesis: A comparison between Variational autoencoder and Autoencoder," *2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, Sousse, Tunisia, 2020, pp. 1-6, doi: 10.1109/ATSIP49331.2020.9231607

[68] M. M. Rahman, M. R. Islam, and M. A. Hossain, “Brain tumor detection from images and comparison with transfer learning methods and 3-layer CNN,” PMC, 2024.

[69] M. S. Rahman, M. A. Hossain, and M. T. Islam, “Auto encoder-guided Feature Extraction for Pneumonia Identification from Chest X-ray Images,” E3S Web of Conferences, 2024.

[70] M. Ş. Tiryaki and A. Yılmaz, “Deep Learning Based Brain Tumor Diagnosis,” DergiPark, 2024.

[71] “Classification of Brain Tumor Using Hybrid Model of Cosine Annealing and Weighted Snapshot Ensembles,” Science World Journal, 2024.

[72] M. A. Alghamdi et al., “Enhancing Brain Tumor Detection from MRI-Based Images Through Deep Transfer Learning Models,” AI, vol. 6, no. 12, 2025.

[73] Y. Zhang et al., “Lightweight U-Net with Depthwise Separable Convolution for Skin Lesion Segmentation,” PMC, 2025.

[74] M. H. Rahman et al., “Bridging the Gap: Exploring Interpretability in Deep Learning Models for Brain Tumor Classification from MRI Images,” Preprints, 2024.

Appendix A

[1] GeeksforGeeks, "VGG-16 | CNN model," *GeeksforGeeks*, May 2026.

[2] K. Haider, "VGG16: The Timeless Deep Learning Architecture That Redefined Image Classification," *Medium - Deep Vision*, Nov. 2025.

[3] S. P. Bijoy, "VGG16 vs VGG19: A Detailed Comparison of the Popular CNN Architectures," *Medium*, Mar. 2025.

[4] Ultralytics, "What is ResNet-50 and what is its relevance in computer vision?," *Ultralytics Blog*, May 2025.

[5] S. Herath, "EfficientNetB0 architecture — Stem Layer," *Medium - Image Processing with Python*, Jan. 2024.

[6] GeeksforGeeks, "Efficientnet Architecture," *GeeksforGeeks*, Jun. 2024.

[7] GeeksforGeeks, "DenseNet Explained," *GeeksforGeeks*, Feb. 2025.

[8] ResearchGate, "Specific parameter configuration of DenseNet121," *ResearchGate - UAV-based tree species classification*, May 2026..

[9] GeeksforGeeks, "Inception V2 and V3 - Inception Network Versions," *GeeksforGeeks*, Oct. 2022.
[10] ResearchGate, "The basic architecture of Inception-v3," *ResearchGate*, Apr. 2018.

Appendix B

[1] GeeksforGeeks, "Variational AutoEncoders," *GeeksforGeeks*, Dec. 16, 2025.

[2] IBM Think, "What is a Variational Autoencoder?", *IBM*, 2026.

[3] Liora.io, "Variational Autoencoder : tout savoir sur ce modèle génératif," *Liora.io*, 2025.

[4] Springer, "Autoencoders and Deep Representation Learning," in *Deep Learning for Medical Image Analysis*, Springer, 2025.

[5] ScienceDirect, "Convolutional Autoencoders," *Computer Science Topics*, Elsevier, 2025.

[6] GeeksforGeeks, "Implement Convolutional Autoencoder in PyTorch with CUDA," *GeeksforGeeks*, Aug. 5, 2025.

[7] ResearchGate, "Convolutional autoencoder (CAE) structure," *ResearchGate*, Art. 352518787, 2021.

[8] DigitalOcean, "How Convolutional Autoencoders Power Deep Learning Applications," *DigitalOcean Community Tutorials*, Apr. 27, 2025.

[9] H. Wijanarko, E. Calista, L.-F. Chen, and Y.-S. Chen, "Tri-VAE: Triplet Variational Autoencoder for Unsupervised Anomaly Detection in Brain Tumor MRI," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024, pp. 3930-3939.

[10] B. Ahmad, J. Sun, Q. You, V. Palade, and Z. Mao, "Brain Tumor Classification Using a Combination of Variational Autoencoders and Generative Adversarial Networks," *Biomedicines*, vol. 10, no. 2, p. 223, 2022.

[11] P. Naga Srinivasu, T. B. Krishna, S. Ahmed, N. Almusallam, F. K. Alarfaj, and N. Allheeib, "Variational Autoencoders- Based Self- Learning Model for Tumor Identification and Impact Analysis from 2- D MRI Images," *Journal of Healthcare Engineering*, vol. 2023, Art. ID 1566123, 2023.

[12] P. Strenge et al., "Enhancing brain tumor detection using optical coherence tomography and variational autoencoders," in *Medical Imaging 2025: Clinical and Biomedical Imaging*, SPIE, vol. 13410, p. 62, 2025.

[13] Premier Science, "Multi-Class Brain Tumor Type Classification Using Convolutional Autoencoder (CAE) With CNN-VGG19: An Experimental Study," *Premier Science*, 2025.

Appendix A
Convolutional Neural Networks

A.1 Fundamentals of Convolutional Neural Networks

CNN Architecture Overview

A Convolutional Neural Network (CNN) is organized as a sequence of processing stages that transform raw image data into high-level feature representations for classification. The architecture begins with an input layer, followed by alternating convolution and pooling layers that progressively reduce spatial dimensions while increasing feature depth. The resulting feature maps are flattened and passed to fully connected layers that produce final class predictions [1][2].

Convolution Layers

these are responsible for extracting spatial features from input images using learnable filters. They filters slide across the image to generate feature maps that capture relevant patterns. Early layers detect low-level features such as edges and orientations, whereas deeper layers learn more complex structures and object-specific characteristics. Stacking multiple convolutional layers enables efficient feature extraction with fewer parameters while increasing the network's representational capacity [1][2].

Pooling Layers

Pooling layers reduce the spatial dimensions of feature maps, decreasing computational cost and helping prevent overfitting. The most commonly used technique is max pooling, which retains the strongest activation within a local region while discarding less important information. In addition to reducing memory requirements, pooling improves the network's robustness to small positional variations in the input image [1][2][4].

Activation Functions

Activation functions introduce non-linearity into CNNs, allowing them to learn complex patterns. The Rectified Linear Unit (ReLU) is widely used because it accelerates training and mitigates the vanishing gradient problem [2]. More recent functions, such as

Swish, provide smoother gradient propagation by allowing small negative values to pass through the network, improving learning stability in deep architectures [3][5].

Fully Connected Layers

Fully connected layers perform the final classification task by combining all extracted features into a global representation. Before entering these layers, feature maps are flattened into a one-dimensional vector. The final layer typically uses a Softmax function to convert output scores into class probabilities, ensuring that predictions form a valid probability distribution [1][2][3].

Feature Extraction Process

CNNs learn features hierarchically, progressing from simple to complex representations. Initial layers capture basic visual patterns such as edges and textures, intermediate layers identify shapes and contours, and deeper layers encode high-level semantic information. This hierarchical feature extraction enables CNNs to effectively represent both local image details and global structural patterns, making them highly effective for image analysis tasks [2][4].

A.2 VGG Architectures

A.2.1 VGG16

Architecture Description

The VGG16 architecture is a landmark convolutional neural network designed by the Visual Geometry Group at the University of Oxford and introduced by Karen Simonyan and Andrew Zisserman at the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The model is defined by its deep, uniform structure containing a total of 16 layers with learnable weights, which specifically comprises 13 convolutional layers and 3 fully connected layers [1], [2]. VGG16 expects a fixed input tensor size of **224×224** pixels with three color channels, forming a standard RGB input.

The architectural flow is organized into five distinct convolutional blocks . Where each block contains a sequence of convolutional layers followed by a max-pooling layer for downsampling [2]. After the final convolutional block, the generated **7×7×512** feature map is flattened into a single-dimensional vector of **25,088** elements. This vector is passed through

two consecutive fully connected layers containing **4,096** neurons each, before terminating at a final dense layer that utilizes a Softmax function to output classification scores across the target categories [1].



Figure A.1: VGG-16 model architecture

Main Characteristics

The key innovation of VGG16 is its simple and uniform architecture, which relies exclusively on stacked 3×3 convolutional filters with a stride of 1 and padding of 1. Spatial dimensions are preserved within convolutional layers and reduced only through 2×2 max-pooling operations. As the network depth increases, the number of feature channels progressively expands from 64 to 512, creating a hierarchical structure that gradually replaces spatial resolution with richer feature representations [1][2].

Advantages and Limitations

One of the main strengths of VGG16 is its simplicity and effectiveness. By using multiple stacked 3×3 filters, the network achieves a large receptive field while introducing additional non-linear transformations, enabling the extraction of complex visual features. Its strong performance on the ImageNet dataset demonstrated the importance of network depth for image recognition tasks [2].

Despite its effectiveness, VGG16 has a very large computational cost, containing approximately 138 million trainable parameters. The model requires substantial memory, storage, and processing resources, with most parameters concentrated in the fully connected layers. As a result, training is slower and less efficient compared to more recent CNN architectures [1, 3].

Typical Applications

VGG16 remains a widely used benchmark model due to its reliable feature extraction capabilities. It is commonly employed as a backbone network in object detection systems and other computer vision applications. The model is also used in generative tasks such as neural style transfer and image synthesis. In medical imaging, VGG16 is frequently adopted as a pre-trained model for transfer learning, demonstrating strong performance in the analysis of X-ray, CT, and MRI images [2, 3].

A.2.2 VGG19

Architecture Description

The VGG19 architecture represents a direct longitudinal scaling of the core VGG design framework, developed by the same group at Oxford to explore the absolute limits of network depth within a uniform convolutional paradigm [4]. The structural organization of VGG19 is expanded to incorporate a total of 19 learnable weight layers, which breaks down into 16 individual convolutional layers and 3 fully connected layers. Like its predecessor, it operates on a standard **224×224×3** input tensor and maintains the classic five-block grouping configuration. The network extends its depth by inserting one additional convolutional layer into the third, fourth, and fifth convolutional blocks, thereby thickening the intermediate processing stages before the feature data is flattened and passed into the terminal fully connected layers [3].

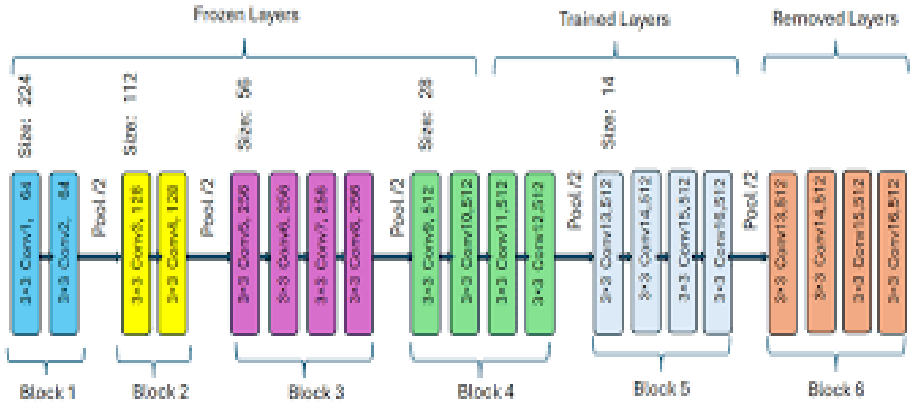


Figure A.2: VGG-19 model architecture.

Main Characteristics

VGG19 retains the identical fundamental characteristics that define the VGG lineage, adhering to a strict configuration of uniform **3×3** convolutional filters, one-pixel padding, and **2×2** max-pooling layers with a stride of two. The channel expansion schedule mirrors VGG16, scaling from **64** to **512** channels [1]. The addition of three extra convolutional layers raises the cumulative parameter count of VGG19 to approximately **144** million trainable weights. This increased layer density is specifically designed to expand the mathematical complexity of the model's internal transformations, enabling it to map more complex structural dependencies [3].

Advantages and Limitations

The extra convolutional depth of VGG19 allows it to capture higher-order feature relationships and subtle visual textures, enabling it to slightly outperform VGG16 on specific image recognition benchmark. This deep hierarchical setup provides a highly granular feature extraction pipeline that works well in transfer learning contexts where fine details are important.

Conversely, this increased depth introduces a clear trade-off with computational efficiency. The inclusion of more layers increases the required processing power, resulting in significantly slower training timelines and longer inference latencies during deployment. Because the accuracy gains over VGG16 are often marginal on standard tasks, the extra computational burden and high memory requirements of VGG19 mean that VGG16 is often preferred for applications where computational efficiency is a priority [3].

Typical Applications

It is primarily deployed in complex computer vision scenarios where maximum feature representation is required and sufficient computational hardware is available. It is used for fine-grained image classification, advanced object detection pipelines, and as a multi-layer feature map provider for deep feature fusion models. In medical imaging, VGG19 is frequently included in ensemble architectures to aid in the automated diagnosis of complex diseases from MRIs and high-resolution diagnostic scans [3].

A.3 ResNet50

Residual Learning Concept

As convolutional neural networks grew progressively deeper, researchers encountered a major obstacle known as the degradation problem: beyond a certain threshold, adding more layers to a highly deep network causes training accuracy to saturate and then degrade. This degradation is distinct from overfitting, as it manifests as a drop in both training and testing accuracy. It indicates that deeply stacked sequential layers struggle to learn identity mappings, making optimization difficult .

To address this challenge, researchers at Microsoft Research introduced the residual learning concept in 2015. Instead of forcing a series of stacked layers to directly approximate a desired underlying target mapping $\boldsymbol{H(x)}$, the network is configured to let these layers approximate a residual mapping, defined as $\boldsymbol{F(x) = H(x) - x}$. The original target mapping is subsequently recovered through a mathematical rearrangement, expressed as

$\boldsymbol{H(x) = F(x) + x}$, which fundamentally shifts how optimization gradients are propagated through the network topology [2][4].

Residual Blocks and Skip Connections

The residual learning framework is structurally operationalized through the integration of residual blocks, which modify the traditional feed-forward processing pipeline by introducing skip connections, also referred to as identity shortcut connections. A standard skip connection takes the original input tensor $\boldsymbol{x}$, bypassing it around one or more convolutional layers, and performs an element-wise mathematical addition with the output feature map $\boldsymbol{F(x)}$, generated by those layers. This architecture ensures that information can flow directly from the initial layers to the terminal stages of the network. If a specific block's convolutional layers perform poorly, the network can easily scale their weights toward zero, allowing the identity mapping to pass through unmodified. This setup addresses the vanishing gradient problem because the skip connections act as architectural conduits, allowing optimization gradients to flow backward through the layers during backpropagation [2][4].

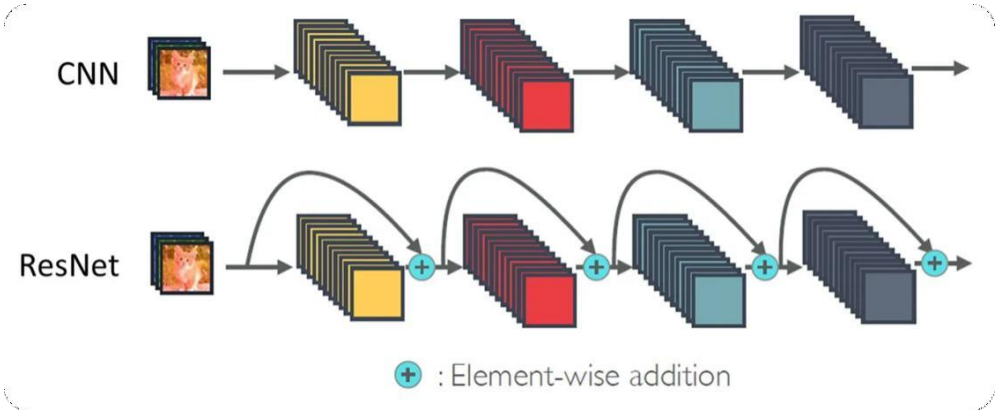


Figure A.3: Skip connections in ResNet50 architecture

Architecture Overview

ResNet50 is a 50-layer deep residual network designed to enable efficient training of very deep architectures. It begins with a 7×7 convolutional layer and a max-pooling layer for initial feature extraction and downsampling. The network is composed of four stages of bottleneck residual blocks, each containing a 1×1 convolution for channel reduction, a 3×3 convolution for feature extraction, and a final 1×1 convolution for channel restoration. Instead of large fully connected layers, ResNet50 uses a Global Average Pooling (GAP) layer followed by a Softmax classifier, reducing the number of trainable parameters [3][4].

Advantages and Limitations

The main advantage of ResNet50 is the use of residual (skip) connections, which alleviate the vanishing gradient problem and enable the successful training of deep networks. The adoption of Global Average Pooling further reduces parameter count, memory consumption, and the risk of overfitting while maintaining high classification performance.

However, ResNet50 has a more complex architecture than traditional CNNs. When feature map dimensions change, additional projection convolutions are required to align residual connections, increasing computational complexity. Moreover, repeated residual additions may introduce extra memory overhead during training and inference [3][4].

Applications

ResNet50 is widely applied in computer vision tasks such as image classification, object detection, semantic segmentation, and object tracking. In industrial settings, it is used

for defect detection and quality inspection. In medical imaging, ResNet50 has demonstrated strong performance in disease diagnosis, including X-ray analysis, diabetic retinopathy classification, and tumor detection in MRI scans [4].

A.4 EfficientNetB0

Compound Scaling Principle

Historically, the optimization of convolutional neural networks for improved performance followed arbitrary, single-dimensional scaling pathways, with developers manually increasing either network depth, network width, or input image resolution in isolation. Introduced by Mingxing Tan and Quoc V. Le of Google, the EfficientNet architecture introduced a new approach to network scaling through the compound scaling principle. This principle states that network dimensions are deeply interconnected, meaning that as input resolution increases, the network requires additional depth to expand its receptive fields and more channel width to capture finer feature granularities. Compound scaling uniformly adjusts depth, width, and resolution simultaneously using a fixed set of compound coefficients. This optimization approach ensures that the network balances all three structural dimensions, maximizing classification performance per computational unit [5][6].



Figure A.4: EfficientNetB0 model architecture

Architecture Overview

EfficientNetB0 serves as the baseline model for the EfficientNet family, developed using multi-objective neural architecture search (AutoML) to optimize both top-1 ImageNet accuracy and computational efficiency. The network architecture is structured around an initial Stem layer and a main body organized into seven distinct processing stages. The Stem layer processes the standard **224×224×3** input image using a **2D convolutional layer** with a **3×3 kernel** and a stride of two, followed by batch normalization and a smooth Swish activation function. The core body consists of a series of Mobile Inverted Residual Bottleneck (MBConv) blocks with varying configurations. The network architecture concludes with a terminal head that includes a **1×1** projection convolution, a global average pooling layer, and a dense layer with a Softmax activation function [4][5][6].

Efficiency and Computational Advantages

The computational efficiency of EfficientNetB0 is primarily driven by its use of depthwise separable convolutions within the MBConv blocks. In contrast to standard convolutions, which filter and combine features across all channels simultaneously, a depthwise separable convolution splits the operation into two distinct stages. First, a spatial depthwise convolution applies a single localized filter to each individual input channel independently to capture spatial characteristics. Second, a pointwise **1×1** convolution aggregates these independent channels to mix information and adjust channel depth. This factorization significantly reduces the number of required mathematical multiplications.

Furthermore, EfficientNetB0 integrates Squeeze-and-Excitation (SE) optimization blocks, which perform channel-wise feature recalibration. The SE block uses global average pooling to summarize each channel, passes this summary through a small bottleneck to learn channel relationships, and generates attention weights that selectively amplify informative feature channels while suppressing irrelevant noise [5][6].

Advantages and Limitations

The main advantage of EfficientNetB0 is its high parameter efficiency, allowing it to match or exceed the accuracy of traditional CNNs while using a fraction of the parameters and computational resources. This makes it well-suited for deployment on resource-constrained edge computing devices and mobile hardware. The inclusion of the smooth Swish

activation function and adaptive SE attention blocks ensures stable gradient propagation and high feature representation capabilities.

However, EfficientNetB0 requires precise hyperparameter tuning during training, and its multi-stage depthwise separable operations and non-monotonic activation functions can introduce execution overhead on hardware platforms that are not fully optimized for non-standard convolutions [5][6].

Applications

EfficientNetB0 is frequently used in high-efficiency visual recognition pipelines, mobile computer vision systems, and embedded real-time processing tasks. It is widely applied in facial emotion expression tracking and automated edge-based diagnostic devices. In medical computer vision, it provides a lightweight architecture for transfer learning, allowing clinical models to achieve high diagnostic accuracy on complex medical scans without requiring high-end computing clusters [6].

A.5 DenseNet121

Dense Connectivity Concept

Introduced in 2017 by Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger, the Densely Connected Convolutional Network (DenseNet) architecture modified traditional feed-forward structures by proposing a dense connectivity pattern. In a standard convolutional layout, each layer receives its input exclusively from the immediate preceding layer. DenseNet changes this paradigm by establishing direct forward-passing connections from every single layer to all subsequent layers within a specific block. Consequently, for a given layer, the feature maps of all prior layers are systematically aggregated and utilized as its input, while its own generated feature maps are passed along as inputs to every layer that follows. This architecture modifies the structural relationship between layers, ensuring a continuous, direct flow of information throughout the network [7].

Dense Blocks and Feature Reuse Mechanism

The core architecture of DenseNet is organized around dense blocks, which contain multiple convolutional layers interconnected through this forward-passing pattern. Crucially, inside a dense block, layers combine feature maps using channel concatenation rather than the

element-wise addition seen in residual networks. If a dense block contains m individual layers, the 1^{st} layer receives a concatenated tensor comprising the original input and the outputs of all preceding layers. The number of processing feature maps generated by each individual layer is governed by a hyperparameter known as the growth rate (k). If each layer produces k feature maps, the network can maintain thin individual layers (e.g., $k = 32$), since deep layers have direct access to the accumulated features of all previous steps. This feature reuse mechanism minimizes visual redundancy, as the network avoids duplicating identical feature maps across different layers [7].

Architecture Overview

DenseNet121 has 121 layers with learnable weights, structured as a DenseNet-BC variant that includes both bottleneck layers and channel compression mechanisms. The network accepts an initial **224×224×3** image tensor, applying a **7×7 convolution** with a stride of two and a **3×3 max-pooling** layer to perform initial downsampling. The main body consists of four distinct dense blocks, which contain 6, 12, 24, and 16 convolutional layers respectively. Within these dense blocks, each layer uses a "BN-ReLU-Conv" processing sequence, utilizing a **1×1 bottleneck** convolution to reduce channel numbers before **3×3 convolution** maps spatial features.

To manage dimensional changes between dense blocks, intermediate transition layers are inserted. A transition layer includes batch normalization, a **1×1** convolution to compress channel depth via a compression factor (θ), and a **2×2** average pooling layer to downsample the spatial grid. The model concludes with a global average pooling layer and a dense classification layer with a Softmax activation function [7][8].

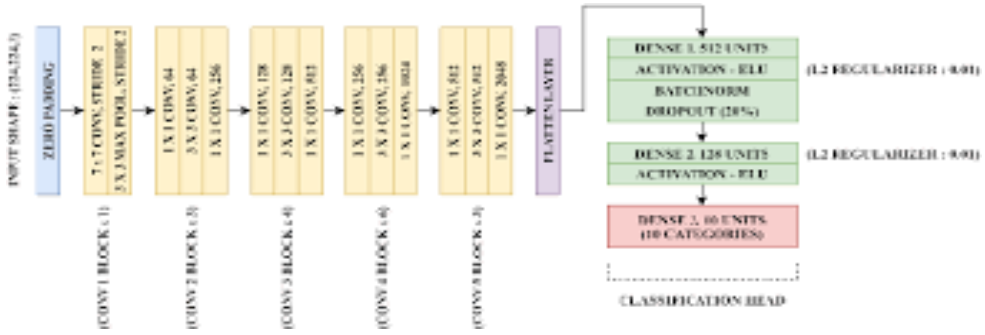


Figure A.5: DenseNet121 model architecture

Advantages and Limitations

The primary advantage of DenseNet121 is its ability to address the vanishing gradient problem in deep network setups. Because every layer is directly connected to all subsequent layers, each layer has direct access to the gradients from the final loss function, ensuring efficient gradient propagation during backpropagation. DenseNet121 achieves high parameter efficiency, extracting rich feature representations without requiring the massive parameter footprints of traditional architecture.

The main limitation of DenseNet121 is its high training memory consumption. Because the network continuously concatenates feature maps, it generates large, multi-channeled tensors that must be stored in hardware memory during backpropagation, resulting in high GPU memory utilization during intensive training cycles [7].

Applications

DenseNet121 is widely utilized in tasks requiring the representation of fine-grained textures and complex spatial patterns. It is heavily deployed in fine-grained object classification, satellite image interpretation, and unmanned aerial vehicle (UAV) light imagery tracking systems. In medical imaging, DenseNet121's feature reuse mechanism makes it effective for computer-aided diagnostics, where it is used to identify subtle lesions, track anatomical variations, and classify structural anomalies in complex radiography and magnetic resonance image slices [7][8].

A.6 Inception Network

Inception Modules and Multi-Scale Feature Extraction

A key challenge in convolutional network design is selecting the optimal filter size for a given layer; larger filters (**e.g., 5×5**) are well-suited for capturing global structural patterns, while smaller filters (**e.g., 3×3**) excel at detecting fine-grained regional details. The Inception network family addresses this challenge by introducing multi-scale feature extraction within a parallel processing block called an Inception module. Instead of forcing a layer to operate with a single filter size, an Inception module executes multiple parallel convolutional operations with different filter sizes (such as **1×1, 3×3**, and **5×5**) alongside a max-pooling operation within the same block. The outputs of these parallel paths are concatenated along the channel dimension, allowing the network to capture both broad topological structures and fine details simultaneously [9][10].

Architecture Overview

To expand network depth and width while managing computational complexity, Inception V2 and Inception V3 introduced advanced structural factorizations. A primary focus was addressing representational bottlenecks, where severe dimensionality reductions cause a loss of spatial information. Inception V2 replaced large **5×5 filters** with two sequentially stacked **3×3 convolutions**, which achieve an identical effective receptive field while reducing computational cost by a factor of 2.78.

Furthermore, the architecture introduced asymmetric convolution factorization, splitting $n \times n$filter into a sequence of $1 \times n$and $n \times 1$convolutions. For example, **3×3** convolution can be factored into **1×3** convolution followed by **3×1** convolution, which is 33% cheaper computationally. This asymmetric factorization is applied in the intermediate layers of the network where feature map sizes are balanced. InceptionV3 operates as 48-**layer** deep network that expects a default input image size of **299×299×3** , The architecture includes auxiliary classifiers to improve gradient flow in deep layers and incorporates Label Smoothing Regularization (LSR) during training. LSR modifies target class labels to prevent the model from predicting outputs with extreme confidence, which improves regularization and helps prevent overfitting [9][10].

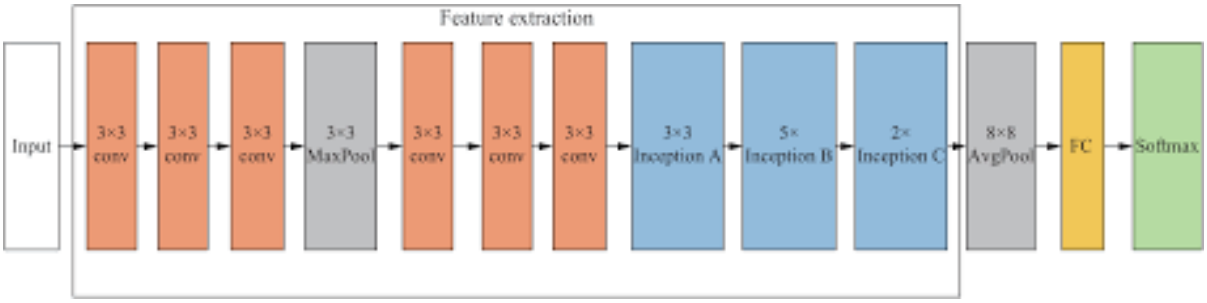


Figure A.6: Inception model architecture

Advantages and Limitations

The primary advantage of the Inception Network, particularly InceptionV3, is its ability to perform multi-scale feature extraction without causing an explosive increase in parameters. The use of factored and asymmetric convolutions allows the model to expand both its depth and width while remaining computationally efficient.

The main limitation of the Inception architecture is its high structural complexity. The parallel configuration of non-uniform convolutional paths makes the network difficult to modify or specialize for distinct clinical tasks. Additionally, the parallel processing layout can introduce memory latency overhead on hardware architectures that prefer uniform, sequential data access patterns.

Applications

InceptionV3 is widely used in large-scale visual recognition tasks, satellite image mapping, and object detection systems. It is frequently deployed in multi-label classification tasks where objects appear at varying scales within the same image frame. In medical imaging, InceptionV3 is used for automated diagnostic classification, including the analysis of Pap screenings for cervical cancer, classification of dermatological lesions, and identification of anomalies in high-resolution medical imaging scans [4][9][10].

A.7 Comparative Analysis of CNN Architectures

Architectural Differences and Parameter Distribution

The CNN architectures considered in this study follow different design strategies. VGG16 and VGG19 employ a simple sequential architecture based on stacked convolutional

layers, providing strong feature extraction capabilities but resulting in a very large number of parameters, mainly concentrated in the fully connected layers [1][3].

ResNet50 introduces residual connections that facilitate the training of deeper networks and mitigate the degradation problem [4]. EfficientNetB0 improves parameter efficiency through compound scaling and neural architecture search, balancing network depth, width, and input resolution [5][6]. DenseNet121 promotes feature reuse through dense connections between layers, reducing redundancy and improving information flow [7]. InceptionV3 utilizes parallel multi-scale convolutions to efficiently capture features at different spatial scales while controlling computational cost [9].

Computational Complexity and Memory Requirements

The computational and memory requirements of these models differ considerably. VGG16 and VGG19 are the most resource-intensive architectures, containing approximately 138–144 million parameters and requiring substantial memory and storage capacity [1][3].

ResNet50 and InceptionV3 offer a more balanced trade-off between performance and efficiency by replacing large fully connected layers with Global Average Pooling, thereby reducing parameter count and memory usage [3][4]. DenseNet121 achieves high parameter efficiency through feature reuse, although its dense concatenation mechanism can increase memory consumption during training [7]. EfficientNetB0 is the most computationally efficient among these architectures, employing depthwise separable convolutions and compound scaling to minimize computational cost while maintaining strong performance [5][6].

Comprehensive Architecture Comparison

The primary structural and operational characteristics of the evaluated convolutional neural network architectures are summarized in Table A.1 below.

Table A.1: Comprehensive comparison of CNN architecure

Model Name	***Year of Publication***	***Network Depth***	***Number of Parameters***	***Key Innovation***	***Main Advantages***	***Main Limitations***
VGG16	2014	16 Learnable Layers	~138 Million	Uniform 3×3 convolution filters stacked sequentially	Symmetrical structure; easy to implement and modify	Large model size (~528 MB); computationally expensive
VGG19	2014	19 Learnable Layers	~144 Million	Deepened stacked uniform sequence	Extracts complex hierarchical features	Slower training speeds; high computational redundancy
ResNet50	2015	50 Learnable Layers	Moderate Parameter Footprint	Identity shortcut skip connections	Avoids vanishing gradients; enables deep training	Managing dimensional matching during shortcuts
InceptionV3	2015/2016	48 Learnable Layers	Moderate Parameter Footprint	Factored and asymmetric parallel convolutions	Multi-scale feature extraction; low resource usage	Complex, non-uniform layout; hard to specialize
DenseNet121	2017	121 Learnable Layers	High Parameter Efficiency	Concatenation-based dense connectivity	Alleviates vanishing gradients; extensive feature reuse	High memory utilization during backpropagation
EfficientNetB0	2019	Baseline MBConv Block Structure	Low Parameter Footprint	Balanced compound scaling via AutoML	High parameter efficiency; optimized for edge hardware	Requires careful hyperparameter optimization

Appendix B : Variational and Convolutional Autoencoders

B.1 Introduction

In the contemporary landscape of deep learning and computer vision, representation learning has emerged as a fundamental paradigm. Historically, machine learning systems relied heavily on manual feature engineering, an intensive process requiring domain expertise to design attributes capable of isolating relevant information from raw data arrays. Representation learning shifts this burden onto deep neural networks, enabling systems to automatically discover, extract, and organize the underlying structures, variations, and latent factors directly from raw, unannotated datasets. By processing data through successive non-linear layers, these networks transform high-dimensional inputs into abstract, high-level representations that capture the core semantic essence of the data [1][4].

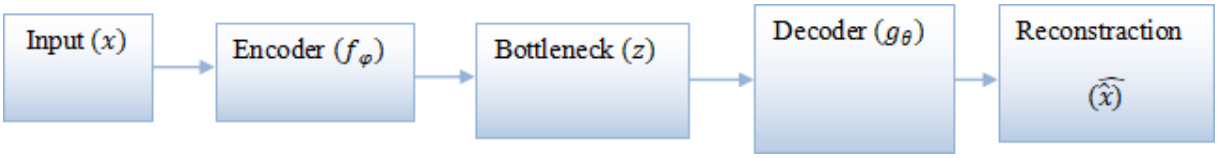
The extraction of these representations is closely bound to the principles of dimensionality reduction, which plays a pivotal role when dealing with high-dimensional data profiles. In fields such as digital image processing and medical imaging, raw data instances are frequently represented as massive multi-dimensional vectors; for instance, even minor pixel arrays can span thousands of individual dimensions where each dimension corresponds to a pixel value or a specific color channel. However, a significant portion of this high-dimensional space contains background noise, redundant structural configurations, or irrelevant textual information rather than the vital underlying variables that dictate the data's true distribution. Compressing the input space down to a lower-dimensional manifold containing only the most salient attributes the latent space is critical to mitigating the curse of dimensionality, enhancing computational efficiency, and boosting the accuracy of downstream analytical tasks[13].

This objective provides the primary motivation for the development of autoencoders. Autoencoders are deep learning, self-supervised connectionist architectures engineered specifically to bridge the gap between unsupervised data compression and structural reconstruction. By establishing an informational bottleneck within the network, they are forced to discard non-essential ambient variations and capture only the principal factors of variation required to rebuild the input instance. Because they operate in a self-supervised

manner, using the input data itself as the target label, autoencoders remove the necessity for manual human annotation, providing a scalable and powerful framework for feature extraction, noise reduction, and data generation across diverse scientific domains[2][8][11].

B.2 Fundamentals of Autoencoders

An autoencoder is formally defined as an unsupervised or self-supervised neural network architecture whose primary training objective is to compress an input data instance into a lower-dimensional representation and subsequently reconstruct the original input from this compressed form with minimal discrepancy. Operating on an identity mapping paradigm, the model is trained to minimize a specified loss function that quantifies the divergence between the original input vector and the network's generated output. This framework enables the network to learn non-linear feature extractions and acquire an unsupervised understanding of complex data manifolds without requiring explicit supervisory labels [4][11].



The structural architecture of a standard autoencoder is bisected into two primary components: the encoder network and the decoder network, which are separated by an intermediate informational bottleneck. The encoder network, denoted functionally as a mapping function $\boldsymbol{f_\varphi}$ parameterized by weights and biases φ, accepts the high-dimensional input data $\boldsymbol{x}$ and projects it into a lower-dimensional space. Structurally, a standard "vanilla" or multilayer perceptron autoencoder implements an encoder where each subsequent hidden layer contains progressively fewer operational nodes than its predecessor, compressing the input data through a sequence of dimensionality-reducing layers. Conversely, the decoder network, denoted as a mapping function $\boldsymbol{g_\theta}$ parameterized by weights and biases θ, accepts the compressed representation from the bottleneck layer and executes the inverse operations. Each subsequent layer within the decoder contains a progressively larger number of active nodes, systematically scaling the compressed representation back up to match the original input dimensions.

The localized layer situated at the intersection of the encoder and decoder is known as the bottleneck, or the code layer. The bottleneck acts as the designated repository for the latent space representation$\boldsymbol{z}$, which constitutes the fully compressed, lower-dimensional embedding of the original input data. Maintaining a highly constrained bottleneck capacity is structurally imperative; if the bottleneck contains an excessive number of operational dimensions, the network can easily learn to copy or memorize the raw input data directly to the output layer. Such trivial memorization would satisfy the mathematical conditions of the reconstruction objective while completely failing to force the network to isolate the core underlying latent variables or learn meaningful, generalizable structural mappings[8].

The reconstruction process operates as a forward-propagation pass through the combined encoder-decoder apparatus, where the input $\boldsymbol{x}$ is converted into the latent vector$\boldsymbol{z = f_{\varphi}(x)}$, which is then mapped to the output reconstruction $\boldsymbol{\hat{x} = g_{\theta}(z)}$. During the training phase, optimization algorithms, such as stochastic gradient descent or the Adam optimizer, are iteratively employed to adjust the network parameters $\boldsymbol{\varphi}$ and $\boldsymbol{\theta}$ after each epoch. This optimization is driven by a pixel-wise or element-wise reconstruction error function such as Mean Squared Error (MSE) or Binary Cross-Entropy (BCE) loss which computes the explicit difference between the input $\boldsymbol{x}$ and the reconstruction $\hat{\boldsymbol{x}}$. As the reconstruction error is iteratively minimized via backpropagation, the encoder parameters adjust to filter out noise and retain the features most conducive to accurate restoration, while the decoder parameters adjust to interpret those latent vectors with increasing fidelity [6].

While standard autoencoders excel at non-linear feature extraction, data compression, and deterministic tasks like image denoising, they possess structural limitations rooted in the unconstrained geometry of their latent spaces. In a standard autoencoder, the latent space is discrete and unregularized, meaning the encoder maps specific inputs to isolated, fixed coordinate positions within the bottleneck layer without establishing any global continuity or distributional constraints. Consequently, the latent space frequently develops significant gaps or "holes" where no training data has been mapped. If an analytical practitioner attempts to sample a random coordinate vector from these unmapped regions, the decoder will output

meaningless, highly corrupted data. This lack of continuity and completeness prevents standard autoencoders from performing smooth data interpolation or acting as viable generative models capable of controlled sample synthesis[1][2][8].

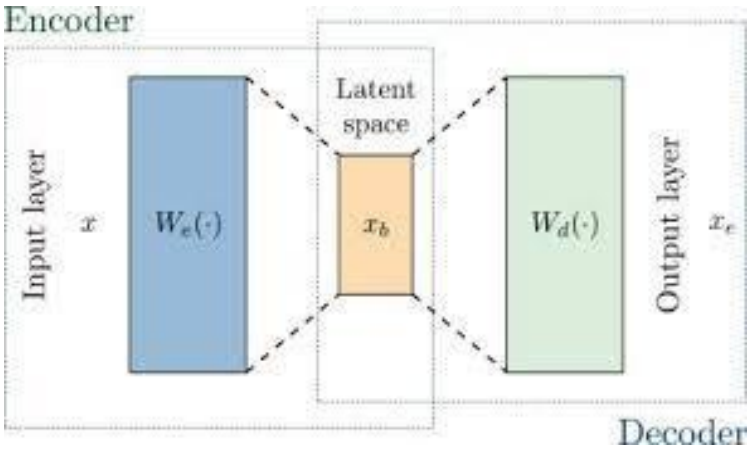


Figure B.1: General autoencoder architecture

B.3 Variational Autoencoders (VAE)

B.3.1 Introduction and Probabilistic Interpretation

To overcome the structural limitations of standard architectures, Kingma and Welling proposed the Variational Autoencoder (VAE) in their 2013 paper, "Auto-Encoding Variational Bayes". Unlike standard autoencoders that rely on a deterministic mapping to output fixed coordinate vectors, VAEs are formal generative models rooted in probabilistic latent variable modeling and variational Bayesian inference. A VAE replaces the discrete, unconstrained latent space of traditional models with a continuous, smooth, and probabilistic latent space distribution. This structural evolution enables the network to accurately reconstruct original inputs while simultaneously empowering it to synthesize entirely new, realistic variations of the training data by sampling directly from the learned latent distribution [1][2].

B.3.2 Latent Variable Modeling

The mathematical foundation of a VAE assumes that the high-dimensional observable data distribution $\boldsymbol{x}$ is generated by a directed model that depends on a set of underlying, continuous, and unobservable latent variables $\boldsymbol{z}$. In this probabilistic framework, the true prior

distribution of the latent variables is denoted as $\boldsymbol{p(z)}$, while the conditional distribution that governs the generation of data instances from the latent space is expressed as $\boldsymbol{p_\theta(x|z)}$, parameterized by the decoder weights $\boldsymbol{\theta}$. To perform accurate inference and generate new data samples, the system must evaluate the posterior distribution $\boldsymbol{p(z|x)}$, which represents the probability of a latent configuration given a specific observed data instance[4].

According to Bayes' theorem, this posterior is formulated as:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \tag{B.1}$$

However, direct computation of the marginal likelihood or evidence, denoted as $p(x)$, is highly intractable because it requires computing an integral across all possible continuous configurations of the entire latent manifold:

$$p(x) = \int p(x|z)p(z)dz \tag{B.2}$$

Because evaluating this integral over a continuous, high-dimensional latent space would require a theoretically infinite amount of computation time, the true posterior distribution $\boldsymbol{p(z|x)}$ is mathematically intractable. To resolve this obstacle, VAEs employ variational inference to approximate the true, intractable posterior distribution $\boldsymbol{p(z|x)}$using a tractable, parameterized distribution denoted as $\boldsymbol{q_\varphi(z|x)}$, which is learned by the encoder network parameterized by weights and biases $\boldsymbol{\varphi}$[1][4][2].

B.3.3 Encoder and Decoder in VAE

In VAE architecture, the encoder network operates as an inference network that maps the high-dimensional input data $\boldsymbol{x}$ to the parameters of the approximate posterior distribution $\boldsymbol{q_{\varphi}(z|x)}$, rather than mapping it to a single fixed point. For every latent feature dimension defined in the bottleneck, the encoder network outputs two distinct statistical vectors: a vector of means $\boldsymbol{\mu}$ representing the central spatial location of the data encoding, and a vector of standard deviations $\boldsymbol{\sigma}$ or log-variances $\boldsymbol{\log \sigma^2}$ quantifying the expected variance or statistical range of possibilities around that mean. These statistical parameters explicitly define a continuous Gaussian distribution from which latent vectors can be sampled. The decoder network then acts as a generative network that accepts a sampled latent vector $\boldsymbol{z}$ from this continuous distribution and maps it back into the high-dimensional data space to produce a reconstruction $\hat{\boldsymbol{x}}$ that structurally resembles the original training data [1][2][4].

B.3.4 Reparameterization Trick

Because the latent representation in a VAE is determined by sampling from the distribution defined by $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, a critical structural problem arises during training: the stochastic sampling step $\boldsymbol{z \sim q_{\varphi}(z|x)}$ introduces a non-differentiable operation into the forward pass. Standard backpropagation algorithms require every node within the computational graph to be fully differentiable so that partial gradients can flow from the loss function back to the network weights. A purely random sampling node blocks this gradient flow, rendering it impossible to optimize the encoder parameters $\boldsymbol{\varphi}$ using standard gradient descent techniques.

To circumvent this limitation without eliminating the required statistical variation, Kingma and Welling introduced the reparameterization trick. This technique isolates the stochasticity of the sampling process by introducing an auxiliary noise variable $\boldsymbol{\epsilon}$ that is

sampled from a standard normal distribution, completely independent of the network's internal parameters:

$$\epsilon \sim N(0, I) \tag{B.3}$$

By shifting the source of randomness outside the trainable architecture, the latent vector $\boldsymbol{z}$ can be computed as a purely deterministic linear combination of the encoder's statistical outputs and the external noise vector $\boldsymbol{\epsilon}$. Mathematically, this transformation is formulated using the mean and log-variance vectors as follows:

$$z = \mu + \exp(0.5 \log \sigma^2) \odot \epsilon \tag{B.4}$$

Where $\boldsymbol{\Theta}$ denotes an element-wise vector product. Because $\boldsymbol{\mu}$ and $\boldsymbol{\log \sigma^2}$ are deterministic outputs of the encoder network layers, the entire equation becomes fully differentiable with respect to the encoder parameters $\boldsymbol{\varphi}$. Gradients can now flow directly through the latent vector $\boldsymbol{z}$ back into the encoder weights during backpropagation, preserving the differentiability of the system while maintaining the continuous randomness required for probabilistic data generation[1][2].

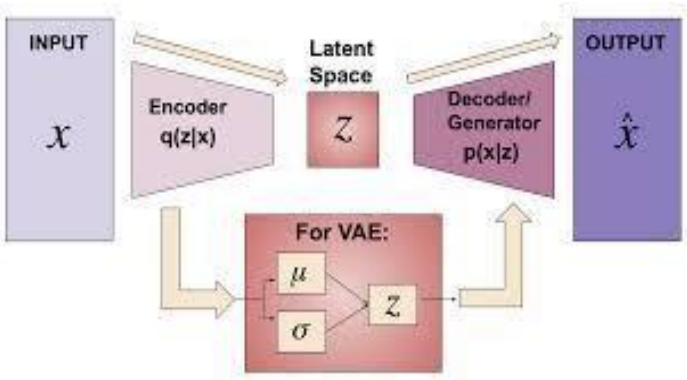


Figure B.2: VAE model architecture

B.3.5 Loss Function and Mathematical Formulation

To train the encoder and decoder parameters simultaneously, the optimization framework seeks to minimize the difference between the approximate distribution $\boldsymbol{q_{\varphi}(z|x)}$ and the true posterior $\boldsymbol{p(z|x)}$ by minimizing their **Kullback-Leibler (KL) divergence**:

$$\min_{\varphi} KL(q_{\varphi}(z|x)||p(z|x) \tag{B.5}$$

Through algebraic expansion of this minimization problem, the objective can be proven to be mathematically equivalent to maximizing a formulation known as the **Evidence Lower Bound (ELBO):**

$$ELBO = E_{q_{\varphi}(z|x)}[\log p_{\theta}(x|z) - KL(q_{\varphi}(z|x)||p(z))] \tag{B.6}$$

Consequently, the total objective loss function minimized during VAE training is formalized as the negative **ELBO**, which comprises two distinct and competing operational terms:

$$L_{VAE}(x,\hat{x}) = L_{recon}(x,\hat{x}) + \sum_{j} KL(q_j(z|x)||p(z) \tag{B.7}$$

The first term represents the reconstruction loss, which optimizes the likelihood $\log p_\theta(x|z)$to ensure that the decoder effectively minimizes the physical discrepancy between the raw input data x and its reconstructed counter-image $\hat{x}$ using metrics like MSE or cross-entropy [1][2]. The second term represents the **KL** divergence regularization loss, which measures how closely the learned approximate posterior distribution $q_\varphi(z|x)$ matches the true prior distribution $p(z)$**.** In a standard VAE, the prior $p(z)$ is defined as a standard Gaussian distribution $N(0,I)$**.** Assuming a diagonal covariance structure, the closed-form component-wise **KL** divergence for each latent dimension j is formulated as:

$$L_{KL} = -0.5 \cdot \sum_j \left(1 + \sigma_j{}^2 - \mu_j{}^2 - \exp(\log \sigma_j{}^2)\right) \tag{B.8}$$

This dual-term optimization creates a balance: while the reconstruction loss drives the network to separate unique attributes for precise reconstruction, the **KL** divergence loss acts as a regularization penalty that forces the latent distributions to conform to a standard normal Gaussian space, preventing the encoder from isolating distinct classes into disjoint clusters [1][2].

B.3.6 Advantages, Limitations, and Applications

The main advantage of Variational Autoencoders (VAEs) is their ability to learn a continuous and complete latent space. Continuity ensures that nearby latent points generate similar outputs, while completeness allows any sampled point from the latent distribution to produce meaningful data. VAEs also provide more stable training than GANs because they optimize a well-defined probabilistic objective. Conditional VAEs (CVAEs) further enable controlled data generation by incorporating class labels into the learning process.

A key limitation of VAEs is the tendency to generate blurry outputs. Since the model reconstructs data from a probability distribution and optimizes pixel-wise reconstruction loss, fine details and sharp edges are often smoothed, resulting in lower visual quality compared to GAN-generated images [2].

VAEs are widely used for data generation, image interpolation, and anomaly detection. In medical imaging, they support tissue characterization and pathology detection by identifying deviations from learned representations of healthy anatomical structures [1][9][12].

B.4 Convolutional Autoencoders (CAE)

B.4.1 Introduction and Feature Extraction

Traditional autoencoders based on fully connected networks are inefficient for image processing because they flatten images into one-dimensional vectors, losing important spatial relationships and requiring a large number of parameters. Convolutional Autoencoders (CAEs) address these limitations by integrating CNN-based convolutional layers into the autoencoder framework.

CAEs automatically learn hierarchical image features directly from raw data. Early convolutional layers capture simple patterns such as edges and color transitions, while deeper layers extract more complex structures, textures, and object boundaries. This hierarchical learning preserves spatial information and improves feature representation for image analysis [5][6][8].

B.4.2 Encoder–Decoder Architecture and Reconstruction Mechanism

A CAE consists of an encoder and a decoder. The encoder compresses the input image through convolutional layers and pooling operations, reducing spatial dimensions while extracting informative features. The decoder reconstructs the original image using transposed convolutional layers that progressively restore spatial resolution. A Sigmoid activation function is commonly used in the output layer to generate pixel values within the normalized range of 0–1 [6].

The quality of reconstruction depends largely on the bottleneck representation. A smaller bottleneck forces the model to learn highly compressed and generalized features,

which may reduce reconstruction detail. Conversely, a larger bottleneck preserves more information and improves reconstruction quality but may reduce the effectiveness of feature abstraction if excessive capacity is provided [8].

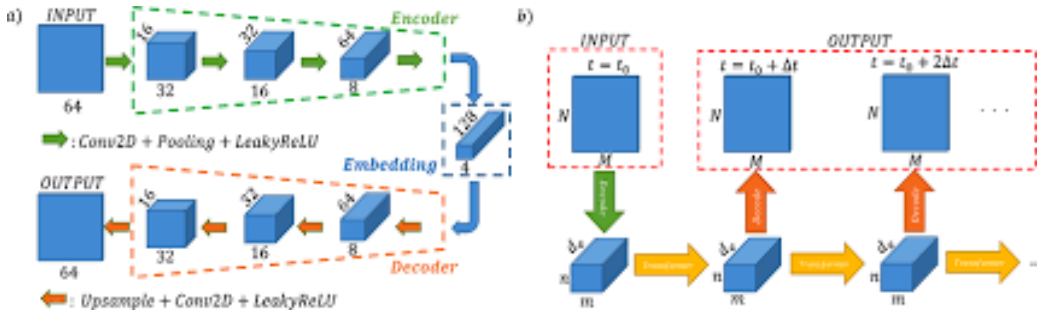


Figure B.3: CAE model architecture

B.4.3 Advantages, Limitations, and Applications

Convolutional Autoencoders (CAEs) preserve local spatial relationships in images while significantly reducing the number of trainable parameters through weight sharing and local connectivity. This improves generalization and enables efficient processing of large-scale image data [5][6]. However, CAEs may lose fine texture details during downsampling and can produce checkerboard artifacts due to transposed convolution operations in the decoder [6].

CAEs are widely applied in image compression, feature extraction, dimensionality reduction, and image denoising [5][8]. In medical imaging, they are particularly useful for denoising corrupted scans and extracting compact feature representations for subsequent classification tasks [4][13].

B.5 Comparison between VAE and CAE

CAEs and VAEs differ mainly in their latent space representation and learning objectives. A CAE is a deterministic architecture based on convolutional and transposed convolutional layers, where the latent space consists of a fixed feature vector optimized to minimize reconstruction error [5]. In contrast, a VAE is a probabilistic model that learns a continuous latent distribution by encoding data into the parameters μ and $\log \sigma^2$ and sampling through the reparameterization trick [2].

Training objectives also differ. CAEs optimize only a reconstruction loss, such as MSE, to preserve image details [6][8]. VAEs use a combined objective that balances reconstruction loss with KL divergence regularization, ensuring a smooth and continuous latent space aligned with a Gaussian prior [2].

These approaches can be combined in a Convolutional Variational Autoencoder (CVAE), which integrates the spatial feature extraction capabilities of CAEs with the probabilistic latent representation of VAEs [1][10].A comprehensive comparative analysis of these individual autoencoding structures is summarized in **Table B.1**:

Table B.1: Comparative summary between VAE & CAE.

Dimension	Variational Autoencoder (VAE)	Convolutional Autoencoder (CAE)
Latent Space Geometry	Continuous, smooth, regularized probabilistic distribution (typically Gaussian).	Discrete,unconstrained, unregularized deterministic coordinate space.
Optimization Objective	Negative ELBO: Balances reconstruction likelihood with a KL divergence penalty.	Reconstruction Loss: Minimizes pixel-wise discrepancy via MSE or BCE.
Primary Strengths	Capable of controlled sample generation, smooth interpolation, and stable training.	Preserves 2D spatial structures, shift/translation invariance, and parameter efficient.
Main Weaknesses	Tends to generate blurrier outputs due to averaging features over a distribution.	Prone to unmapped latent regions ("holes"), rendering it unsuitable for controlled generation.
Representative Use Cases	Controlled data generation, image synthesis, molecular discovery, anomaly detection.	Image denoising, spatial dimensionality reduction, structural feature extraction.

Appendix C : Pipelines of This Study

C.1 ***Pneumonia Detection***

C.1.1 CNN models

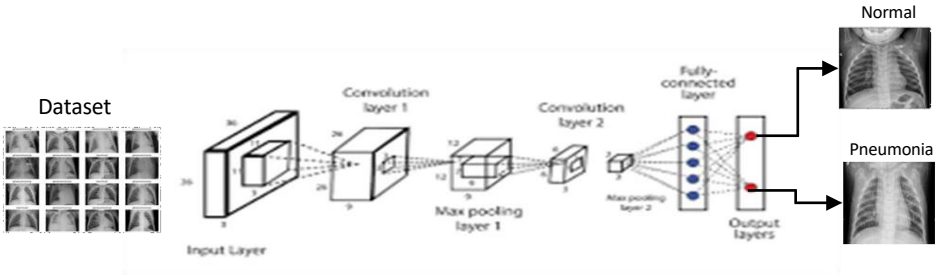


Figure C.1: Pneumonia detection flowchart by using CNNs architecture

C.1.2 AE models

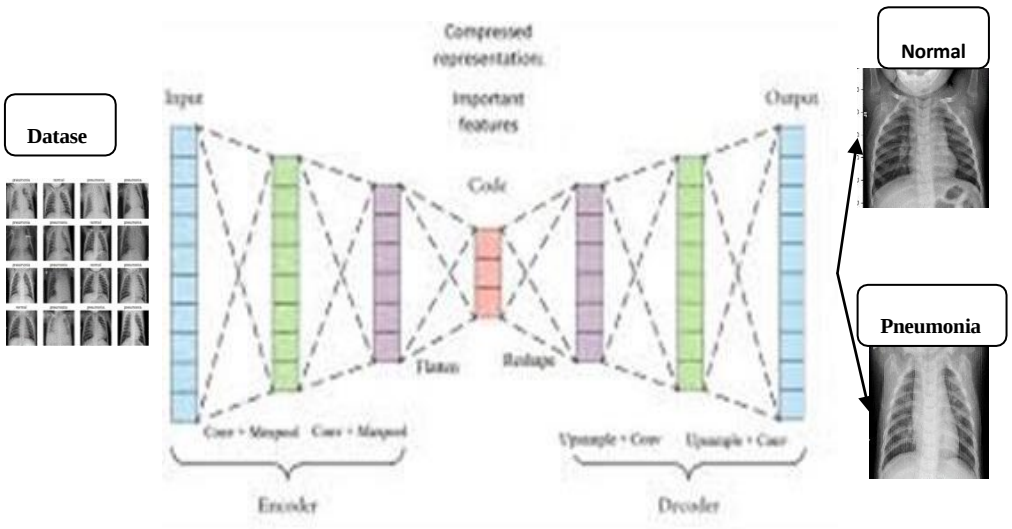


Figure C.2: Pneumonia detection flowchart by using AEs architecture

C.2 Brain tumor Classification

C.2.1 CNN models

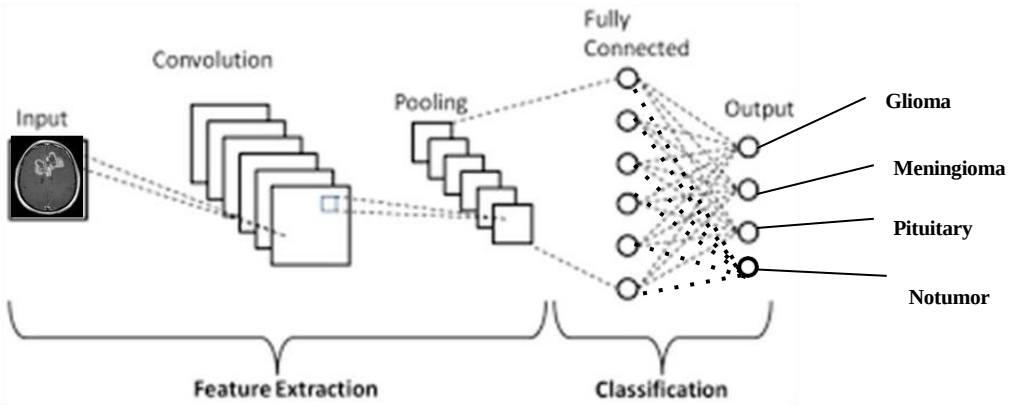


Figure C.3: Brain Tumor classification flowchart by using CNNs architecture

C.2.2 AE models

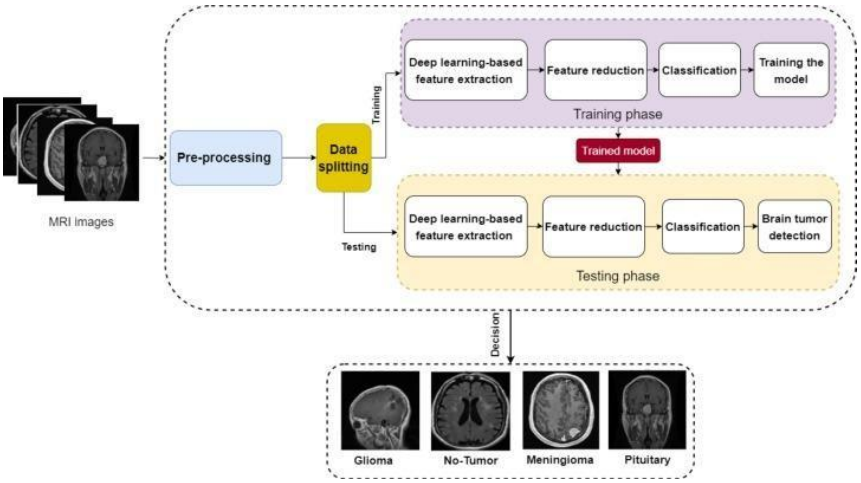


Figure C.4: Brain Tumor classification flowchart by using AEs architecture

Appendix D : ROC Curves

D.1 Brain Tumor Classification

D.1.1 CNN models

D.1.1.1 **VGG-16 model**

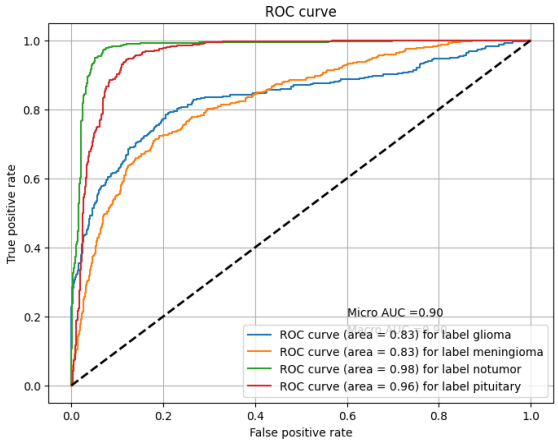


Figure D.1: ROC curves for VGG-16 in brain tumor classification

D.1.1.2 VGG-19 model

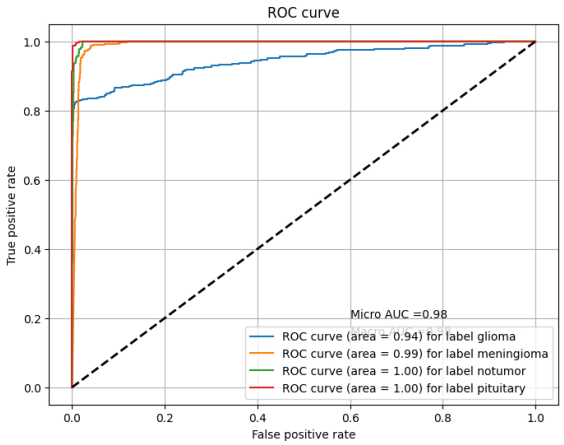


Figure D.2: ROC curves for VGG-19 in brain tumor classification

D.1.1.3 EfficientNet-B0

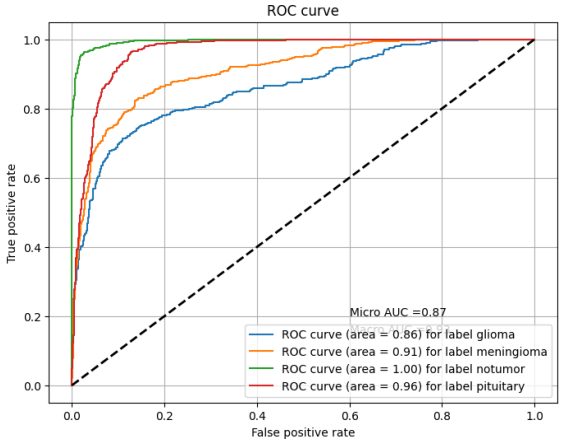


Figure D.3: ROC curves for EfficientNet-B0 in brain tumor classification

D.1.2 AE models

D.1.2.1 VAE

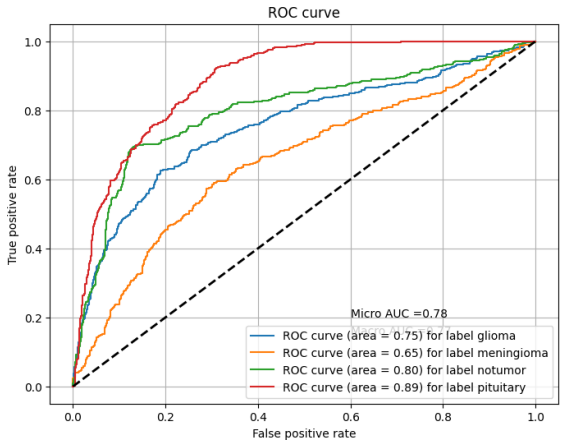


Figure D.4: ROC curves for VAE in brain tumor classification

D.1.2.2 CAE

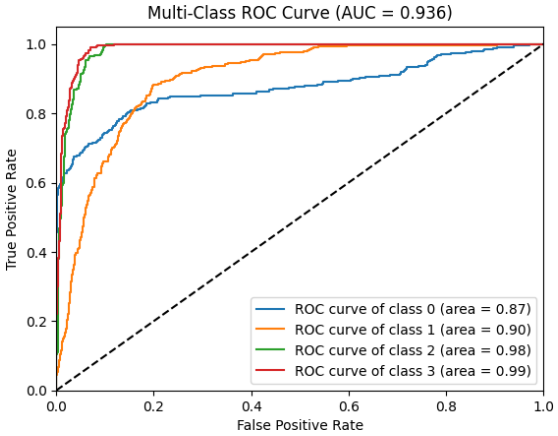


Figure D.5: ROC curves for CAE in brain tumor classification

D.2 Pneumonia Detection

D.2.1 CNN models

D.2.1.1 VGG-16

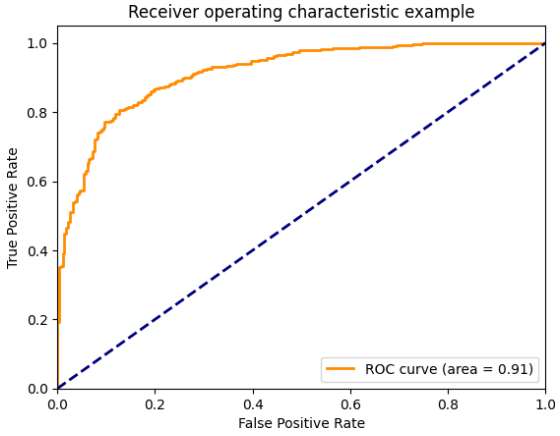


Figure D.6: ROC curves for VGG-16 in pneumonia detection

D.2.1.2 VGG-19

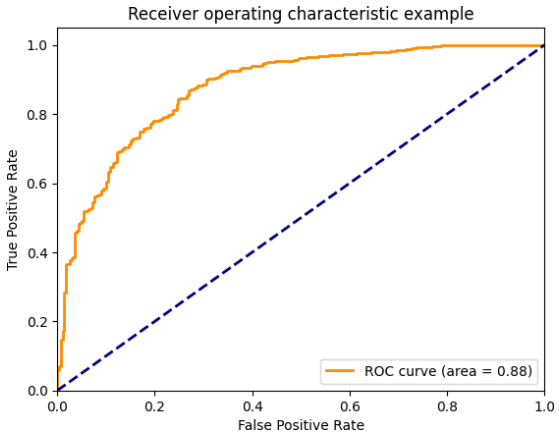


Figure D.7: ROC curve for VGG-19 in pneumonia detection

D.2.1.3 EfficientNet-B0

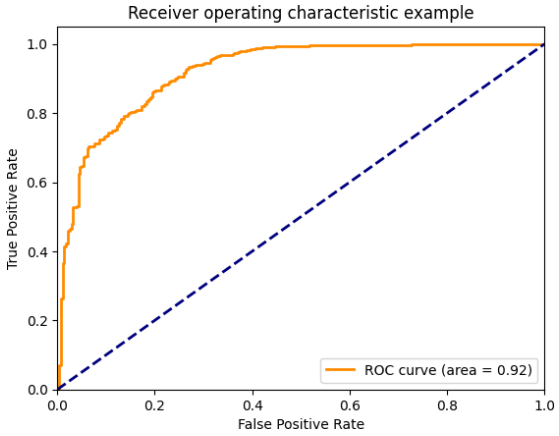


Figure D.8: ROC curve for EfficientNet-B0 in pneumonia detection

D.2.1.4 ResNet50

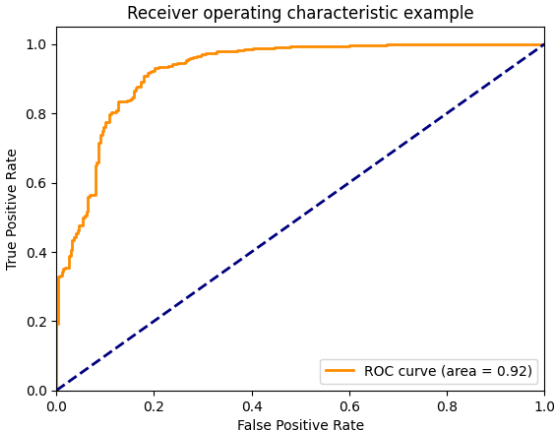


Figure D.9: ROC curve for ResNet50 in pneumonia detection

D.2.1.5 DenseNet121

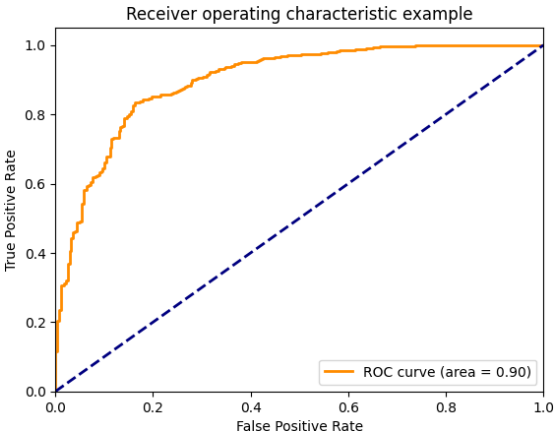


Figure D.10: ROC curve for DenseNet121 in pneumonia detection

D.2.1.6 InceptionV3

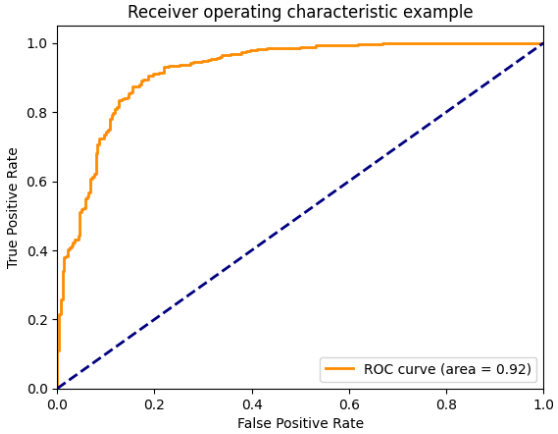


Figure D.11: ROC curve for InceptionV3 in pneumonia detection

D.2.2 AEs models

D.2.2.1 VAE

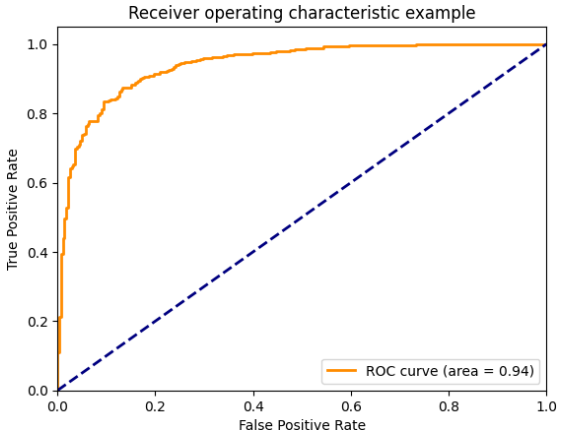


Figure D.12: ROC curve for VAE in pneumonia detection

D.2.2.2 CAE

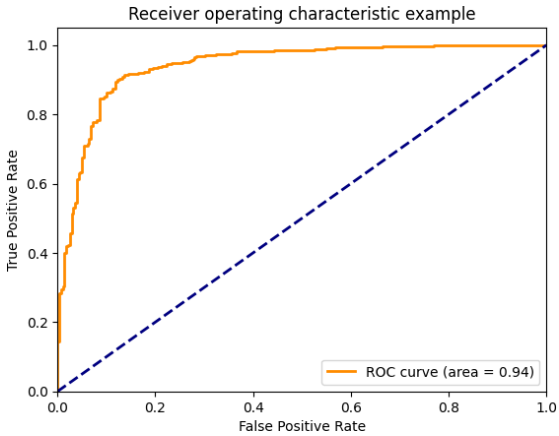


Figure D.13: Roc curve for CAE in pneumonia detection.

الملخص

يُعد تحليل الصور الطبية عنصرًا أساسيًا في تشخيص الأمراض، إلا أن تفسيرها لا يزال معقدًا. يتناول هذا العمل استخدام التعلم العميق، لأتمتة تصنيف الصور الطبية. وتمت دراسة حالتين: (Autoencoders) والمشفرات التلقائية (CNN) وخاصة الشبكات العصبية الالتفافية ، والكشف عن الالتهاب الرئوي من صور الأشعة السينية للصدر. وتُظهر (MRI) تصنيف أورام الدماغ باستخدام صور الرنين المغناطيسي .النتائج دقةً عالية في التصنيف، مما يؤكد إمكانات هذه الأساليب في تطوير أنظمة التشخيص الطبي بمساعدة الحاسوب

الكلمات المفتاحية

التعلم العميق، تحليل الصور الطبية، الشبكات العصبية الالتفافية، المشفرات التلقائية، المشفرات التلقائية التباينية، المشفرات التلقائية الالتفافية، التعلم بالنقل، تصنيف أورام الدماغ، كشف الالتهاب الرئوي، التشخيص بمساعدة الحاسوب.

.

Resumè :

L’analyse des images médicales est essentielle pour le diagnostic, mais son interprétation reste complexe. Ce mémoire étudie l’utilisation de l’apprentissage profond, notamment les CNN et les autoencodeurs, pour automatiser la classification des images médicales. Deux cas d’étude sont abordés : la classification des tumeurs cérébrales sur des images IRM et la détection de la pneumonie sur des radiographies thoraciques. Les résultats obtenus démontrent une bonne précision de classification et confirment le potentiel de ces approches pour le développement de systèmes de diagnostic assisté par ordinateur.

Mots-clés : Apprentissage profond, Images médicales, CNN, Autoencodeurs, VAE, CAE, Apprentissage par transfert, Tumeurs cérébrales, Pneumonie, Diagnostic assisté par ordinateur.

Abstract :

Medical image analysis is essential for disease diagnosis, but its interpretation remains challenging. This thesis investigates the use of deep learning, particularly Convolutional Neural Networks (CNNs) and autoencoders, to automate medical image classification. Two case studies are presented: brain tumor classification from MRI images and pneumonia detection from chest X-ray images. The obtained results demonstrate high classification accuracy and confirm the potential of these approaches for developing computer-aided diagnosis systems.

Keywords: Deep Learning, Medical Image Analysis, Convolutional Neural Networks, Autoencoders,Variational Autoencoders, Convolutional Autoencoders, Transfer Learning, Brain Tumor Classification, Pneumonia Detection, Computer-Aided Diagnosis.