



# AI-based Skin Cancer Detection Algorithms: Challenges, Countermeasures and Way Forward

By

**Tayyaba Arooj**

(1603-2020)

**Mahjabeen Sadeeq**

(1614-2020)

**2024**

Department of Computing

**Faculty of Engineering Sciences and Technology**  
Hamdard University Islamabad Campus, Pakistan



# AI-based Skin Cancer Detection Algorithms: Challenges, Countermeasures and Way Forward

By

**Tayyaba Arooj**

(1603-2020)

**Mahjabeen Sadeeq**

(1614-2020)

Under the supervision of

**Dr. SHAHEER Muhammad**

**2024**

**Faculty of Engineering Sciences and Technology**

Hamdard University Islamabad Campus, Pakistan

# **AI-based Skin Cancer Detection Algorithms: Challenges, Countermeasures and Way Forward**

By

**Tayyaba Arooj**

(1603-2020)

**Mahjabeen Sadeeq**

(1614-2020)

A project presented to the

**Faculty of Engineering Sciences and Technology**

In partial fulfillment of the  
requirements for the degree of

Bachelors of Science

In

Computer Science

**Faculty of Engineering Sciences and Technology**

Hamdard University Islamabad Campus, Pakistan



## **Faculty of Engineering Sciences and Technology**

Hamdard University Islamabad Campus, Pakistan

### **CERTIFICATE**

This project "**AI-based Skin Cancer Detection Algorithms: Challenges, Countermeasures and Way Forward**" presented by **Tayyaba Arooj** and **Mahjabeen Sadeeq** under the direction of their project advisor, Dr. SHAHEER Muhammad, and approved by the project examination committee, has been presented to and accepted by the Faculty of Engineering Sciences and Technology, in partial fulfillment of the requirements for Bachelor of Science in Computer Science.

---

Dr. SHAHEER Muhammad

(Project Advisor)

---

Dr. Tahir Saleem

(Examiner 1)

---

Ms. Sakha

(Examiner 2)

---

Dr. Hannan Adeel

(Chairman, Department of Computing)

---

Dr. Hassan Raza

(Associate Dean, FEST)



# **Abstract**

Skin cancer, particularly melanoma, is a significant global health concern. Early and accurate detection is critical for improving patient outcomes. This study conducts an in-depth literature review to identify commonly used CNN variants, datasets, and key evaluation metrics to assess their performance in classifying benign and malignant skin lesions. Furthermore, we implemented commonly used convolutional neural networks (CNN) including ResNet, EfficientNet, DenseNet, AlexNet, VGG, GoogleNet, LeNet-5, Xception and MobileNet and evaluated their performance in classifying benign and malignant skin lesions using publicly available dataset on Kaggle. Additionally, a comparative analysis is conducted based on identified metrics such as accuracy, precision, recall, sensitivity and F1-score, highlighting the strengths and limitations of each algorithm. Our results demonstrate that VGG emerged as the best performer with an accuracy of 97metrics which provides critical insights for deploying AI-based skin cancer detection systems in clinical practice.

# **Dedication**

We dedicate this thesis to our families, whose constant support, guidance, and encouragement have been a source of strength and inspiration throughout this journey.

# Acknowledgments

We extend our deepest gratitude to our supervisor, Dr. SHAHEER Muhammad, whose unwavering support, insightful critiques, and dedication to academic excellence have been pivotal throughout our research journey. His guidance and meticulous attention to detail have significantly shaped this thesis.

We also wish to thank the Chairman of the Department of Computing, Dr. Hannan Adeel, for his invaluable feedback and essential suggestions, which have enhanced the quality of our work.

Our sincere appreciation goes to the faculty and staff in the Department of Computing at Hamdard University, Islamabad Campus, whose resources and assistance have been invaluable.

We would also like to thank our families for their unwavering support and encouragement, which motivated us throughout this journey.

Finally, we are grateful to everyone who, directly or indirectly, contributed to the successful completion of this project.

Tayyaba Arooj  
Mahjabeen Sadeq

# Project Brief

<b>Project name:</b>	AI Based Skin Cancer Detection Algorithms: Challenges, Countermeasures and Way Forward
<b>Objective:</b>	To evaluate and recommend AI-based skin cancer detection algorithms using key performance metrics
<b>Undertaken by:</b>	Tayyaba Arooj, Mahjabeen Sadeeq
<b>Supervised by:</b>	Dr. Shaheer Muhammad
<b>Date started:</b>	October, 2023
<b>Date completed:</b>	October, 2024
<b>Tools used:</b>	Microsoft Office, Google Colab, Kaggle, Overleaf, LaTeX
<b>System used:</b>	Dell Latitude E5570, Intel Core i5, 16GB RAM, 256GB SSD, Windows 10 Pro

# Contents

<b>Abstract</b>	<b>i</b>
<b>List of Tables</b>	<b>viii</b>
<b>List of Figures</b>	<b>x</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction . . . . .	2
1.1.1 Problem Statement . . . . .	3
1.1.2 Organization of the Thesis . . . . .	3
<b>2 Literature Review</b>	<b>4</b>
2.0.1 Performance Evaluation of CNN Architectures Based on Literature Review . . . . .	10
<b>3 System Background</b>	<b>12</b>
3.1 Image Preprocessing . . . . .	13
3.1.1 Image Normalization . . . . .	13
3.1.2 Image Augmentation . . . . .	15
3.1.3 Image Resizing . . . . .	16
3.1.4 Histogram Equalization . . . . .	18
3.2 Convolutional Neural Network . . . . .	19
3.2.1 VGG-16 . . . . .	21
3.2.2 VGG-19 . . . . .	23
3.2.3 AlexNet . . . . .	24

3.2.4	LeNet-5 . . . . .	25
3.2.5	GoogleNet . . . . .	26
3.2.6	ResNet . . . . .	27
3.2.7	MobileNet . . . . .	29
3.2.8	Xception . . . . .	31
3.2.9	DenseNet . . . . .	32
3.2.10	EfficientNet . . . . .	33
3.3	Evaluation Metrics . . . . .	35
3.3.1	Accuracy . . . . .	36
3.3.2	Precision . . . . .	36
3.3.3	Recall (Sensitivity or True Positive Rate) . . . . .	36
3.3.4	Specificity . . . . .	37
3.3.5	F1-score . . . . .	37
3.3.6	False Positive Rate (FPR) . . . . .	37
3.3.7	False Negative Rate (FNR) . . . . .	38
3.3.8	False discovery Rate (FDR) . . . . .	38
3.3.9	Area under the curve (AUC) . . . . .	38
3.3.10	Time Complexity . . . . .	38
3.4	Time Complexity of CNN . . . . .	39
3.5	Time complexity of VGG-16 . . . . .	39
3.6	Time complexity of VGG-19 . . . . .	39
3.7	Time complexity of AlexNet . . . . .	39
3.8	Time complexity of Mobilenet-v2 . . . . .	40
3.9	Time complexity of Xception . . . . .	40
3.10	Time complexity of ResNet . . . . .	41
3.11	Time complexity of GoogleNet . . . . .	41
3.12	Time complexity of EfficientNet . . . . .	41
3.13	Time complexity of DenseNet . . . . .	41
3.14	Time complexity of Lenet-5 . . . . .	42
<b>4</b>	<b>Methodology</b> . . . . .	<b>43</b>
4.1	Dataset Collection . . . . .	44
4.1.1	HAM10000 . . . . .	44

4.1.2	ISIC-2017 . . . . .	44
4.1.3	ISIC-2018 . . . . .	45
4.1.4	PH2 . . . . .	45
4.2	Preprocessing . . . . .	46
4.3	Algorithm Selection . . . . .	47
4.4	Algorithm Implementation . . . . .	47
4.5	Identifying Performance Metrics . . . . .	47
4.6	Algorithm Evaluation . . . . .	48
4.7	Recommendation . . . . .	48
<b>5</b>	<b>Results and Discussion</b>	<b>49</b>
5.1	Accuracy . . . . .	51
5.2	Precision . . . . .	52
5.3	Recall . . . . .	53
5.4	F1-Score . . . . .	54
5.5	Specificity . . . . .	55
5.6	False Positive Rate (FPR) . . . . .	56
5.7	False Negative Rate (FNR) . . . . .	57
5.8	False Discovery Rate (FDR) . . . . .	58
5.9	Area Under the Curve (AUC) . . . . .	59
<b>6</b>	<b>Conclusion</b>	<b>62</b>

# List of Tables

2.1	Performance Comparison of CNN Architectures . . . . .	11
2.2	Performance Comparison of CNN Architectures . . . . .	11
4.1	Datasets commonly used in literature for AI-based skin cancer detection	45
4.2	Dataset . . . . .	46
5.1	Performance Comparison of CNN Architectures . . . . .	50
5.2	Performance Comparison of CNN Architectures . . . . .	50

# List of Figures

3.1	Convolutional Neural Network . . . . .	19
3.2	Working of CNN . . . . .	21
3.3	Architecture of VGG-16 . . . . .	22
3.4	Architecture of VGG-19 . . . . .	23
3.5	Architecture of AlexNet . . . . .	24
3.6	Architecture of LeNet-5 . . . . .	25
3.7	Architecture of GoogleNet . . . . .	27
3.8	Architecture of ResNet . . . . .	28
3.9	Residual blocks . . . . .	28
3.10	Architecture of MobileNet . . . . .	30
3.11	Architecture of Xception . . . . .	31
3.12	Architecture of DenseNet . . . . .	33
3.13	Architecture of EfficientNet . . . . .	34
4.1	Benign and Malignant images from ISIC archive . . . . .	46
4.2	Proposed Approach Schematic . . . . .	48
5.1	Accuracy of CNN models . . . . .	51
5.2	Precision of CNN models . . . . .	52
5.3	Recall of CNN models . . . . .	53
5.4	F1-Score of CNN models . . . . .	54
5.5	Specificity of CNN models . . . . .	55
5.6	FPR of CNN models . . . . .	56
5.7	FNR of CNN models . . . . .	57
5.8	FDR of CNN models . . . . .	58

5.9	AUC of CNN models . . . . .	59
5.10	Comparison of CNN models based on performance metrics . . . . .	60

# **Chapter 1**

## **Introduction**

## 1.1 Introduction

Cancer emerges when healthy cells undergo abnormal changes, leading to uncontrolled growth and the formation of tumors [1]. These tumors can be categorized as either Benign or Malignant. Malignant tumors are characterized by their ability to grow and spread to other parts of the body, posing a significant health risk [2]. Skin cancer is a widespread and potentially life-threatening condition that affects millions of individuals globally [3]. Timely and accurate diagnosis is critical for effective treatment and improved patient outcomes [1]. Traditional diagnosis methods often rely on subjective visual assessment by dermatologists, which can be time-consuming, expensive and prone to variability [4]. Many regions lack access to skilled dermatologists, resulting in delayed or missed diagnoses, which can negatively impact patient prognosis and increase treatment costs [5]. Skin cancer detection refers to techniques used to detect cancer using skin lesions. AI has been used to develop various algorithms and techniques for skin cancer detection that increase accuracy and improve patient diagnostic outcomes and reduce the burden on healthcare systems[6].

Advancements in machine learning and deep learning have revolutionized the field of medical imaging and diagnostics and since there are many algorithms that have been developed for skin cancer detection [7] each employing varying techniques of implementation, so it's crucial for the industry to know the pros and cons of these algorithms so that appropriate algorithm can be chosen as per their requirement. Some popular deep learning algorithms used in skin cancer detection are ResNet, GoogLeNet, VggNet, Xception, InceptionNet,etc. Evaluating the implementation of these algorithms is a big challenge. Therefore, we will identify metrics for evaluation of these algorithms. Accuracy, Precision and Recall are widely used metrics for evaluation in the state-of-the-art [8]-[9].

This study focuses on the survey of AI-based skin cancer detection algorithms. We will conduct an in-depth literature review to explore the state-of-the-art and will proceed to implement a selection of algorithms which are commonly used for AI-based skin cancer detection. Eventually, we will evaluate these algorithms based on their performance in terms of identified metrics. By systematically comparing these techniques, we seek to provide recommendations for their practical application in clinical settings, thereby facilitating the adoption of AI-driven diagnostic tools to improve patient outcomes and

optimize healthcare resources.

### **1.1.1 Problem Statement**

Despite recent advancements in AI-based skin cancer detection algorithms, there remains a need to systematically evaluate these algorithms based on some metrics (e.g. Accuracy, Specificity). Since it's crucial for the industry to know the pros and cons of these algorithms, so that appropriate algorithm can be chosen as per their requirement, so this study seeks to address this gap in the field by conducting a comprehensive survey of AI algorithms for skin cancer detection, identifying evaluation metrics, defining an evaluation mechanism, analyzing the algorithm based on the evaluation mechanism and providing recommendations for industry and users according to their requirements, ultimately contributing to improved diagnostic accuracy, cost-effectiveness, and patient outcomes.

### **1.1.2 Organization of the Thesis**

The organization of this thesis is organized as follows: A comprehensive literature review is conducted to investigate the latest advancements in AI-based skin cancer detection. Based on this review, a selection of state-of-the-art algorithms and key metrics is made based on their relevance and popularity in literature. Identified evaluation metrics are used to assess the selected algorithms' performance. Following their implementation, a detailed analysis is conducted to evaluate and compare results. Finally, the findings are thoroughly documented for future reference and research.

# **Chapter 2**

## **Literature Review**

In this chapter, an extensive literature review of various studies on AI-based skin cancer detection and classification is provided, presenting a comprehensive overview of the current state-of-the-art in AI-based skin cancer detection..

Several techniques are used for classification of skin cancer using skin lesion images in state of the art, one such study proposed a novel approach for melanoma skin cancer detection using a hybrid feature extractor (HFF) combining HOG, LBP, SURF, and VGG-19 based CNN techniques in [1]. Furthermore, they combined a hybrid feature extractor (HFE) and a VGG-19-based convolutional neural network (CNN) feature extractor for classification. This study used HAM10000 dataset for evaluation of the proposed model. In addition to this, they used accuracy, precision, specificity, and sensitivity as their performance metrics. The proposed model (HFF+CNN) achieved accuracy of 99.4%.

Another study has done a comparative analysis of different AI-based algorithms [2]. It evaluates VGG-16, Support Vector Machine (SVM), ResNet50, and self-built sequential models with differing layers. The results show VGG-16 has achieved the highest accuracy at 93.18%. However, it could have been compared with other CNN models (such as InceptionV3, MobileNet, etc.) as well.

A hybrid approach with combination of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) for skin lesion detection is proposed in [3]. They used data from HAM10000 dataset. Results of this study show that they achieved a classification accuracy of 94% for CNN and 97.3% for RNN. Although the combination of CNN and RNN shows good results in enhancing early skin disease detection, however, there remains a need to compare the proposed models with other popular techniques to assess its overall performance and effectiveness.

An artificial intelligence-assisted detection model for melanoma diagnosis using deep learning has been proposed in [10]. They used AlexNet, MobileNet, ResNet, VGG-16, and VGG-19 algorithms which were evaluated on the dataset from Kaggle that contains 8598 images. Furthermore, the evaluation metrics used to assess the performance of models include accuracy, F1-score, precision, specificity, F1-score. The results show that MobileNet has achieved the highest accuracy of 84.94%. However, it needs further validation on larger and different datasets.

A novel Parallel CNN Model using deep learning for skin cancer detection and classification is proposed in [11]. They used dataset of 25,780 images from Kaggle consisting of nine classes of skin cancer. Their results show the proposed model outperforms

VGG-16 and VGG-19 models by achieving precision of 76.17%, recall of 78.15%, and F1-score of 76.92%. However, there remains a need for further validation on larger and different datasets to ensure robustness of the proposed model.

A novel web application named DUNEScan (Deep Uncertainty Estimation for Skin Cancer) is presented in [12]. This web application features six CNN models which includes Efficient Net, Inceptionv3, ResNet50, MobileNetv2, BYOL and SwAV. Moreover, this application was tested on HAM10000 dataset. This web application allows uploading skin lesions images on which it then applies different CNN models. However, this model doesn't predict accurate results and it still relies on dermatologists' diagnosis.

A model for diagnosing melanomas using deep learning is proposed in [13]. Inception-V3 and InceptionResnet-V2 were used for melanoma recognition. The study used HAM1000 dataset for the evaluation of proposed models. The study employs enhanced super-resolution generative adversarial network (ESRGAN) using 10,000 training photos to generate highquality images for the Human against Machine dataset. Moreover, they concluded that the proposed models outperformed the results of the previous investigation with an effectiveness of 0.89 for Inception-V3 and 0.91 for InceptionResnet-V2. However, it is important to evaluate the models on additional datasets to verify the consistency of the results.

A novel method for the automated detection of melanoma lesions via using a deep-learning technique has been proposed in [14]. They've combined faster region-based convolutional neural networks (RCNN) with fuzzy k-means clustering (FKM)clustering and performance has been evaluated on ISBI-2016, ISIC-2017, and PH2 datasets. Moreover, 95.40, 93.1, and 95.6% accuracy has been achieved on the ISIC-2016, ISIC-2017, and PH2 datasets, respectively which concludes that it outperforms the current state of the art.

Skin cancer segmentation model is proposed in [6] using the deep learning algorithm called Feature Pyramid Network (FPN). They used ResNet34, DenseNet121, and MobileNet-v2 for segmentation and DenseNet121 is used for classification. Furthermore, the dataset they used for evaluation is HAM10000 consisting of 10,015 images of seven classes. The result shows that the proposed methodology achieved 80% accuracy with ResNet34, 70% with DenseNet121, and 75% with MobileNetv2 in segmentation, and 80% accuracy in classification.

A hybrid approach using deep learning and classical machine learning techniques is described to detect melanoma skin cancer [15]. The dataset consists of 640 skin lesion images from ISIC archive that were used for evaluation. Their system relied on the prediction of three different methods which includes KNN, SVM and CNN trained with a set of features achieved accuracy of 57.3%, 71.8%, 85.5%, respectively. The prediction of these 3 methods were combined using majority voting and it achieved accuracy of 88.4%, it shows that using the hybrid approach gives the highest accuracy level. However, this study could have used more skin lesion images as it requires huge data to effectively train the model.

With advancement in AI, many tools have emerged that can help in diagnosis and classification of dermatological images. To systematically review these advancements, one such study [8] evaluated different AI methods used for skin cancer detection and classification. This study analyzed 18 papers related to skin cancer detection and classification. It shows that the popular datasets used in the studies include HAM10000 and ISIC and the common metrics include accuracy, sensitivity, and specificity. Additionally, the results show that CNNs, especially ResNet, has high performance. However, there remains a need to provide a comparative analysis of the algorithms based on different algorithms.

An automatic diagnosis of skin cancer using Deep Convolutional Neural Network (DCNN) has been proposed by using deep learning with Deep CNN and machine learning with Naive Bayes and Random Forest [16]. Moreover, International Skin Imaging Collaboration (ISIC) dataset consisting of 3297 images of benign and malignant has been used in this study to evaluate the proposed system. Their proposed system outperforms past literature work and has achieved accuracy of 99.5%. However, there remains a need to further test the proposed method with different datasets. Additionally, they used 200 epochs for training the DCNN which is quite time consuming. A comprehensive review of the automated skin cancer identification processes with deep learning and machine learning approaches to date has been provided in [17]. This survey covers preprocessing, segmentation, feature extraction, selection, and classification methods for recognizing skin cancer. Additionally, the survey shows that CNN outperforms the traditional methods while classifying image samples and segmentation. However, this survey only used accuracy as evaluation metric. A hybrid deep learning model by combining VGG-16 and ResNet50 is proposed to classify skin lesions in [18]. Moreover, they also em-

ployed various other deep learning models and machine learning techniques including Densenet121, VGG-16 SVM, and KNN. The dataset of 3000 images of nine different classes is used for evaluation. The proposed hybrid model achieved a training accuracy of 98.75%, a validation accuracy of 97.50%, a precision of 97.60%, a recall of 97.55%, and an F1 score of 97.58%. However, this study could have evaluated the proposed model on different datasets as well.

The systematic review has been done in [19] to explore the types of algorithms applied to detect skin cancer, types of optimizers applied to improve accuracy, the popular data sets being used in studies, and the metrics used to validate the algorithms. This systematic review shows that HAM10000, ISIC, and PH2 datasets are widely used by different studies to validate their models. Moreover, the popular evaluation metrics used in studies include accuracy, precision, recall (sensitivity) and specificity. It also shows that CNN particularly Resnet outperforms other algorithms, especially when used in hybrid models.

A methodology to develop an Android application that utilizes the MobileNet v2 architecture has been proposed in [20]. They used a diverse dataset of skin lesion images, including both melanoma and non-melanoma cases, to enhance the accuracy of skin cancer detection. Their proposed methodology achieved an accuracy of 91.346. However, the proposed model's performance could be investigated further on larger dataset. A deep learning-based methodology for the classification of skin cancer lesions by utilizing transfer learning with CNN variants such as InceptionV3, MobileNetV2, and DenseNet201 has been proposed in [21]. They used the HAM10000 and ISIC 2017 datasets for the evaluation, which included a total of 3,297 images of 2 classes i-e, malignant and benign. Additionally, the proposed methodology achieved accuracy of 95.5%. Additionally, they used Grad-CAM visualization which enhances the interpretability of the model's predictions. However, optimization strategies could be used to further improve the results.

Skin cancer classification method using deep learning and transfer learning using the AlexNet architecture has been propose din [22]. The proposed methodology replaced the last layer of AlexNet with a softmax layer for classification and fine-tuned the network with data augmentation. Moreover, the proposed model is trained on the PH2 dataset, which has images of 3 classes i-e melanoma, common nevus, and atypical nevus. Additionally, the proposed method achieved 98.61%, 98.33%, 98.93% and 97.73%

of accuracy, sensitivity, specificity, and precision respectively.

A novel convolutional neural network (CNN) model to improve skin cancer detection on mobile platforms is proposed in [23]. They trained a compact CNN from scratch on a balanced dataset using advanced regularization techniques such as dropout and data augmentation. The model was evaluated using the PHDB melanoma dataset which consists of high-resolution skin lesion images. The proposed model achieved an accuracy of 86%. However, more datasets such as HAM10000 and Dermofit could be incorporated to achieve even higher accuracy.

A novel multiclassification framework for skin cancer detection by using a combination of Xception and ResNet101 deep learning models (XR101) has been proposed in [24]. This methodology use the strengths of both Xception and ResNet101 for feature extraction and classification of various skin cancer types. They evaluated the proposed algorithm on three public datasets i-e PH2, DermPK, and HAM10000, they also focused on balancing class distributions with the Borderline-SMOTE technique. The proposed model achieved an accuracy of 98.21%.

Deep learning approach for melanoma skin cancer detection using various CNN architectures is proposed in [25]. They trained models on a dataset of around 36,000 images from public sources such as SIIM-ISIC and they also applied data augmentation to improve the performance of their model. They tested VGG, ResNet, EfficientNet, and DenseNet while EfficientNetB7 outperformed by achieving an accuracy of 99.33%, sensitivity of 98.78%, and AUC of 99.01%. However, the study could have used some techniques to handle the imbalance problem in the datasets.

A transfer learning approach using the GoogLeNet architecture for classifying various skin lesions has been proposed in [26]. Their methodology include preprocessing images, data augmentation, and using pretrained models for classification. Moreover, they trained the model on the ISIC dataset which has nine types of skin cancer images. It achieved accuracy of 89.93%, precision of 0.785, recall of 0.687, and F1-score of 0.733. Systematic review of deep learning methods for melanoma classification has been done in [27]. They identified a total of 5112 studies , and out of 5112 studies they selected 55 well-reputed studies. This study shows that the most commonly used datasets in literature are PH2, ISIC (2016, 2017, 2018) and DermIS. Furthermore, the algorithms for skin cancer detection that are commonly used include AlexNet, VGG-16, ResNet, Inception-v3, and DenseNet. In this study, they also highlighted that ensemble deep

learning methods and pre-trained CNN models can show promising results. Additionally, this study shows that ensemble methods that combine models like ResNet and DenseNet have achieved highest classification accuracy.

A novel deep learning model called SkinNet-16, based on convolutional neural network (CNN) architecture, is proposed in [28] for the classification of benign and malignant skin lesions. Their proposed methodology includes pre-processing pipeline that has various pre-processing steps such as digital hair removal, background noise reduction and various filtering techniques such as non-local means de-noising, Gaussian filtering. Moreover, they used Principal Component Analysis (PCA) for dimensionality reduction. ISIC and HAM10000 datasets were used for evaluation of SkinNet-16. They achieved accuracy of 99.19% in skin cancer detection.

A comprehensive review of deep learning techniques for early detection of skin cancer has been done in [29]. They provided the overview of skin diseases, including their types, datasets, and data pre-processing techniques, deep learning approaches and popular methods used in literature for skin cancer detection. They compared the studies that include Convolutional Neural Networks (CNN), Artificial Neural Networks (ANN), and k-Nearest Neighbors (KNN) for skin cancer detection. Addititonally, the study conclude that the most commonly used datasets include HAM10000, PH2, ISIC archive, DermIS, Dermnet, and DermQuest.

### **2.0.1 Performance Evaluation of CNN Architectures Based on Literature Review**

The table 2.1 and 2.2 presents the performance comparison of various Convolutional Neural Network (CNN) architectures based on literature reviews. The architectures are evaluated on several metrics, including accuracy, precision, recall, F1-score, specificity, false positive rate (FPR), false discovery rate (FDR), false negative rate (FNR), and area under the curve (AUC).

Table 2.1: Performance Comparison of CNN Architectures

<b>Architecture</b>	<b>Accuracy</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>
VGG-16	90% [30]	91% [30]	90% [30]	90% [30]
VGG-19	97% (25 epochs) [31]	86% [32]	85% [32]	80% [32]
ALEXNET	99% [33]	99% [34]	93% [34]	96% [34]
MOBILENET-V2	94% [34]	98% [34]	90% [34]	94% [34]
XCEPTION	93% [30]	89% [30]	93% [30]	93% [30]
LENET-5	89.7% [30]	89.5% [30]	89.5% [30]	89.5% [30]
RESNET	69% [35]	92% [35]	82.3% [35]	90% [36]
DENSENET	89% [37]	90% [38]	88% [38]	88.94% [38]
EFFICIENTNET	98% [39]	86% [39]	86% [39]	85% [39]
GOOGLENET	93% [40]	98% [26]	96% [40]	97% [40]

Table 2.2: Performance Comparison of CNN Architectures

<b>Architecture</b>	<b>Specificity</b>	<b>FPR</b>	<b>FDR</b>	<b>FNR</b>	<b>AUC</b>
VGG-16	86.6% [41]	8.7% [42]	47% [42]	50.6% [42]	81% [42]
VGG-19	88% [43]	65.8% [44]	28% [44]	11% [44]	84% [43]
ALEXNET	75.9% [45]	24% [45]	23.6% [45]	9.8% [45]	92% [46]
MOBILENET-v2	97% [47]	48% [44]	23.5% [44]	16.5% [44]	95% [46]
XCEPTION	44.9% [44]	55% [44]	25% [44]	12% [44]	96%[46]
LENET-5	89% [30]	10.4% [30]	10.3% [30]	10.3% [30]	89% [30]
RESNET	100% [48]	4% [49]	4% [49]	3% [49]	73% [50]
DENSENET	92% [51]	19% [52]	22% [52]	34% [52]	74% [53]
EFFICIENTNET	91% [39]	5% [54]	50% [55]	7% [54]	77% [53]
GOOGLENET	100% [33]	22% [52]	33% [52]	28% [52]	92% [56]

# **Chapter 3**

## **System Background**

In this chapter, we explore the foundational processes and methodologies for AI-based skin cancer detection. The chapter begins with an overview of essential image pre-processing techniques, including normalization, augmentation, resizing, and histogram equalization, which enhance the quality and consistency of input images. This is followed by a detailed examination of Convolutional Neural Networks (CNNs) and its architectures such as VGG-16, AlexNet, and ResNet. We then discuss key evaluation metrics (including accuracy, precision, recall, specificity, etc) that assess the performance of these models. Finally, the time complexity of each CNN variant is reviewed to highlight computational efficiency in real-world applications.

### **3.1 Image Preprocessing**

Preprocessing is a critical step in preparing image data for machine learning models, especially in skin lesions classification or detection tasks. It involves techniques like augmentation to artificially expand the dataset, resizing to maintain uniform input dimensions, and normalization to standardize pixel values.

#### **3.1.1 Image Normalization**

Image normalization is an important preprocessing step [57], especially for tasks like skin lesion detection in medical imaging. The goal of normalization is to standardize the pixel intensity values across images [58] which improves the model generalization. Normalization makes the model more robust to variations in the input data, leading to better generalization on unseen data [59]. There are various techniques used for image normalization including Min-Max Normalization, Z-Score Normalization (Standardization), and Decimal Normalization [60].

#### **Min-Max Normalization**

Min-Max normalization rescales pixel values to a fixed range like [0, 1] or [-1, 1], this ensures that all pixel values are on the same scale, though it can be sensitive to outliers. Mathematical formula given below [61]

$$A' = \left( \frac{A - \text{min value of } A}{\text{max value of } A - \text{min value of } A} \right) \cdot (D - C) + C \quad (3.1)$$

where,  $A'$  contains Min-Max Normalized data one,

If pre defined boundary is  $[C, D]$

If  $A$  is the range of the original data and  $B$  is the mapped data [60]

### Z-score Normalization

Z-score normalization, or standardization, transforms pixel values so that they have a mean of 0 and a standard deviation of 1. This technique is useful when pixel intensity values need to be standardized across different images or modalities. This can reduce the influence of outliers by centering the data. Z-score normalization is given as [60] :

$$v'_i = \frac{v_i - \bar{E}}{\text{std}(E)} \quad (3.2)$$

where,

$v'$  is the Z-score normalized value.

$v_i$  is the value of the row  $E$  in the  $i$ th column.

$$\text{std}(E) = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (v_i - \bar{E})^2}$$

$$\bar{E} = \frac{1}{n} \sum_{i=1}^n v_i \quad (\text{mean value})$$

### Decimal Scaling

Decimal scaling gives the range between -1 and 1 [60]. In Decimal scaling the original data is divided by a power of 10, determined by the number of digits in the largest absolute value of the data. The main goal is to bring the data into a normalized range where all values are less than 1 in magnitude. Decimal Scaling is given as [60]:

$$v^i = \frac{v}{10^j} \quad (3.3)$$

where,

$v^j$  is the scaled values

$v$  is the range of values

$j$  is the smallest integer such that  $\max(|v_i|) < 1$

Additionally, using batch normalization during training, which normalizes the activations of intermediate layers that can help in stabilizing and accelerating the training process [61].

### 3.1.2 Image Augmentation

Image augmentation is a technique used in computer vision to artificially increase the diversity of a training dataset by creating modified versions of images [62]. This process involves applying various transformations to the original images, such as rotation, flipping, scaling, noise injections and color adjustments, etc. without altering the underlying content [63]. Furthermore, Image augmentation is important in training deep learning models especially when dealing with limited or imbalanced datasets, as it helps to improve the model's generalization ability and robustness [64].

It addresses the issue of data scarcity [65] as it generates larger and more varied dataset from a limited number of available images. Additionally, when the classes of datasets are imbalanced [66], like some classes are overrepresented while others are underrepresented. Augmentation can help mitigate this imbalance by generating more examples of the underrepresented classes. There are various augmentation techniques based on basic image manipulation, for instance, geometric transformations like rotation, flipping, translation, cropping, etc. [67].

Translation refers to shifting an image from one position to another in the plane, this operation moves every pixel of the image either in the horizontal, vertical, or both directions [67]. Flipping involves mirroring an image either horizontally or vertically. Horizontal flipping is commonly used in image augmentation [67] for tasks like object detection and classification.

Photometric transformations also known as color space transformation [67] are equally significant. Color space represents the range of colors that can be shown in an image. Common color spaces include RGB (Red, Green, Blue), CMYK (Cyan, Magenta, Yel-

low, Black), and HSV (Hue, Saturation, Value). Changing color spaces is often done to enhance certain features or for tasks like color-based segmentation. To adjust an image that's too bright or too dark, we can loop through the image and modify each pixel's brightness by adding or subtracting a fixed value [67]. Another easy way to manipulate color is by isolating the red, green, or blue layers of the image [67]. Additionally, you can apply a transformation to ensure that pixel values don't go beyond a specific minimum or maximum limit, keeping the colors within a controlled range [67]. Both Geometric and Photometric transformation have disadvantages like increase memory and training time [67].

Kernel filters are commonly used to alter images in various ways, such as blurring, sharpening, or detecting edges [68]. These filters use a small matrix, called a kernel, that slides across the entire image [67]. Gaussian blur is another technique used in image augmentation. By applying a Gaussian filter, which gives more weight to pixels closer to the center of the kernel and less weight to those farther away [69], you can simulate depth or reduce noise in the image.

It works by applying a Gaussian function to the image, smoothing the pixel intensities and producing a blur effect and 2-Dimensional Gaussian function is the result of product of 2 1-Dimensional Gaussian functions [70]. The Gaussian function used for blurring is given by the equation [70]:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.4)$$

where,

$G(x, y)$  is two-dimensional Gaussian function and  $(x, y)$  are coordinates and  $\sigma$  is the standard deviation of the Gaussian distribution [70].

Image augmentation techniques help in expanding the training dataset, improving the model's generalization ability, and ensuring robust performance across a wide range of imaging conditions.

### 3.1.3 Image Resizing

Image resizing ensures that all images in a dataset have the same dimensions [71], which is crucial for batch processing and consistent input to the model [72]. Skin lesions may

vary in size so it's important to resize them. Moreover, Neural networks often expect fixed-size inputs so it is very important to resize the images [73]. Additionally, Pre-trained models, such as those based on architectures like ResNet or VGG, are trained on datasets with specific image sizes (such as ImageNet with size of 224x224 pixels) [74]. There are several methods used for resizing images, each with different levels of interpolation and interpolation is a way by which images are scaled [75]. Furthermore, it works in two directions i-e horizontally and vertically and approximate a pixel's color and intensity based on surrounding pixel values [75]. Interpolation techniques are mainly divided into two categories i-e Non-adaptive techniques and Adaptive techniques , however, we will only discuss a few non-adaptive techniques below.

### **Nearest Neighbor Interpolation**

The simplest method, where the pixel value at a new position is assigned the value of the nearest pixel in the original image [75]. It is fast but can produce blocky or pixelated results when it enlarges an image [75]. The nearest neighbor interpolation's kernel is given as [75]:

$$u(x) = \begin{cases} 0 & |x| > 0.5 \\ 1 & |x| < 0.5 \end{cases} \quad (3.5)$$

where  $u(x)$  is the interpolation kernel for nearest neighbor interpolation and  $x$  is distance between interpolated point and grid point [75].

### **Bilinear Interpolation**

This method considers the closest 2x2 pixel neighborhood in the original image and calculates the weighted average of these four pixels to determine the new pixel value [75]. It provides smoother results compared to nearest neighbor interpolation [75]. Bilinear interpolation's kernel is given as [75]:

$$u(x) = \begin{cases} 0 & |x| > 1 \\ 1 - |x| & |x| < 1 \end{cases} \quad (3.6)$$

where  $u(x)$  is Bilinear interpolation's kernel and  $x$  is distance between interpolated point and grid point

## Bicubic Interpolation

Bicubic interpolation considers the closest 4x4 pixel neighborhood (total of 16 pixels) and computes the weighted average using cubic polynomials [75]. It produces smoother and more natural-looking resized images as compared to bilinear interpolation however, it is computationally more expensive. Bicubic interpolation's kernel is given as [75]:

$$u(x) = \begin{cases} \frac{3}{2}|x|^3 - \frac{5}{2}|x|^2 + 1 & 0 \leq |x| < 1 \\ -\frac{1}{2}|x|^3 + \frac{5}{2}|x|^2 - 4|x| + 2 & 1 \leq |x| < 2 \\ 0 & 2 < |x| \end{cases} \quad (3.7)$$

where  $u(x)$  is Bicubic interpolation's kernel and  $x$  is distance between interpolated point and grid point.

Resizing ensures that lesion images of varying sizes are normalized to a consistent input size, which is essential for deep learning models to process the images in batches.

### 3.1.4 Histogram Equalization

Histogram equalization is a technique used in image processing to improve the contrast of an image [76]. It does this by spreading out the most frequent intensity values [76], it effectively makes the dark areas darker and the light areas lighter, which can enhance the visibility of features in an image. The histogram of an image is a graphical representation of the distribution of pixel intensities (brightness values). It plots the number of pixels (frequency) for each intensity level.

#### Steps for Histogram equalization:

- 1. Compute the Histogram:** Firstly, calculate the histogram of the image, which gives the frequency of each intensity level in the image [76].

$$p_r(r_k) = \frac{n_k}{n}, \quad k = 0, 1, 2, \dots, L-1 \quad (3.8)$$

where,  $p_r(r_k)$  represents the probability or normalized frequency of each intensity level  $r_k$ ,  $n$  is the total number of pixels,  $n_k$  is the number of pixels that have gray level  $r_k$ , and  $L$  is total number of possible gray level [76].

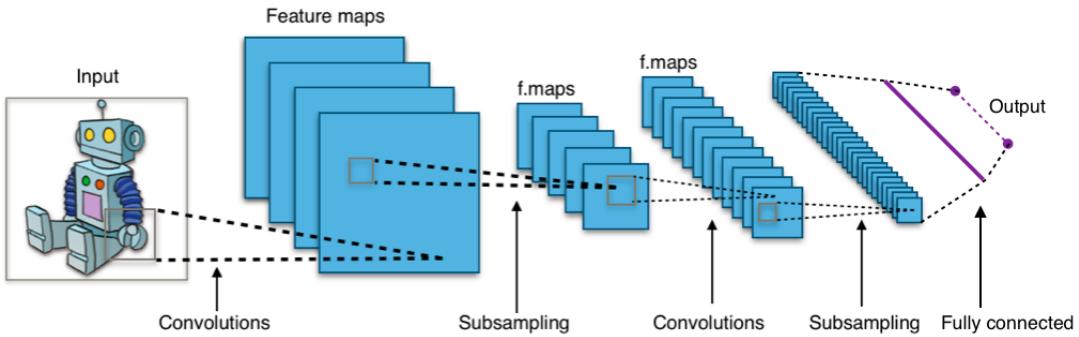


Figure 3.1: Convolutional Neural Network

**2. Calculate the Cumulative Distribution Function (CDF):** The CDF is computed from the histogram. The CDF at a particular intensity level is the sum of the histogram values for that intensity and all previous intensity levels. It gives a mapping from the original intensity levels to new levels based on the cumulative frequency [76].

**3. Normalize the CDF:** Divide each CDF value by the maximum value of the CDF. This scaling will ensure that the normalized CDF values are in the range [0-1][76].

**4. Map the Original Intensities:** Replace each pixel's original intensity with the corresponding intensity in the normalized CDF. This mapping effectively redistributes the intensity values, leading to a more uniform histogram and improved contrast [76]. Histogram equalization is a fundamental tool in image preprocessing and has many applications including medical imaging, satellite imaging and Object detection.

### 3.2 Convolutional Neural Network

A Convolutional Neural Network (CNN) is a type of deep learning model specifically designed for computer vision tasks [77]. It can easily process grid-like data, such as images [78]. Moreover, CNNs are widely used in image recognition, object detection, and other computer vision tasks due to their ability to automatically capture features from images [79]. CNN's structure is inspired by human and animal brains [80]. The core components of a CNN are convolutional layers, pooling layers, and fully connected layers (see figure 3.1 [81]) [82].

**Convolutional layers:** Convolutional layers are the first building blocks of a CNN, where the network applies filters (also known as kernels) to the input data. It takes an image as its input and then apply 3x3 or 5x5 filters on it [83]. These filters slide across the input image and multiplies its values with overlapping values of input image, and then it combines all these values to produce a single output for each overlapping region and it continues this process until the entire image has been processed [83]. The result of this operation is called a feature map. Moreover, the padding is applied in each layer to retain the important information [83]. Additionally, there is a stride which is nothing but the number of pixels by which kernel moves. We can calculate the output volume by using the formula given below [83]:

$$W_{\text{out}} = \frac{W - F + 2P}{S} + 1$$

where W is the size of an input image ( $W \times W \times D$ ). F is the number of kernels with a spatial dimension, S is the stride, and P is the padding.

The figure 3.2 [84] shows how a Convolutional Neural Network (CNN) uses a filter to detect patterns in an image. It starts with a 6x6 padded image and a 3x3 filter which slides across the image. At each step, the filter overlaps a region of the image and performs element-wise multiplication, and sums the result to form one value in the output matrix. For example, when applied to the region (in gray), filter produces a sum of 7. This process continues across image to generate a 6x6 result matrix which represents detected features.

**Pooling layers:** Pooling Layers are used to reduce the spatial dimensions of the feature maps (output of convolutional layers) while keeping the most important information [83]. There are two types of pooling, i.e., max pooling and average pooling. In Max pooling, layers select the maximum value from a region of the feature map where the kernel overlaps and in average pooling, layer takes the average value from a region of the feature map where the kernel overlaps [83]. This process reduces the computational load. Convolutional layers apply filters to the input images which outputs feature maps, which are then processed by pooling layers to reduce dimensions while retaining key information.

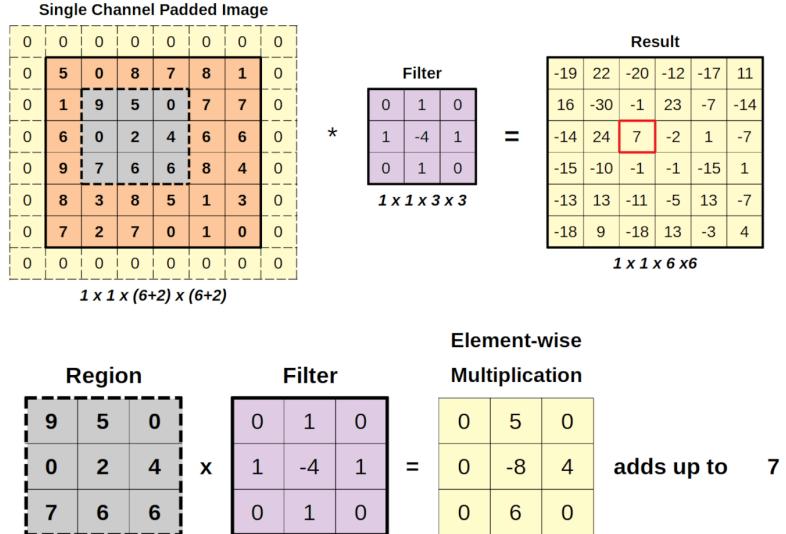


Figure 3.2: Working of CNN

**Activation functions:** Activation functions are very important for CNN layers [83]. Activation function is a mathematical formula that introduces non-linearity into the model filters. One most common activation function is ReLU (Rectified Linear Unit) that is applied after each convolutional layer to introduce non-linearity into the model [83]. ReLU sets all negative values to zero, allowing the network to learn more complex patterns.

**Fully connected layers:** Fully connected layers are simply a feed-forward neural network that takes the flattened output of the last pooling layer [83] and uses it to make predictions. These layers work similarly to traditional neural networks, where each neuron is connected to every neuron in the previous layer, combining all learned features to classify the input.

### 3.2.1 VGG-16

VGG-16 is Convolutional Neural Network (CNN) architectures, developed by the Visual Geometry Group (VGG) at the University of Oxford [85]. It was introduced in 2014 by Simonyan and Zisserman [85]. VGG-16 quickly became a benchmark in the field of computer vision. The model achieved an accuracy of 92.77% on test data using the ImageNet dataset, which contains 14 million images and 1,000 different classes [85]. Vgg-16 consists of 16 weight layers. Its structure consists of 13 convolutional layers

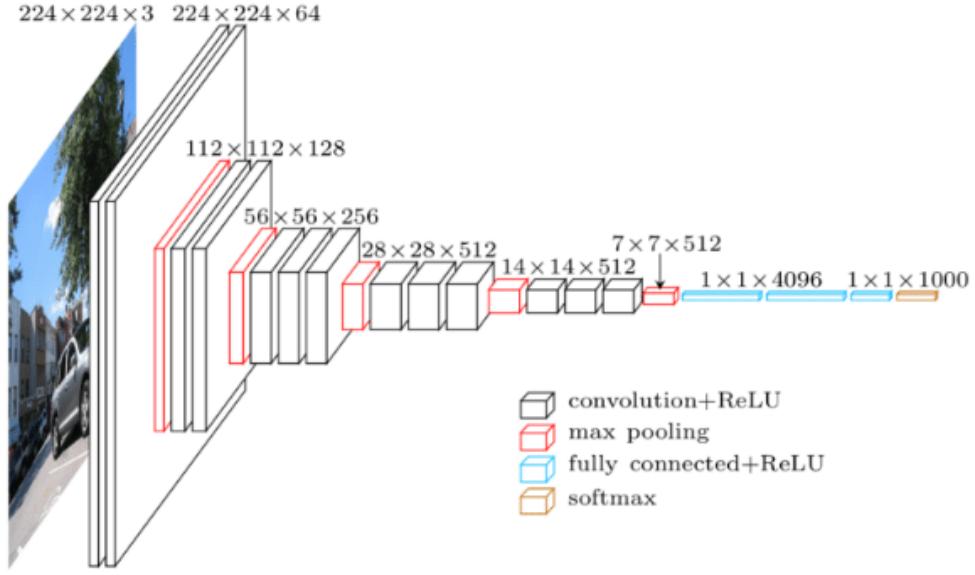


Figure 3.3: Architecture of VGG-16

organized into five blocks [86] (see Figure 3.3 [87] ). Each block has multiple convolutional layers and a max-pooling layer. Moreover, there are 4096 channels in the first two layers, and 1000 channels in the third layer and represents 1000 different labels categories. The last layer is SoftMax activation function that outputs the probability distribution over the classes typically used with 1000 output units for ImageNet classification and all hidden layers are followed by relu nonlinear activation function [88].

The input image that VGG-16 receives is typically  $224 \times 224$  pixels with 3 color channels, RGB. When this input image goes through model’s first set of convolutional layers, then the number of channels increases from 3 to 64. Then, after it passes through a max-pooling layer, the input image’s width and height are reduced by half making it  $112 \times 112$  pixels from  $224 \times 224$  pixels, it will keep 64 channels though. This pattern continues as the input image moves through vgg-16’s network. The final set of convolutional layers produces an output of  $7 \times 7$  pixels with 512 channels. After this, the output is passed through three fully connected layers which results in  $1 \times 1 \times 1000$ , which means 1000 values. These 1000 values are then fed into the SoftMax activation function that normalizes them into a range between 0 and 1 (with all the values adding up to 1). This process helps determine the probability that the image belongs to each of the 1000 categories that VGG-16 can classify [88] .

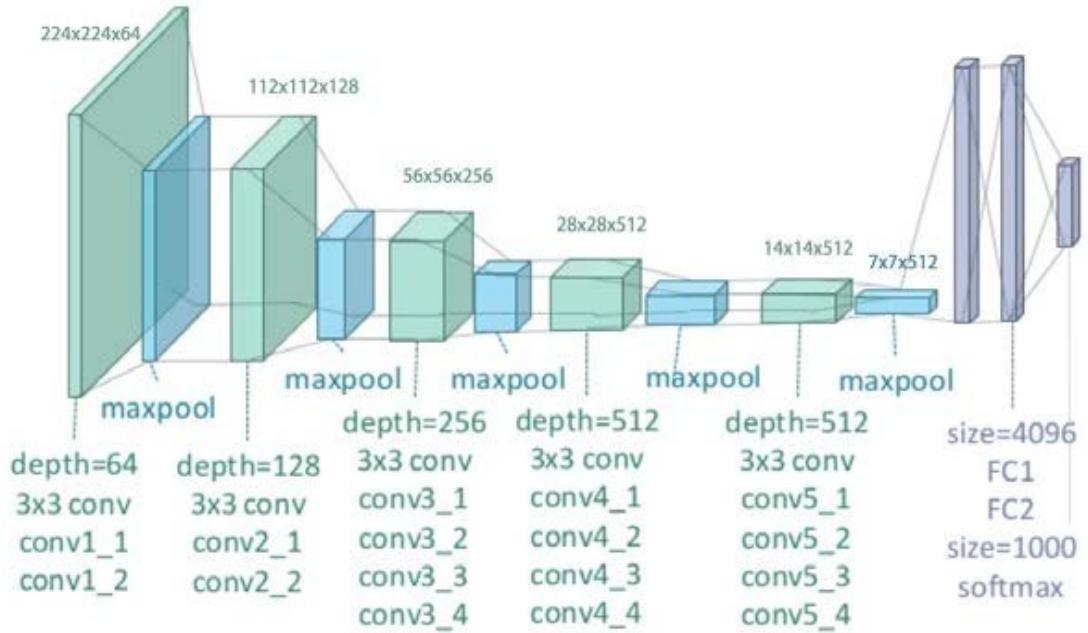


Figure 3.4: Architecture of VGG-19

### 3.2.2 VGG-19

VGG-19 is another convolutional neural network (CNN) architecture, developed by the Visual Geometry Group (VGG) at the University of Oxford [85]. VGG 19 architecture has 3 Fully connected layers, 16 convolution layers, 1 SoftMax layer, and 5 MaxPool layers [85]. Just like Vgg 16, Vgg-19 was also introduced in 2014 by Simonyan and Zisserman. VGG-19 features 19 weight layers, comprising 16 convolutional layers and 3 fully connected layers. The architecture is similar to VGG-16 but with increased depth, achieved by adding more convolutional layers to the last three blocks (see figure 3.4 [89]). Each convolutional block in VGG-19 is followed by a max-pooling layer similar to vgg-16, which helps reduce the spatial dimensions, and the network ends with the same fully connected layers as VGG-16.

The additional layers in VGG-19 make it a bit more powerful than VGG-16 as it enables the model to capture even finer details in images which improves its performance on more complex visual tasks, such as object detection and image segmentation [90]. This increased depth contributes to better feature extraction, making VGG-19 a powerful tool for tasks that require detailed visual analysis. However, VGG-19 has an even higher computational cost than VGG-16, with approximately 143 million parameters [91].

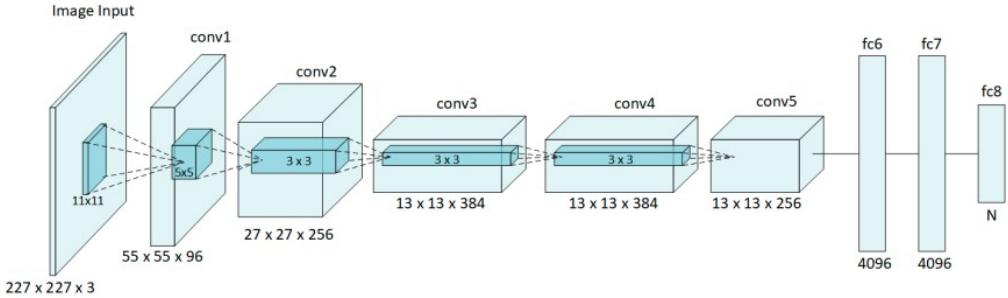


Figure 3.5: Architecture of AlexNet

### 3.2.3 AlexNet

AlexNet is developed by Krizhevsky et al in 2012. This model won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. Before AlexNet, CNN was limited to hand digit recognition tasks and Alexnet is recognized as the first model that revolutionized image classification and recognition [78]. Moreover, AlexNet introduced several key innovations that contributed to its success. One of them was the use of the Rectified Linear Unit (ReLU) activation function that significantly sped up the training process and to prevent overfitting, Alexnet also employed dropout in the fully connected layers which randomly "drops" neurons during training to encourage the network to learn more robust features. Another innovation was the use of overlapping max-pooling which enhanced the richness of features and AlexNet also makes use of GPUs for computing acceleration [92].

The architecture of AlexNet consists of eight layers with five convolutional layers followed by three fully connected layers [93] (see figure 3.5 [94]). The first convolutional layer uses 96 filters of size  $11 \times 11$  with a stride of 4, applied to the input image, and is followed by a max-pooling layer. The second convolutional layer applies to 256 filters of size  $5 \times 5$  and also includes max-pooling. The third, fourth, and fifth convolutional layers use 384, 384, and 256 filters, respectively, with only the fifth layer followed by max-pooling. These convolutional layers are designed to extract increasingly complex features from the input image, while the max-pooling layers reduce the spatial dimensions, helping to control the number of parameters and computational load. Additionally, there are 4096 neurons in each fully connected layer [93]. Following the convolutional layers, the output is flattened and passed through three fully connected layers. The final fully connected layer has 1000 neurons, corresponding to the 1000

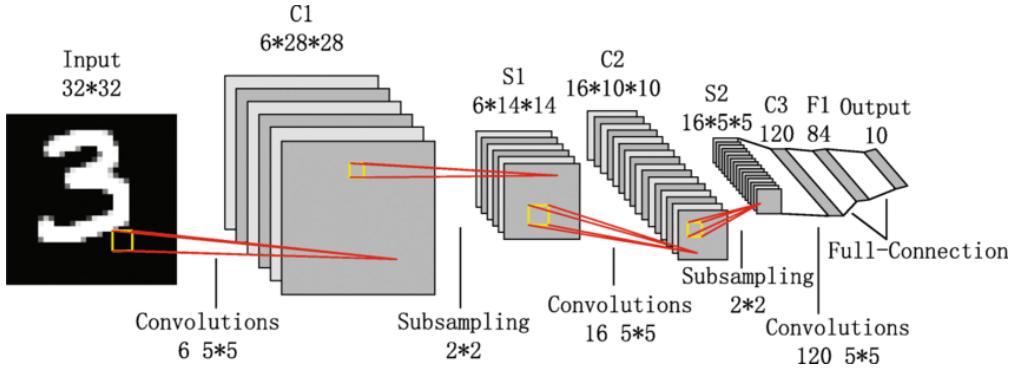


Figure 3.6: Architecture of LeNet-5

classes in the ImageNet dataset and finally, the output of this last layer is processed through a softmax function to produce the class probabilities [95].

### 3.2.4 LeNet-5

LeNet-5 is one of the earliest Convolutional Neural Networks (CNNs) [96], developed by Yann LeCun [97] in 1998. It was specifically designed for handwritten digit recognition, such as the digits used in the MNIST dataset [98]. The architecture of LeNet-5 is relatively simple by modern standards and is very famous as it was the first CNN [78]. LeNet-5 is a feed forward neural network [78] and consists of seven layers with two convolutional layers, two pooling layers, and three fully connected layers [92] as shown in figure 3.6 [99]. The input to the network is a 32x32 grayscale image. The first layer is a convolutional layer (C1) with six 5x5 filters which produce six feature maps of size 28x28. This layer captures low-level features such as edges and simple shapes. The next layer is a subsampling (S2) or average pooling layer that reduces the dimensionality of the feature maps to 14x14 by applying a 2x2 filter with a stride of 2, along with an averaging operation. This step helps in reducing the computational complexity and ensuring the network is more invariant to small translations of the input. Following this, another convolutional layer (C3) with sixteen 5x5 filters is applied, generating sixteen 10x10 feature maps. Unlike the first convolutional layer, this one does not connect every input map to every output map; instead, it uses a specific pattern of connections, which reduces the number of parameters and introduces some degree of specialization among the feature maps. The next layer is another subsampling (S4) layer, which further reduces the feature maps to 5x5. After this, a third convolutional layer (C5) with 120 5x5 filters is applied, but since the input size to this layer matches the filter size, the output

is a set of 120  $1 \times 1$  feature maps, effectively functioning as fully connected layers. The final layers include a fully connected layer (F6) with 84 neurons, which is connected to the output layer that has ten neurons—one for each digit class in the MNIST dataset. The activation function used throughout LeNet-5 is typically the sigmoid or hyperbolic tangent (tanh), which was standard at the time [97].

### 3.2.5 GoogleNet

Googlenet [100] also known as Inception v1 [101] is a Convolutional Neural Network (CNN) architecture introduced by Google in 2014. The main goal was to achieve the highest accuracy with less computational cost [100]. Additionally, it won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014. The architecture of GoogLeNet is 22 layers deep (or 27 layers if pooling is included) [100] and its innovative use of the Inception module, which allows the network to capture complex features at multiple scales with reduced computational cost [78] (see figure 3.7 [102]). The Inception module is the core building block of GoogLeNet [103], designed to address the challenge of deciding the optimal filter size at each layer. Instead of choosing a single filter size, the Inception module applies multiple filters ( $1 \times 1$ ,  $3 \times 3$ , and  $5 \times 5$ ) as well as a  $3 \times 3$  max-pooling operation in parallel to the same input. The outputs from these operations are then concatenated along the depth dimension, creating a rich feature representation that captures both local and global features simultaneously. The  $1 \times 1$  convolutions within the module serve a dual purpose: they help reduce the dimensionality of the data, thereby lowering the computational complexity, and they enable the network to learn more intricate patterns by combining multiple feature maps [100]. The overall architecture of GoogleNet begins with traditional convolutional and max-pooling layers to process the input image, which is typically  $224 \times 224$  pixels. The initial layers apply a  $7 \times 7$  convolution followed by max-pooling, then a series of  $1 \times 1$  and  $3 \times 3$  convolutions to further refine the feature maps. After these preliminary layers, the network transitions into the main body, which consists of nine stacked Inception modules. These modules are grouped into three sections, each progressively increasing in complexity. This modular design allows the network to capture increasingly abstract representations of the input data as it moves deeper into the network.

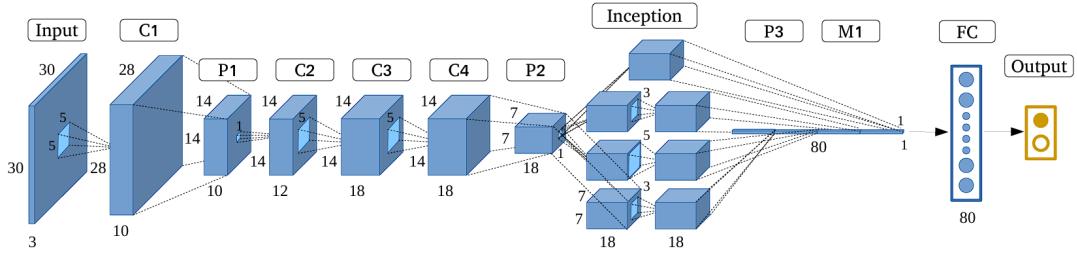


Figure 3.7: Architecture of GoogleNet

To address the vanishing gradient problem, which can hinder the training of deep networks, GoogLeNet includes two auxiliary classifiers connected to intermediate layers [104]. These auxiliary classifiers act as additional sources of gradient flow during training, helping the network converge more effectively.

Rather than using fully connected layers as we have seen in earlier CNN architectures, GoogLeNet employs a global average pooling layer before the final output [105]. This layer averages the spatial dimensions of the feature maps, producing a compact 1x1 feature vector for each class, which is then fed into a softmax layer to generate the final classification probabilities. This approach significantly reduces the number of parameters, making GoogLeNet more efficient in terms of both memory and computation. With approximately 5 million parameters [106], much fewer than the AlexNet [107], GoogLeNet demonstrates that deep networks can achieve high performance without being overly resource intensive.

### 3.2.6 ResNet

ResNet stands for Residual Network, was introduced by He et al [108] in 2015. ResNet employed residual learning that can handle the vanishing gradient problem [109]. This problem occurs when gradients become too small during back-propagation which makes it difficult to update the weights effectively in very deep networks. In a traditional neural network, each layer learns a function that directly maps input to output. However, as networks grow deeper, it becomes difficult to optimize this mapping. ResNet addresses this by introducing residual blocks that allow each layer to learn the residual or difference between the input and the desired output, rather than directly learning the output [110]. This skip connection helps preserve the flow of gradients during backpropagation, making it feasible to train very deep networks without encountering the vanishing

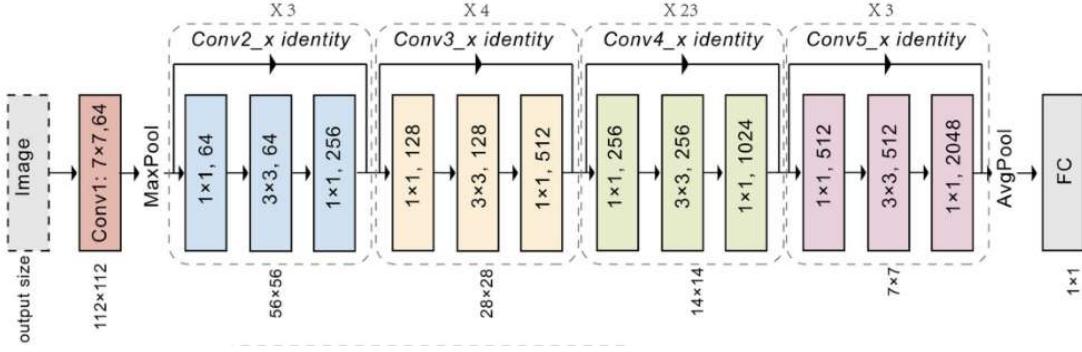


Figure 3.8: Architecture of ResNet

gradient problem. The architecture typically starts with a conventional convolutional layer, followed by a series of residual blocks grouped into stage where each stage is responsible for learning features at different levels of abstraction. In a residual block the input  $x$  is passed through two weight layers, each followed by a ReLU activation function, producing an output  $\mathcal{F}(x)$ . This output  $\mathcal{F}(x)$  is then added to the original input  $x$ . The combined result  $\mathcal{F}(x) + x$  is then passed through another ReLU activation as shown in figure 3.9 [111]

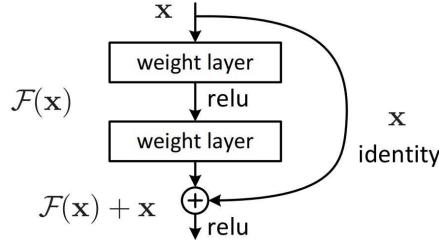


Figure 3.9: Residual blocks

A key component of the residual block is the use of skip connection [112]. The identity shortcut simply passes the input directly to the output, while the convolutional shortcut involves a 1x1 convolution operation, often used when the dimensions of the input and output need to be matched. This approach allows ResNet to handle cases where the input and output have different dimensions, ensuring the shortcut connections are valid. Additionally, ResNet blocks often include batch normalization and ReLU (Rectified Linear Unit) activation functions, which further stabilize training and improve the network's ability to learn complex patterns(see figure 3.8 [111] ).

Another significant aspect of ResNet is its bottleneck design in deeper variants like ResNet-50, ResNet-101, and ResNet-152 [113]. In these architectures, each residual block contains three layers instead of the standard two, i-e a 1x1 convolution layer that

reduces the number of channels (dimensionality reduction), followed by a 3x3 convolution layer that processes the reduced representation, and finally another 1x1 convolution layer that restores the original number of channels. This bottleneck structure reduces the computational load and memory usage, making it more feasible to train very deep networks.

At the end of the network, ResNet typically uses a global average pooling layer, similar to GoogLeNet, to reduce the spatial dimensions of the final feature maps before feeding them into a fully connected layer that produces the output classification. The global average pooling layer helps to minimize the number of parameters and avoid overfitting, while also ensuring that the network remains computationally efficient.

ResNet architectures come in various depths such as Resnet-50,101 and 152. However, Resnet-152, which has 152 layers, won the 2015-ILSVRC competition. The performance of ResNet in image recognition tasks highlights the crucial role of representational depth in a wide range of visual recognition tasks [78].

### 3.2.7 MobileNet

MobileNet is Convolutional Neural Networks (CNNs)’s architecture designed by Google [114] with the primary goal of providing efficient, lightweight models suitable for mobile and embedded devices, where computational resources are limited [115]. The key innovation behind MobileNet is the introduction of depth wise separable convolutions [115], a technique that significantly reduces the number of parameters and computational complexity compared to traditional convolutional layers, making it particularly well-suited for environments with restricted computational power.

In a standard convolutional layer, each filter is applied to all input channels, and then the results are combined to produce the output feature map. This process is computationally expensive, especially as the number of filters and input channels increases. MobileNet addresses this inefficiency by breaking the standard convolution operation into two simpler and more efficient operations, i-e depthwise convolution and pointwise convolution [116]. In depthwise convolution, a single convolutional filter is applied independently to each input channel, rather than across all channels [115]. This step reduces the number of computations significantly because each filter only needs to process one channel at a time. Architecture of Mobilenet is shown in figure 3.10 [117].

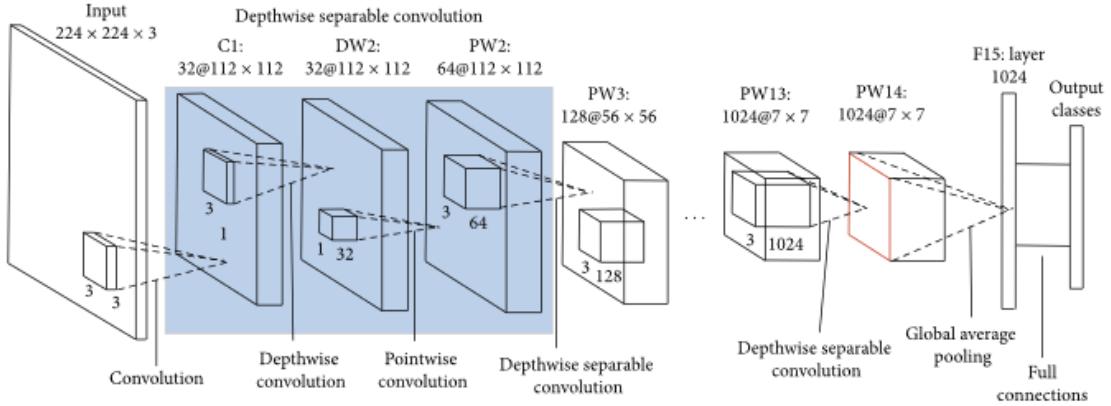


Figure 3.10: Architecture of MobileNet

Following the depthwise convolution, MobileNet uses a pointwise convolution, which is essentially a 1x1 convolution applied across all channels [115]. The pointwise convolution combines the outputs from the depthwise convolution and allows the network to mix the information across channels. By separating the spatial and channel-wise operations, MobileNet dramatically cuts down the number of parameters and floating-point operations (FLOPs) required, without a significant loss in accuracy.

The original MobileNet architecture, often referred to as MobileNetV1, is composed of a series of these depthwise separable convolutions, organized in blocks that allow the network to learn increasingly complex features at each layer. MobileNetV1 also introduces a couple of hyperparameters, width multiplier and resolution multiplier [118], that allow users to tradeoff between model size, speed, and accuracy. The width multiplier reduces the number of channels in each layer, effectively thinning the network, while the resolution multiplier reduces the input image size, making the model faster and lighter at the expense of some accuracy. Building on the success of MobileNetV1, MobileNetV2 introduced several enhancements, including the inverted residual block with linear bottlenecks [92]. The inverted residual block begins with a pointwise convolution to expand the number of channels, then applies a depthwise convolution, and finally uses another pointwise convolution to reduce the number of channels back to the original count [92]. The use of linear bottlenecks at the end of these blocks prevents the loss of information during dimensionality reduction, which can occur with non-linear activations like ReLU [92]. This structure maintains efficiency and improves the performance of the model.

MobileNetV3 further refines the architecture by incorporating advances like the swish

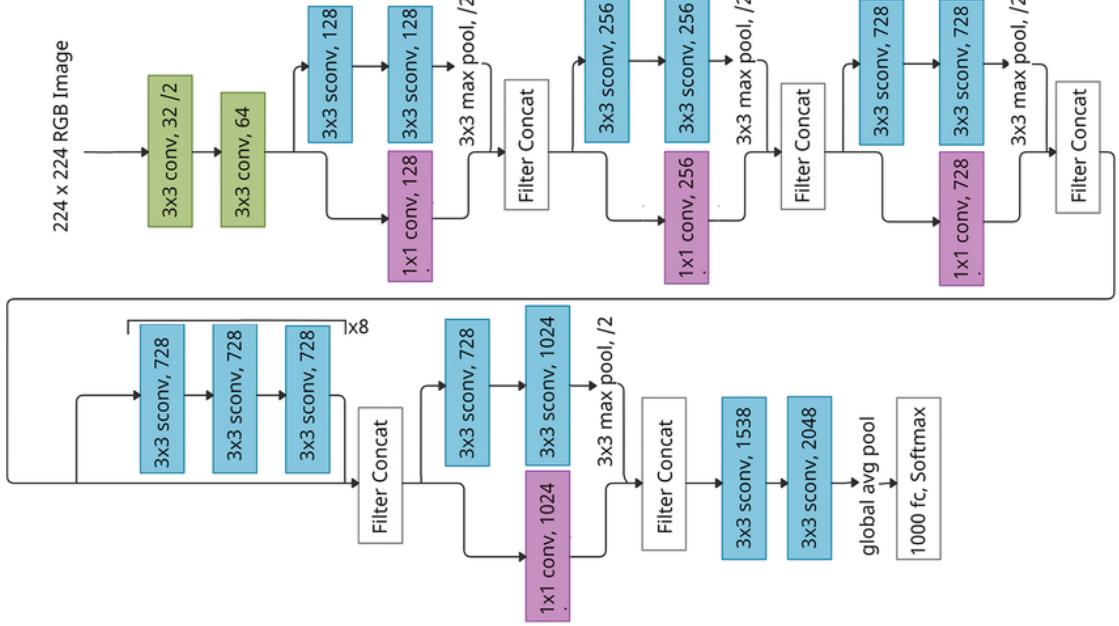


Figure 3.11: Architecture of Xception

activation function and squeeze-and-excitation (SE) modules [92]. These SE modules adaptively recalibrate the channel-wise feature responses, enhancing the representational power of the network. MobileNetV3 also leverages NAS (Neural Architecture Search) [92] to automatically discover and optimize the model architecture, striking a better balance between latency and accuracy across a range of mobile devices.

### 3.2.8 Xception

Xception [119] stands for “Extreme Inception” is a deep Convolutional Neural Network (CNN) architecture proposed by François Chollet in 2017 as an extension and improvement of the Inception model family. The core idea behind Xception is to replace the Inception modules with depthwise separable convolutions [78], a form of convolution that significantly enhances both the computational efficiency and the representational power of the network. The architecture of Xception is founded on the principle of depthwise separable convolutions, which decompose the standard convolution operation into two separate steps [120] i-e depthwise convolution and pointwise convolution . In a depthwise convolution, a single filter is applied to each input channel independently, rather than applying multiple filters across all channels simultaneously. This step captures spatial relationships within each channel without mixing the channels. Following

this, a pointwise convolution (a  $1 \times 1$  convolution) is applied across all the channels to combine the features learned in the depthwise step [78]. This separation of spatial and cross-channel convolutions reduces the computational complexity significantly while allowing the network to learn more nuanced and fine-grained features.

The Xception architecture consists of 36 convolutional layers organized into 14 modules, with each module containing one or more depthwise separable convolution layers [121] (see figure 3.11 [122]).

Xception uses residual connections [105], inspired by the ResNet architecture. In Xception, residual connections are employed across most modules to allow for more efficient gradient flow and to mitigate the vanishing gradient problem. These connections help the network maintain high accuracy even as the number of layers increases, enabling the training of deeper networks without significant degradation in performance.

In the final stages of the architecture, Xception employs global average pooling instead of fully connected layers [123]. This layer averages each feature map into a single value which drastically reduces the number of parameters and prevents overfitting. The output from this layer is then fed into a softmax classifier to produce the final predictions.

### 3.2.9 DenseNet

DenseNet [124] stands for Dense Convolutional Network and was introduced by Gao, et al. in 2017. DenseNet is the improvement of Resnet [125]. The core idea behind DenseNet is its unique connectivity pattern, where each layer is directly connected to every other layer in a feed-forward manner to improve the information flow between layers [124]. In a DenseNet, each layer receives inputs from all preceding layers and passes on its output to all subsequent layers [124] as shown in figure 3.12 [126]. This is achieved by concatenating the feature maps from previous layers, rather than summing them up as done in traditional residual networks like ResNet. As a result, the input to any given layer includes not only the raw input data but also the feature maps from all preceding layers, providing the network with a rich set of features at each stage [124]. This dense connectivity allows for a more efficient flow of information and gradients throughout the network.

The structure of DenseNet is organized into blocks known as Dense Blocks [127]. Within each dense block, every layer is connected to every other layer, forming a highly

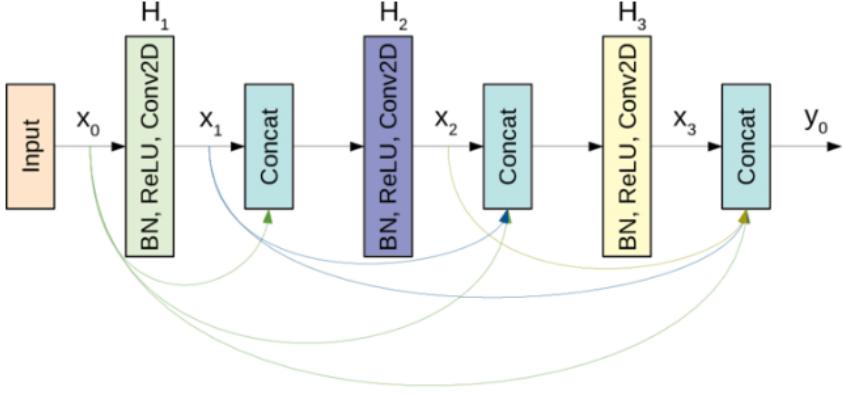


Figure 3.12: Architecture of DenseNet

interconnected network. Between these dense blocks, transition layers are used to control the complexity of the model by reducing the size of the feature maps through a combination of convolution and pooling operations [124]. DenseNet has better parameter efficiency as compared to Resnet [124]. Despite the dense connections, DenseNet requires fewer parameters. For instance, there are 15.3 M parameters in 250-layer Densenet model. DenseNet also benefits from improved feature reuse because of input concatenation, the feature-maps learned by any of the DenseNet layers is accessible to all subsequent layers [124].

DenseNet is available in several variants, such as DenseNet-121, DenseNet-169, DenseNet-201, and DenseNet-264, which differ primarily in the number of layers and the depth of the dense blocks.

### 3.2.10 EfficientNet

EfficientNet is introduced by Google in 2019 [128], designed to achieve state-of-the-art performance on image classification tasks [129] while being more computationally efficient than previous architectures. The key innovation behind EfficientNet is the development of a compound scaling method to balance the depth, width, and resolution of model [130] in a systematic and efficient manner. This approach allows EfficientNet to scale up the model to achieve higher accuracy while maintaining a lower computational cost compared to traditional methods that scale only one dimension at a time, such as depth or width alone.

EfficientNet begins with EfficientNet-B0 that is a relatively small and simple model trained using techniques like neural architecture search (NAS) which is a technique for



Figure 3.13: Architecture of EfficientNet

automating the design of neural network [128]. The real power of EfficientNet comes from its scaling strategy. The authors of EfficientNet introduced a compound scaling method that uniformly scales all dimensions of the network [131], depth (the number of layers), width (the number of channels in each layer), and resolution (the input image size). Compound scaling applies a carefully balanced scaling factor to all three dimensions simultaneously, ensuring that the network grows in a balanced and optimized way. This method allows EfficientNet to maintain high accuracy while using significantly fewer parameters and FLOPs (floating point operations) than previous state-of-the-art models [128]. Furthermore, it utilizes Squeeze-and-Excitation Networks that improve channel interdependencies [128]. Architecture of EfficientNet is shown in 3.13 [132]. The architecture of EfficientNet is built upon MobileNetV2’s inverted residual blocks [128]. These blocks consist of an expansion phase, a depthwise convolution, and a projection phase, which helps the network learn more complex features without a significant increase in computational cost.

EfficientNet-B0, the smallest model in the EfficientNet family, starts with a 224x224 input image size and uses these inverted residual blocks throughout its architecture. As you move up the EfficientNet family from B0 to B7, the models become progressively larger, and use higher input resolutions, more channels, and deeper networks, all according to the compound scaling formula. One of the most impressive features of EfficientNet is its scalability across a wide range of model sizes. The smallest model, EfficientNet-B0, has only 5.3 million parameters [133]. At the other end of the spectrum, EfficientNet-B7, the largest model in the family, achieves top-tier performance on benchmarks such as ImageNet [128].

### 3.3 Evaluation Metrics

Evaluation metrics are important tools in machine learning [134], especially when we are dealing with classification tasks such as benign and malignant classification. These metrics give us a quantitative means to see how well a model performs [135] and also helps us to compare different models to determine the most effective one for a certain task. Additionally, evaluation metrics are derived from the confusion matrix [136], which summarizes the performance of a model by comparing predicted labels to actual labels. The key components of the confusion matrix are True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) [137]. We have discussed all of these in details below.

**True Positive (TP):** True positive is when both prediction and actual values are correct or we can say when the model's prediction matches the actual values [137]. For instance, correctly identifying a malignant case as malignant.

**True Negative (TN):** When both prediction and actual values are same such as correctly identifying a benign case as benign, it is considered a true negative [137].

**False Positive (FP):** False positive is when a model prediction is yes, however, actual value is no [137] such as model predicts a benign case as malignant (Type I error).

**False Negative (FN):** False positive is when a model prediction is no, however, actual value is yes [137] such as model predicts a malignant case as benign (Type II error).

In this section, we will discuss several key performance metrics used to evaluate classification models, including Accuracy, Precision, Recall, Specificity, F1-Score, False Positive Rate (FPR), False Negative Rate (FNR), False Discovery Rate (FDR), Area Under the Curve (AUC), and Time Complexity, each of which is explained in detail below.

### 3.3.1 Accuracy

Accuracy measures the overall correctness of the model and it is calculated as the number of all correct predictions (both benign and malignant) divided by the total number of the dataset [137]. And is given by the equation [137]:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (3.9)$$

where  $TP$  is True Positive,  $FP$  is False Positive,  $FN$  is False Negative, and  $TN$  is True Negative.

### 3.3.2 Precision

Precision focuses on the quality of positive predictions [137] (malignant cases). It answers the question, "Out of all predicted malignant cases, how many were actually malignant?". It is calculated as amount of True positive divided by predicted [137]. It is given by the equation [137]:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3.10)$$

where  $TP$  is True Positive and  $FP$  is False Positive.

### 3.3.3 Recall (Sensitivity or True Positive Rate)

Recall measures the model's ability to identify the all actual malignant cases. It answers "Out of all actual malignant cases, how many did the model correctly predict?". It is calculated as an amount of True positive divided by actual yes [137]. It is given by the equation [137]:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3.11)$$

where  $TP$  is True Positive and  $FN$  is False Negative.

### 3.3.4 Specificity

Specificity is how correctly model predicts the benign cases (true negative rate) [138]. It answers the question, "Out of all actual benign cases, how many were correctly identified?". It is given by the equation [138]:

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (3.12)$$

where  $FP$  is False Positive and  $TN$  is True Negative.

### 3.3.5 F1-score

The F1-Score is the harmonic mean of precision and recall, a balance between the two metrics [139]. It is useful when you want to find a balance between precision and recall, especially in cases of uneven class distribution. It is given by the equation [139]:

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.13)$$

or

$$\text{F1 score} = 2 \times \frac{\left(\frac{TP}{TP+FP}\right) \times \left(\frac{TP}{TP+FN}\right)}{\left(\frac{TP}{TP+FP}\right) + \left(\frac{TP}{TP+FN}\right)}$$

where  $TP$  is True Positive,  $FP$  is False Positive,  $FN$  is False Negative, and  $TN$  is True Negative.

### 3.3.6 False Positive Rate (FPR)

FPR measures the proportion of benign cases that were incorrectly classified as malignant. It is also known as the probability of a Type I error and is important in understanding the model's performance in distinguishing between benign and malignant cases. FPR can be calculated by the equation [140]

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN} \quad (3.14)$$

where  $FP$  is False Positive and  $TN$  is True Negative.

### 3.3.7 False Negative Rate (FNR)

FNR represents the proportion of malignant cases that were incorrectly classified as benign, also called probability of Type II error [141]. FNR is given by the equation [140]:

$$\text{False Negative Rate (FNR)} = \frac{FN}{FN + TP} \quad (3.15)$$

where  $TP$  is True Positive and  $FN$  is False Negative.

### 3.3.8 False discovery Rate (FDR)

FDR measures the proportion of incorrect predictions [142] such as malignant cases that were actually benign. A high FDR indicates that the model has a high number of false positives. FDR is given as [143]:

$$\text{False Negative Rate (FNR)} = \frac{FP}{FP + TP} \quad (3.16)$$

where  $TP$  is True Positive and  $FP$  is False Positive.

### 3.3.9 Area under the curve (AUC)

The AUC is derived from the Receiver Operating Characteristic (ROC) curve [138], which plots the True Positive Rate (Sensitivity) against the False Positive Rate at various threshold levels [144]. AUC is a graph of true positive rate vs false positive rate [138] and AUC value of 1 indicates perfect classification, while a value of 0.5 indicates random guessing [145].

### 3.3.10 Time Complexity

Time complexity specifically refers to the amount of computational time an algorithm takes to complete as the size of the input data grows, often expressed using Big O notation (e.g.,  $O(n)$ ,  $O(\log n)$ ), and is a crucial factor in determining the scalability and efficiency of machine learning models.

### 3.4 Time Complexity of CNN

$$TC : O \left( \sum_{i=1}^D M_i^2 \cdot K_i^2 \cdot C_{i-1} \cdot C \right) \quad (3.17)$$

where D is the number of convolution layers of the neural network; i is the ith convolution layer of the neural network; M denotes the side length of the output characteristic graph of each convolution kernel; K is the side length of each convolution kernel;  $C_i$  represents the output channel of the ith convolution layer of the neural network[146].

### 3.5 Time complexity of VGG-16

$$TC : O \left( \sum_{l=1}^D M_l^2 \cdot k_l^2 \cdot C_{l-1} \cdot C_l + \sum_{j=1}^{D_{FC}} C_{j-1} \cdot M_j^2 \cdot m_j \right) \quad (3.18)$$

where  $D$  represents the number of all convolutional layers of the neural network, that is, the network depth;  $l$  represents the first convolutional layer of the neural network;  $C_l$  represents the number of output channels  $C_{out}$  of the  $l$ th convolutional layer of the neural network, that is, the number of convolutional kernels in that layer.  $D_{FC}$  represents the number of fully connected layers, and  $m_j$  represents the number of features to be output in that layer [O].

### 3.6 Time complexity of VGG-19

$$TC : O \left( \sum_{l=1}^d n_{l-1} \cdot s_l^2 \cdot n_l \cdot m_l^2 \right) \quad (3.19)$$

where l is the subscript of a convolutional layer. d is the number of the convolutional layers.  $n_l$  is the number of convolution kernels in the lth network.  $n_{l1}$  is the number of input channels in the lth network.  $s_l$  is the size of a convolution kernel.  $m_l$  is the size of the output feature map [147]

### 3.7 Time complexity of AlexNet

$$TC : O \left( \sum_{j=1}^n y_{j-1} x_j^2 y_j z_j^2 \right) \quad (3.20)$$

Here,  $n$  represents the number of convolutional layers,  $y_{j-1}$  is the number of input channels of the  $j$ th layer,  $y_j$  is the number of filters of the  $j$ th layer,  $x_j$  is the spatial size of the filters, and  $z_j$  denotes the size of the output feature map[148].

### 3.8 Time complexity of Mobilenet-v2

$$N_{op} = \sum_{i=1}^L \sum_{j=1}^M \sum_{k=1}^N F(i, j, k) \quad (3.21)$$

where L is the number of layers, M is the number of output feature maps in each layer, N is the number of input feature maps in each layer, and  $F(i,j,k)$  is the number of operations performed[149].

The total number of operations is given by the following equation:

$$N_{tot} = N_{op} \times D \quad (3.22)$$

where D is the number of images in the dataset. The computational complexity is given by the following equation:

$$TC : \frac{N_{tot}}{F} \quad (3.23)$$

where F is the number of floating-point operations per second [149].

### 3.9 Time complexity of Xception

$$TC : O \left( \sum_{i=1}^D M_i^2 \cdot K_i^2 \cdot C_{i-1} + M_i^2 \cdot C_{i-1} \cdot C_i \right) \quad (3.24)$$

where D is the number of convolution layers of the neural network; i is the ith convolution layer of the neural network; M denotes the side length of the output characteristic graph of each convolution kernel; K is the side length of each convolution kernel;  $C_i$  represents the output channel of the ith convolution layer of the neural network[146].

### 3.10 Time complexity of ResNet

$$TC : O \left( (N_{PAM} - 1) \sum_{l=1}^L \left( (1 + \ln P)^2 \cdot H_k^{l^2} \cdot N_{in}^l \cdot N_{out}^l \right) + \sum_{l=1}^L \left( H_{out}^{l^2} \cdot H_k^{l^2} \cdot N_{in}^l \cdot N_{out}^l \right) \right) \quad (3.25)$$

where  $N_{PAM}$  is the number of target symbols.

$L$  represents the number of convolution layers in the  $i$ -th residual block (res-block).

$H_k^l$  is a characteristic of the convolution layer (e.g., kernel size).

$N_{in}^l$  and  $N_{out}^l$  denote the input and output sizes of the  $l$ -th layer in the residual block.

$H_{out}^l$  refers to the output characteristic at the  $l$ -th layer (e.g., height or dimensionality of the output) [150].

### 3.11 Time complexity of GoogleNet

$$1.5 \times 10^{10} \quad (3.26)$$

Computational complexity can be given by Floating Point Operations Per Second (FLOPs) [151]. FLOPs indicates how many floating-point calculations a system can perform per second.

### 3.12 Time complexity of EfficientNet

FLOPs(Floating Point Operations Per Second) of EfficientNet are only 22.34 Million [152].

### 3.13 Time complexity of DenseNet

$$TC : O(D3.H.W.F1.F2.T) \quad (3.27)$$

where, F1 is input feature map channel, the output feature map dimensions be height H, width W, and channel F2, and the feature map temporal dimension stacking be T, and the convolutional kernel size be D [153].

### 3.14 Time complexity of Lenet-5

$$TC : \frac{1}{3} \cdot O \left[ \left( 1 \cdot 3^2 \cdot 32 \cdot \frac{N_R}{2} \cdot \frac{N_T}{2} \right) + \left( 32 \cdot 3^2 \cdot 64 \cdot \frac{N_R}{4} \cdot \frac{N_T}{4} \right) \right] \propto O(N_R \cdot N_T) \quad (3.28)$$

where  $N_T$  is the number of transmit antennas at the transmitter and  $N_R$  is the number of receive antennas at the receiver[154].

# **Chapter 4**

## **Methodology**

In this chapter, we outline the systematic approach we used to conduct our research on AI-based skin cancer detection. This process involves data collection, preprocessing, algorithm selection and implementation, performance evaluation, and providing recommendations based on the findings. Our study begins with a comprehensive literature review to identify relevant research papers, articles, and sources related to AI-based skin cancer detection. We focused on gathering information about state-of-the-art algorithms, dataset sources, and best practices currently employed in the field. This review helped us understand the existing landscape, identify gaps in the research, and determine the most promising techniques for further investigation.

## 4.1 Dataset Collection

Firstly, we conducted in depth literature review to identify common datasets used in skin cancer detection and classification studies and the most commonly used datasets are HAM10000, dataset from ISIC archive, and PH2 in the state-of-the-art [19]. These datasets have been sourced from reputable archives such as the ISIC Archive and the Hospital Pedro Hispano. We've summarized these datasets in Table 4.1.

### 4.1.1 HAM10000

The HAM10000 (Human Against Machine) dataset is one of the largest and most widely used datasets for the detection of melanoma and classification of skin lesions [155]. It contains 11,720 images [156] It is sourced from International Skin Imaging Collaboration (ISIC) archive and it's created to facilitate the training of machine learning models in diagnosing skin cancer and other skin conditions. The dataset includes seven different classes of skin lesions including Actinic Keratosis, Basal Cell Carcinoma , Benign Keratosis , Dermatofibroma , Melanoma , Melanocytic nevus and Vascular lesion [157]. The dataset is publicly available through the ISIC (International Skin Imaging Collaboration) Archive.

### 4.1.2 ISIC-2017

The ISIC 2017 dataset was developed as part of the 2017 ISIC Challenge and its main goal was to develop image analysis tools that can automatically diagnose melanoma

from dermoscopic images [158]. ISIC 2017 dataset consists of 2750 skin cancer images with 2000 images for training datasets, 150 images for test datasets, and 600 images for validation datasets [159]. Additionally, the size range for ISIC-2017 is  $540 \times 722 \times 3$  to  $4499 \times 6748 \times 3$  pixels [159].

#### 4.1.3 ISIC-2018

The ISIC 2018 dataset was developed as a part of 2018 ISIC Challenge. This dataset has more images as compared to ISIC 2017 dataset. It consists of 11527 images with 10,015 images for training and 1512 images for testing dataset [160]. Moreover, ISIC 2018 consists of 7 different classes of skin lesions including Melanoma (MEL), Nevi (NV), Basal cell carcinoma (BCC), Actinic keratosis / Bowens disease (intraepithelial carcinoma) (AKIEC), Benign keratosis (BKL), Der- matofibroma (DF) and Vascular (VASC) [160].

#### 4.1.4 PH2

The PH2 dataset is significantly smaller compared to the other datasets listed above. It is collected from the Hospital Pedro Hispano in Portugal, and it consists of 200 skin lesions [161]. Each image has a size of  $768 \times 560$  pixels [161]. It has 3 different types of skin lesions including Atypical Nevus, Common Nevus and Melanoma [162]. Additionally, it includes 80 images of atypical nevi, 80 images of common nevi, and 40 images of melanoma cases [161].

Ref	Dataset	Source	No of Images
[19]	HAM10000	ISIC ARCHIVE	11,720
[14]	ISIC-2017	ISIC ARCHIVE	2750
[9]	ISIC-2018	ISIC ARCHIVE	11527
[19]	PH2	Hospital Pedro Hispano, Portugal	200

Table 4.1: Datasets commonly used in literature for AI-based skin cancer detection

For this study, we used a dataset from Kaggle, which is from the ISIC archive, consisting of 3297 skin lesion images with a resolution of  $224 \times 224$  pixels. Additionally, the dataset consists of two classes: benign and malignant. The benign class has 1440 training

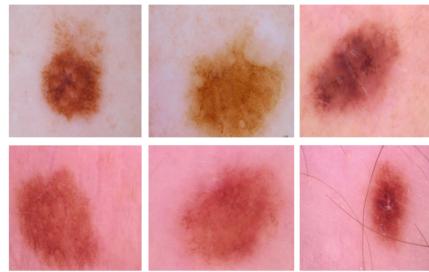


Figure 4.1: Benign and Malignant images from ISIC archive

images and 360 testing images while the malignant class has 1197 training images and 300 testing images.

	<b>Benign</b>	<b>Malignant</b>
<b>Training Dataset</b>	1440	1197
<b>Testing Dataset</b>	360	300
<b>TOTAL</b>	1800	1497

Table 4.2: Dataset

## 4.2 Preprocessing

The data preprocessing stage involves several steps to ensure the images are suitable for algorithmic analysis:

**Resizing:** Using OpenCV, images will be resized to a uniform dimension to ensure consistency across the dataset and to meet the input requirements of various CNN architectures.

**Normalization:** Pixel values will be normalized to a standard range (e.g., 0-1) to improve the convergence of the neural networks during training.

**Augmentation:** Data augmentation techniques such as rotation, flipping, and zooming will be applied to increase the diversity of the training set and help prevent overfitting. This step enhances the model's ability to generalize by exposing it to various transformations of the images.

### **4.3 Algorithm Selection**

The selection of algorithms is based on their demonstrated effectiveness and widespread usage in the field of skin cancer detection. We reviewed literature to identify the most promising deep learning architectures, focusing on those that have achieved high accuracy and robustness in previous studies. The selected algorithms for our study include: DenseNet, AlexNet, VGG-16, VGG-19, MobileNet, Xception, EfficientNet, LeNet-5, ResNet and GoogLeNet. These models are chosen for their proven track record in image classification tasks and their architectural diversity, allowing for a comprehensive evaluation of different approaches.

### **4.4 Algorithm Implementation**

The selected algorithms were implemented in Python with TensorFlow and Keras libraries. We used Goggle Collab for implementation of the selected algorithms which is a cloudbased platform that provides free access to GPUs, which significantly accelerates the training process and allows for efficient experimentation with different models. The results of implemented algorithms are given in Table 5.1 and 5.2.

### **4.5 Identifying Performance Metrics**

To assess the performance of each algorithm, we conducted an in-depth literature review to identify the key performance metrics used in the field for evaluating the performance of machine learning models in medical image analysis. These metrics include accuracy, precision, recall, F1-score and specificity [8] [19]. We used these key performance metrics to assess the performance of popular algorithms used for skin cancer detection. Additionally, we also used computational cost to evaluate the algorithms.

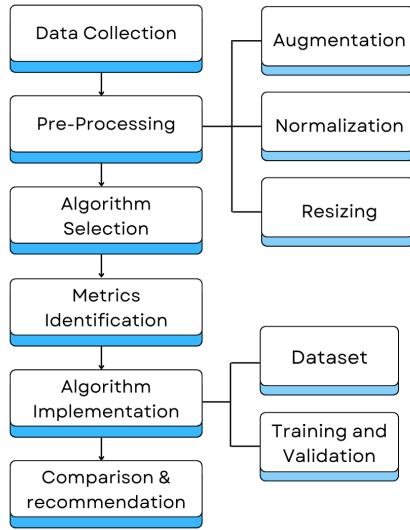


Figure 4.2: Proposed Approach Schematic

#### 4.6 Algorithm Evaluation

We applied the selected algorithms to the preprocessed datasets and evaluated their performance based on the defined metrics. Furthermore, we conducted a comparative analysis of selected algorithms on the basis of identified key metrics.

#### 4.7 Recommendation

Based on the evaluation and analysis, we will provide recommendations for specific algorithms based on defined metrics for skin cancer detection. Our recommendations will take into account not only the accuracy and F1 score but also the computational cost and practicality of deploying these models in real-world clinical settings. By highlighting the strengths and limitations of each approach, we aim to guide future research and development efforts towards the most effective and efficient solutions for skin cancer detection.

# **Chapter 5**

## **Results and Discussion**

In this chapter, we will provide a detailed comparison of the various Convolutional Neural Network (CNN) architectures mentioned in the literature review, which we have implemented in our study. The results are compiled in Tables 5.1 and 5.2, which shows the performance of these architectures across several key metrics: accuracy, precision, recall, F1-score, specificity, False Positive Rate (FPR), False Negative Rate (FNR), False Discovery Rate (FDR), and the Area Under the Curve (AUC).

Table 5.1: Performance Comparison of CNN Architectures

<b>Architecture</b>	<b>Accuracy</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>
VGG-16	97%	87%	87%	87%
VGG-19	88%	87%	88%	87%
ALEXNET	87%	87%	87%	87%
MOBILENET-V2	88%	88%	88%	88%
XCEPTION	82%	82%	81%	81%
LENET-5	51%	50%	50%	49%
RESNET	77%	78%	77%	76%
DENSENET	81%	82%	80%	81%
EFFICIENTNET	55%	27%	50%	35%
GOOGLENET	82%	82%	82%	82%

Table 5.2: Performance Comparison of CNN Architectures

<b>Architecture</b>	<b>Specificity</b>	<b>FPR</b>	<b>FNR</b>	<b>FDR</b>	<b>AUC</b>
VGG-16	87%	13%	13%	15%	87%
VGG-19	88%	12%	13%	14%	88%
ALEXNET	84%	16%	10%	18%	87%
MOBILENET-V2	87%	13%	10%	10%	88%
XCEPTION	88%	12%	26%	16%	81%
LENET-5	63%	37%	64%	55%	50%
RESNET	68%	33%	13%	31%	77%
DENSENET	90%	10%	29%	15%	80%
EFFICIENTNET	100%	0%	100%	NAN	50%
GOOGLENET	86%	14%	23%	17%	82%

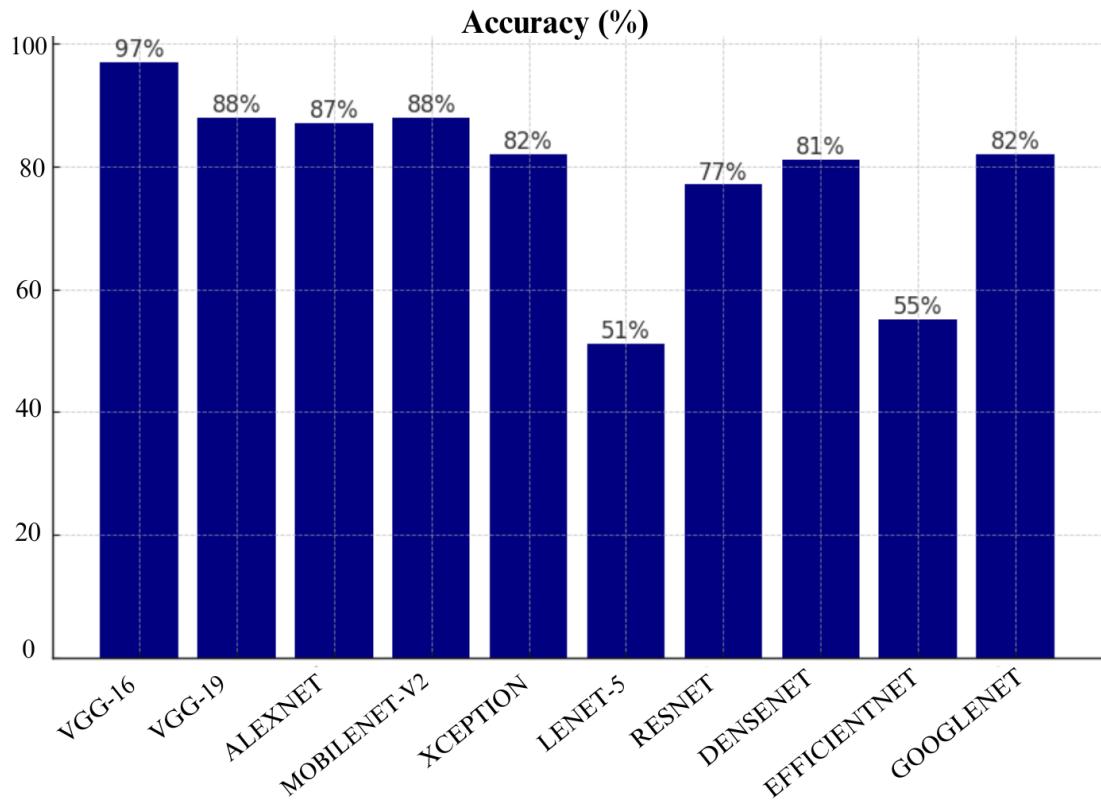


Figure 5.1: Accuracy of CNN models

## 5.1 Accuracy

Accuracy is an important metric as it indicates the overall performance of the algorithm by calculating the proportion of correct predictions (both true positives and true negatives) out of the total predictions. As shown in Table 5.1, the VGG-16 model achieved the highest accuracy at 97%, followed by VGG-19 with 88%, and MobileNet-V2 with 88%. These results indicate that VGG-16 is the most reliable in terms of making correct predictions between malignant and benign cases. Other CNN architectures such as GoogLeNet and Xception performed moderately with 82% accuracy each. However, models like Lenet-5 and EfficientNet showed poor performance, with Lenet-5 achieving only 51% accuracy and EfficientNet only reaching 55%.

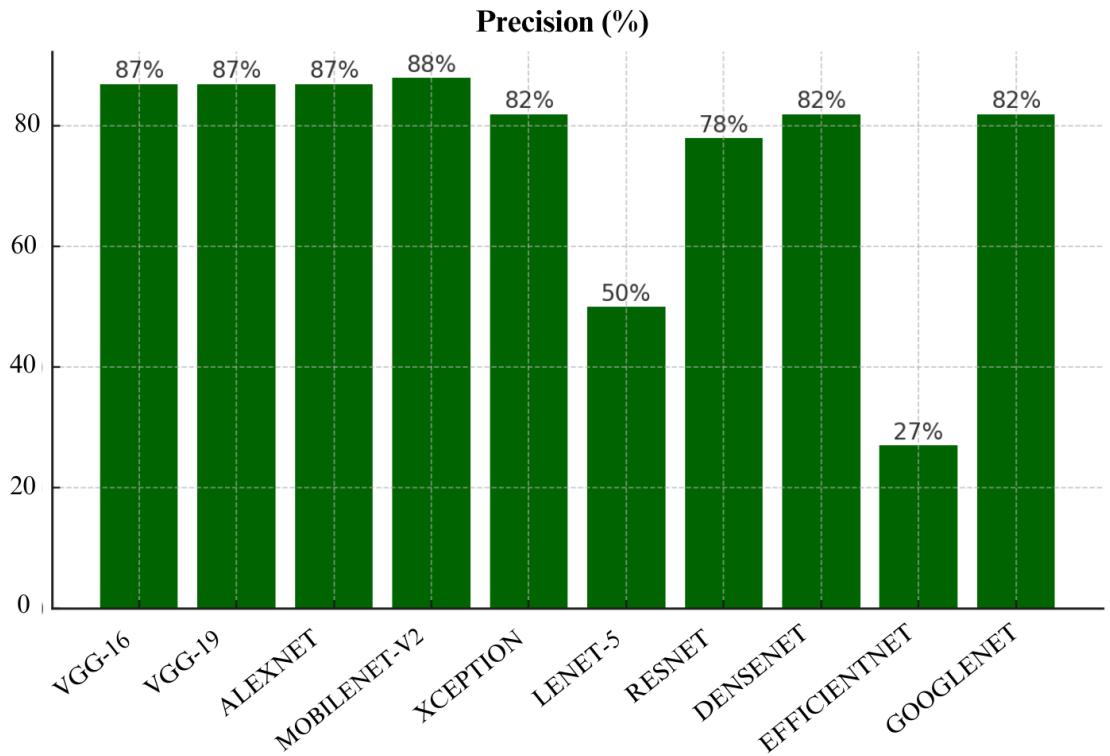


Figure 5.2: Precision of CNN models

## 5.2 Precision

Precision measures the proportion of true positive predictions over all positive predictions. This is crucial in medical applications like skin cancer detection because we aim to reduce false positives that could lead to unnecessary treatments. From Table 5.1, both VGG-16 and VGG-19 performed equally well, achieving 87% precision. Moreover, MobileNet-V2 and Xception closely followed with 88% and 82%, respectively. These models demonstrated consistent precision which indicates that they can be trusted to avoid over-predicting malignant cases. EfficientNet, on the other hand, had a precision score of only 27% which makes it highly unreliable in this aspect. Lenet-5, with a precision of 50% performed similarly poorly which shows that it needs significant improvement in this metric.

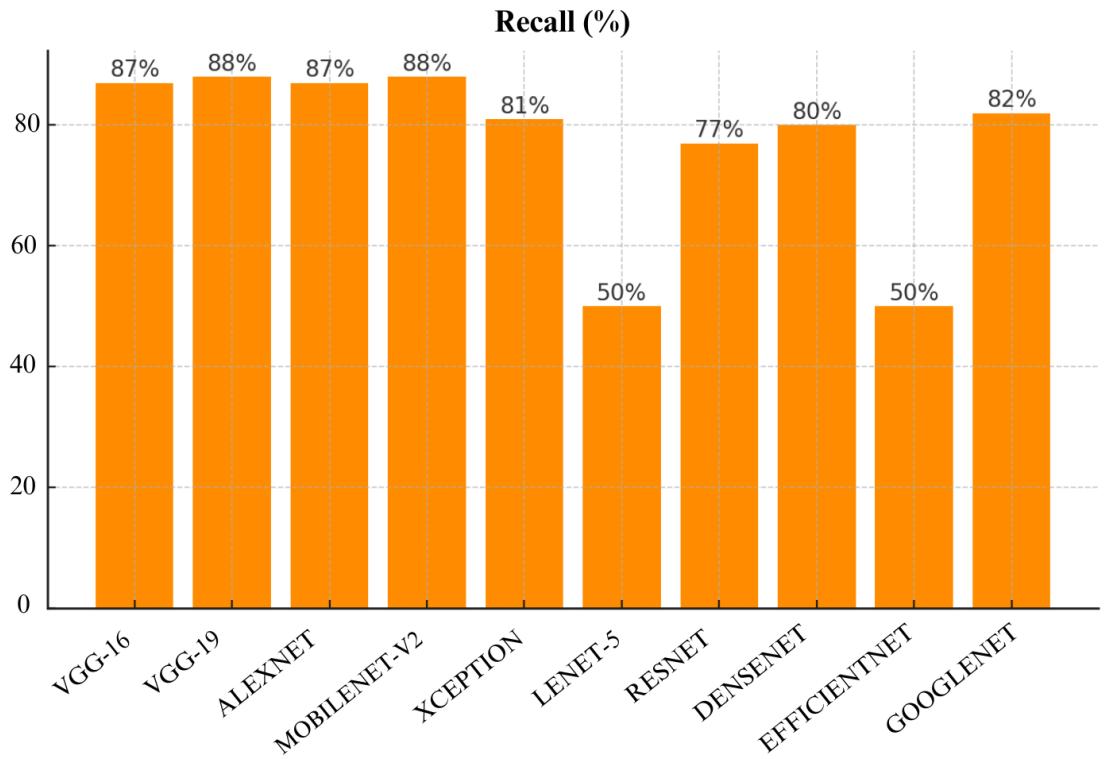


Figure 5.3: Recall of CNN models

### 5.3 Recall

Recall is a metric that calculates the proportion of true positive cases detected out of all actual positives. It is highly significant in skin cancer detection as we prioritize detecting malignant cases (true positives). Based on Table 5.1, VGG-16 had the best recall score, again reaching 87%, followed closely by MobileNet-V2 and VGG-19, both scoring 88% and 87% respectively. GoogLeNet and Xception also performed well with 82% recall each. However, models like Lenet-5 and EfficientNet performed poorly, achieving only 50% recall which indicates that they have failed to detect a huge portion of malignant cases which may result in missed diagnoses in real-world applications.

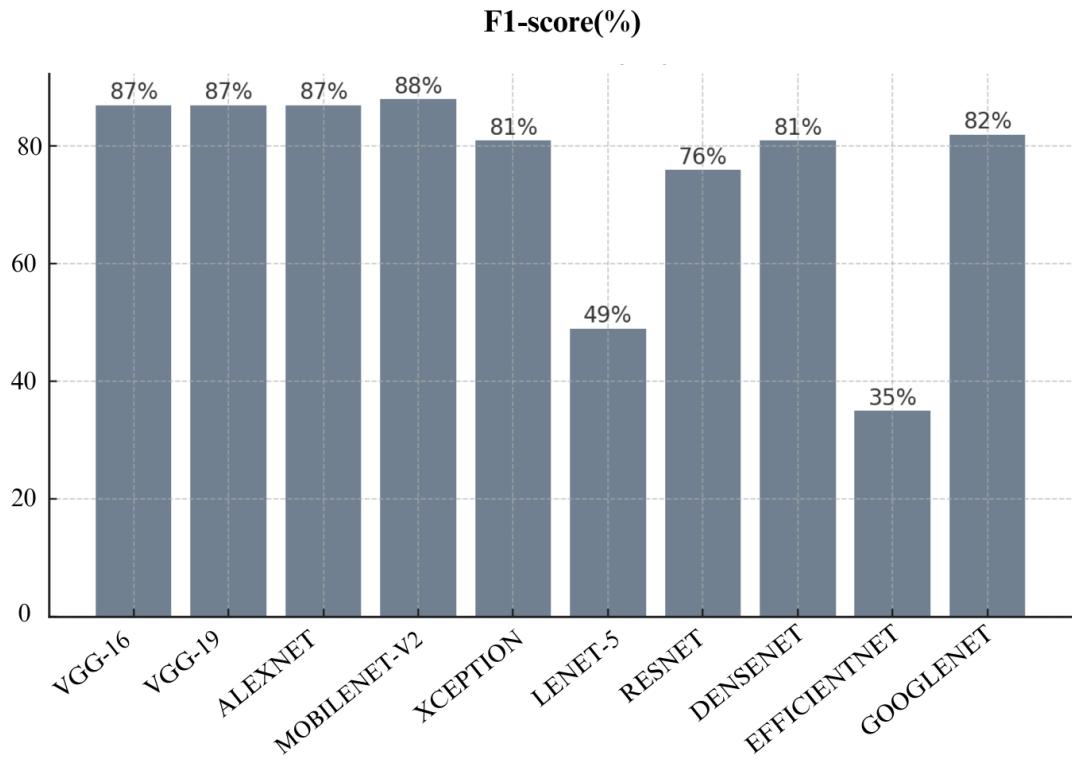


Figure 5.4: F1-Score of CNN models

#### 5.4 F1-Score

The F1-score is a balance between precision and recall, and it is very important when there is a tradeoff between these two metrics. A model with a high F1-score is generally considered more reliable in making predictions as compared to that which has lower F1-score. In Table 5.1, VGG-16 stands out with the highest F1-score of 87% which indicates its balanced and superior performance in both precision and recall. VGG-19 and MobileNet-V2 follow closely, both have achieved an F1-score of 87% and 88%, respectively. EfficientNet, with its F1-score of just 35%, demonstrates poor performance in terms of this metric as compared to other architectures, this also shows that it does not perform well in both precision and recall. Lenet-5 also performed poorly here with only 50% which makes it less reliable overall.

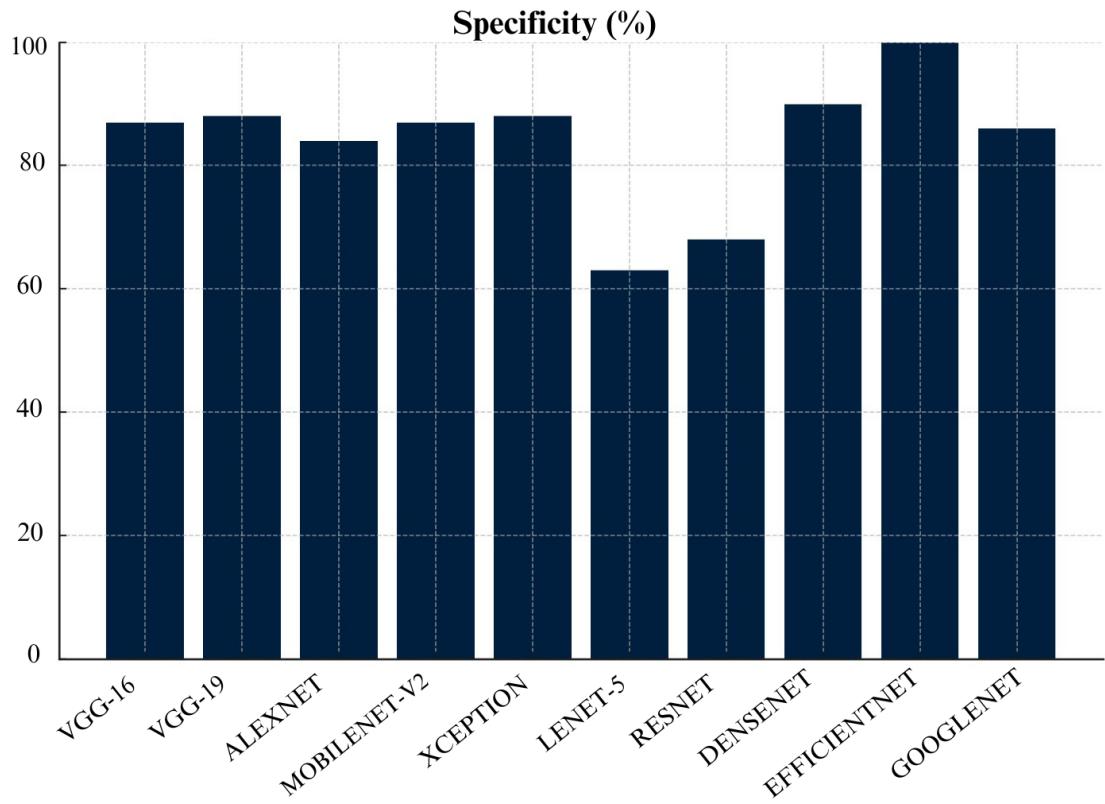


Figure 5.5: Specificity of CNN models

## 5.5 Specificity

Specificity measures the proportion of true negatives out of all actual negatives. In the context of skin cancer detection, high specificity reduces the false positives which means it avoid any unnecessary anxiety or treatment for patients. As you can see in Table 5.2, Densenet has achieved highest specificity of 90%, followed by VGG-19 at 88%, and VGG-16 at 87%. These architectures excel in correctly identifying benign cases. Lenet-5 performed the worst, with a specificity score of 63%, which means that it frequently misclassified benign cases as malignant cases. EfficientNet also scored poorly, with specificity of only 70%.

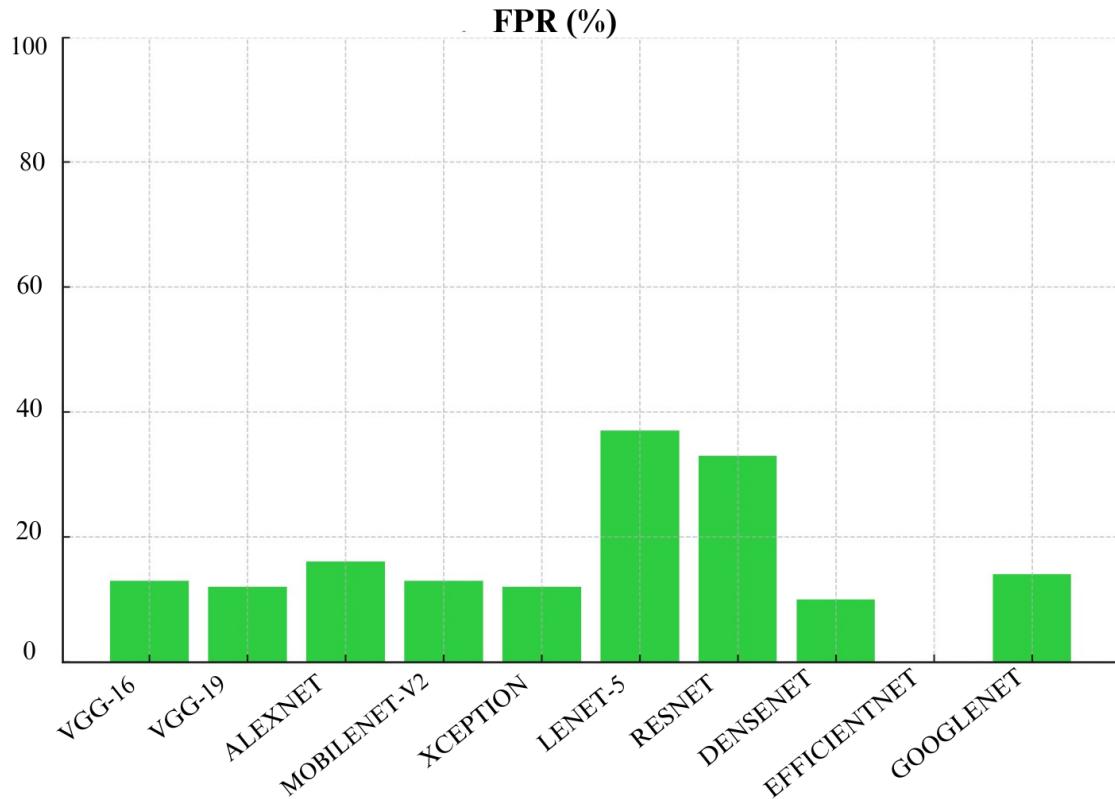


Figure 5.6: FPR of CNN models

## 5.6 False Positive Rate (FPR)

The False Positive Rate (FPR) is the proportion of benign cases incorrectly classified as malignant. It is the inverse of specificity. As shown in Table 5.2, Densenet had the lowest FPR which is only 10%, followed by VGG-19 at 12% and VGG-16 at 13%. These models performed well in minimizing false positives which is very important when making sure that patients are not given any unnecessary treatments. Lenet-5 had the highest FPR at 37% which means significant tendency to misclassify benign cases and this could lead to overdiagnosis. EfficientNet also had a relatively high FPR which is 30%, which shows its underperformance in comparison to the other top models.

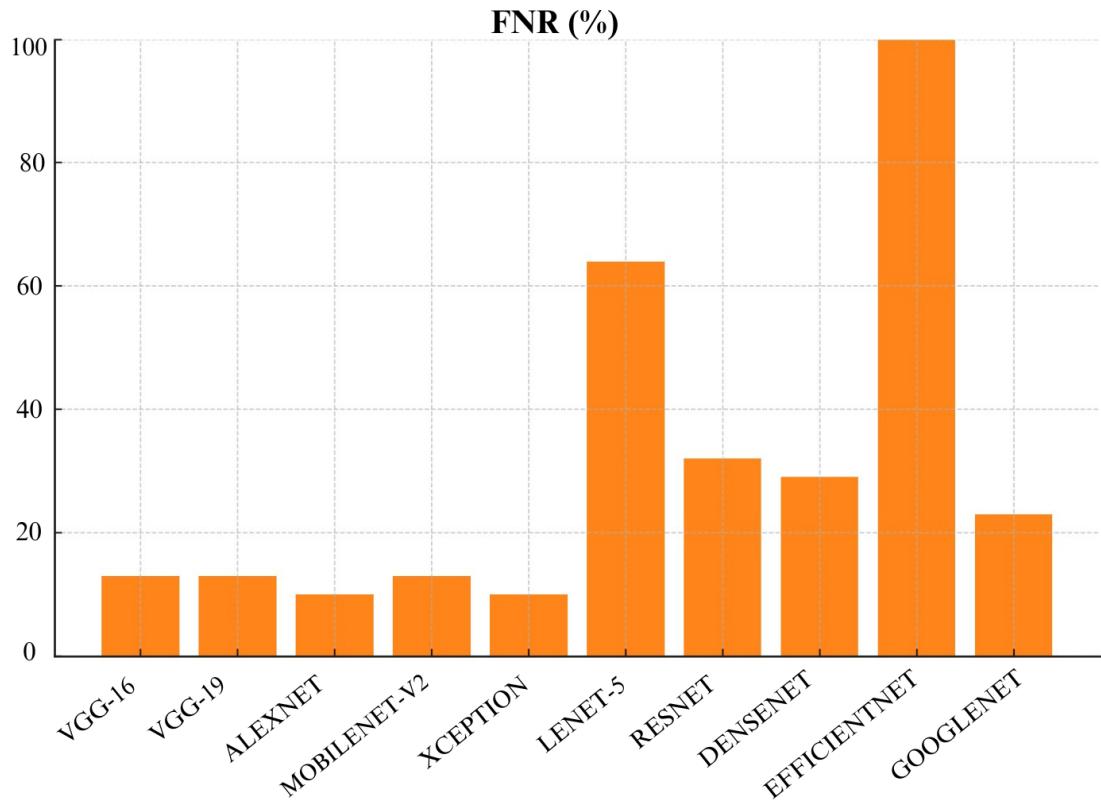


Figure 5.7: FNR of CNN models

### 5.7 False Negative Rate (FNR)

The False Negative Rate (FNR) refers to the proportion of malignant cases incorrectly classified as benign, which can be particularly dangerous in medical diagnostics. In Table 5.2, EfficientNet had the lowest FNR of 0%, but we have to be careful when considering this result since the model's overall performance was not good in other metrics. Densenet also performed well and has achieved a low FNR of 10% which shows that it can effectively identify malignant cases. On the other hand, Lenet-5 had the highest FNR at 50% which indicate that the half of the malignant cases were missed. This makes Lenet-5 a poor choice for skin cancer detection. Xception and GoogLeNet both had moderate FNRs of 18% which makes them reasonably effective, though they are not the best performers.

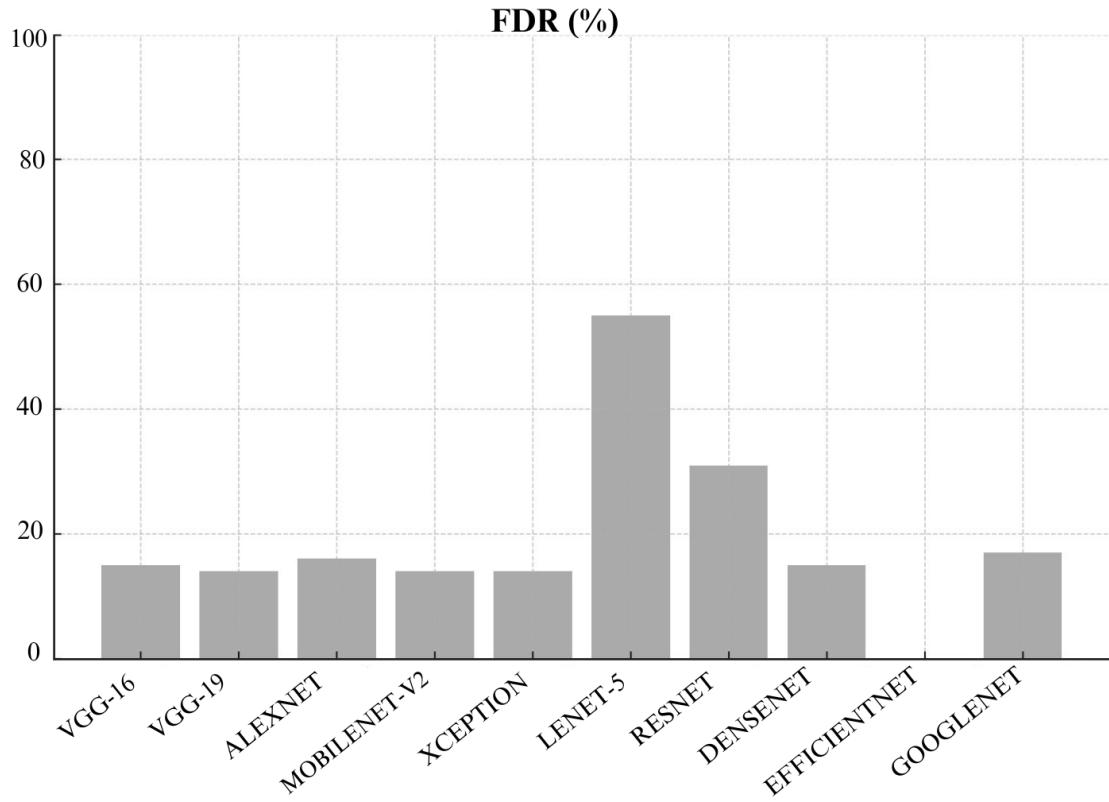


Figure 5.8: FDR of CNN models

## 5.8 False Discovery Rate (FDR)

The False Discovery Rate (FDR) is the proportion of false positives among all positive predictions. In Table 5.2, Densenet again performed best with FDR of only 9%, closely followed by VGG-19 and VGG-16, which both achieved 13% and 15% FDRs, respectively. These models show robustness in minimizing false discoveries. EfficientNet, with an FDR of 70% performed extremely poorly which highlights its tendency to misclassify benign cases as malignant. Similarly, Lenet-5 had a high FDR of 50% which further shows its unreliability in this study.

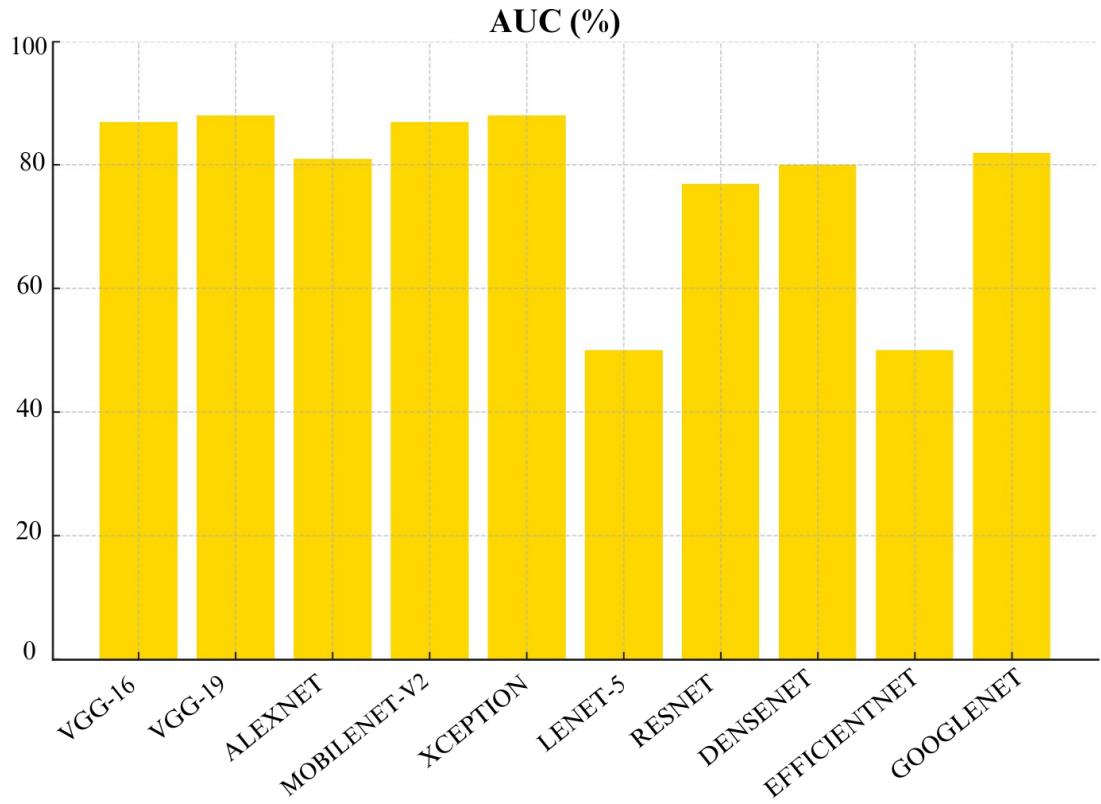


Figure 5.9: AUC of CNN models

### 5.9 Area Under the Curve (AUC)

The AUC is a critical metric that summarizes the overall performance of the model, taking into account both the true positive rate and false positive rate across different thresholds. As shown in Table 5.2, VGG-19 achieved the highest AUC at 88%, followed by VGG-16 at 87%. These models demonstrate a very strong ability to distinguish between malignant and benign cases across various thresholds which makes them highly reliable. MobileNet-V2 and AlexNet also performed well and have achieved AUC scores of 88% and 87%, respectively. EfficientNet and Lenet-5 had the lowest AUC of 51% and 50%, respectively, which further shows their inadequacy in this application.

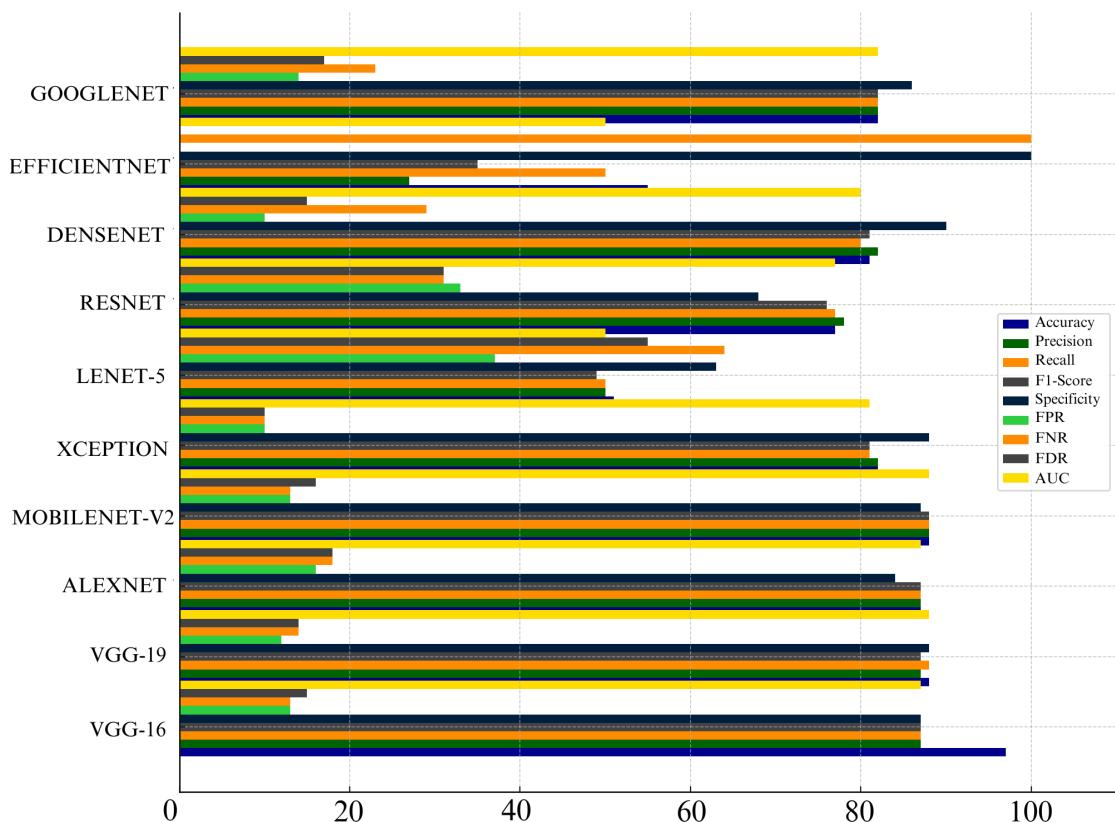


Figure 5.10: Comparison of CNN models based on performance metrics

When we look at the overall performance of the algorithms across all metrics, we can see there's a clear trend that certain CNN architectures excel in specific areas. VGG-16 stands out as the best performer in terms of accuracy, recall, and F1-score, it shows a very strong balance between detecting the true positives and minimizing the false negatives. VGG-19 isn't far behind with its performance specifically in high specificity and AUC, which means it's particularly good at distinguishing between the benign and malignant cases without too many false positives. MobileNet-V2 also performed admirably across several metrics which prove it to be a solid, well-rounded model. DenseNet while not as accurate as VGG-16 or VGG-19, had the lowest False Positive Rate (FPR) and False Discovery Rate (FDR) which makes it highly reliable for avoiding unnecessary treatments. Meanwhile, EfficientNet and Lenet-5 have consistently struggled across almost all metrics which shows that simpler architectures or those that are not optimized

for this particular task might miss a significant number of cases or generate many false positives. Overall, the choice of algorithm really depends on the specific context and priorities in a medical setting, it depends whether you prioritize avoiding missed diagnoses (recall) or reducing false alarms (specificity).

# **Chapter 6**

## **Conclusion**

Based on our evaluation, we can conclude that VGG-16 and VGG-19 were the top-performing CNN architectures in this study. VGG-16 achieved the highest accuracy of 97%, recall and F1-score of 87%, which makes it the most reliable model for detecting skin cancer. It also achieved high precision of 87%, which ensures less false positives. VGG-19 also performed exceptionally well with 88% accuracy and the highest AUC at 88% which shows its strong ability to differentiate between malignant and benign cases. Its specificity was also impressive which is 88% which indicate it can effectively reduce chances of over-diagnosis. Furthermore, MobileNet-V2 and Densenet also demonstrated strong performance, particularly in minimizing false positives (FPR) and false negatives (FNR). However, models like Lenet-5 and EfficientNet proved to be unsuitable for this task, performing poorly across most metrics, particularly in recall, F1-score, and AUC. Therefore, they are not recommended for practical skin cancer detection. In a practical clinical setting, the choice of model would depend on the priorities of the healthcare provider. For instance, VGG-16 with 87% recall would be an excellent choice if detecting malignant cases (recall) is prioritized, However, if avoiding false positives (specificity) is more critical then Densenet with 90% specificity or VGG-19 with 88% specificity would be more appropriate choices.

Overall, the performance of these models show promising results for improving the detection and diagnosis of skin cancer using AI-based approaches, though careful selection of the architecture is essential for optimizing diagnostic accuracy and reliability. Furthermore, the choice of algorithm may depend on specific clinical requirements, computational resources, and the nature of the dataset. Lastly, the interpretability of AI models in medical contexts remains a critical concern as understanding how decisions are made by the algorithms is crucial for trust and acceptance among healthcare professionals. Additionally, Challenges such as data privacy, model interpretability, ensuring robust performance under varied conditions, and managing computational costs is crucial. To address these challenges, robust countermeasures such as data augmentation techniques, model regularization, and interpretability methods can be implemented. Overall, the application of deep learning in skin cancer detection shows significant potential to augment clinical decision-making and improve patient outcomes.

# Bibliography

- [1] M. M. Rahman, M. K. Nasir, A. Nur, S. I. Khan, S. Band, I. Dehzangi, A. Beheshti, H. A. Rokny *et al.*, “Hybrid feature fusion and machine learning approaches for melanoma skin cancer detection,” 2022.
- [2] M. R. Hasan, M. I. Fatemi, M. Moniruzzaman Khan, M. Kaur, and A. Zagaria, “Comparative analysis of skin cancer (benign vs. malignant) detection using convolutional neural networks,” *Journal of Healthcare Engineering*, vol. 2021, no. 1, p. 5895156, 2021.
- [3] R. R. Kumar, R. Varun, J. Sreeshwan, K. A. Kumar, U. Rana, and A. Rajyalakshmi, “Feasible skin lesion detection using cnn and rnn,” in *E3S Web of Conferences*, vol. 430. EDP Sciences, 2023, p. 01050.
- [4] A. Javaid, M. Sadiq, and F. Akram, “Skin cancer classification using image processing and machine learning,” in *2021 international Bhurban conference on applied sciences and technologies (IBCAST)*. IEEE, 2021, pp. 439–444.
- [5] K. Das, C. J. Cockerell, A. Patil, P. Pietkiewicz, M. Giulini, S. Grabbe, and M. Goldust, “Machine learning and its application in skin cancer,” *International Journal of Environmental Research and Public Health*, vol. 18, no. 24, p. 13409, 2021.
- [6] S. Kavitha, R. Shalini, N. H. Sree, and J. Akash, “Intelligent segmentation and classification for skin cancer prediction,” in *2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*. IEEE, 2023, pp. 1–6.

- [7] Y. Filali, S. Abdelouahed, and A. Aarab, “An improved segmentation approach for skin lesion classification,” *Statistics, Optimization & Information Computing*, vol. 7, no. 2, pp. 456–467, 2019.
- [8] B. C. Furriel, B. D. Oliveira, R. Prôa, J. Q. Paiva, R. M. Loureiro, W. P. Calixto, M. R. Reis, and M. Giavina-Bianchi, “Artificial intelligence for skin cancer detection and classification for clinical environment: a systematic review,” *Frontiers in Medicine*, vol. 10, p. 1305954, 2024.
- [9] S. K. Singh, V. Abolghasemi, and M. H. Anisi, “Fuzzy logic with deep learning for detection of skin cancer,” *Applied Sciences*, vol. 13, no. 15, p. 8927, 2023.
- [10] H. Orhan and E. Yavşan, “Artificial intelligence-assisted detection model for melanoma diagnosis using deep learning techniques,” *Mathematical Modelling and Numerical Simulation with Applications*, vol. 3, no. 2, pp. 159–169, 2023.
- [11] N. Rezaoana, M. S. Hossain, and K. Andersson, “Detection and classification of skin cancer by using a parallel cnn model,” in *2020 IEEE international women in engineering (WIE) conference on electrical and computer engineering (WIECON-ECE)*. IEEE, 2020, pp. 380–386.
- [12] B. Mazoure, A. Mazoure, J. Bédard, and V. Makarenkov, “Dunescan: a web server for uncertainty estimation in skin cancer detection with deep neural networks,” *Scientific Reports*, vol. 12, no. 1, p. 179, 2022.
- [13] G. Alwakid, W. Gouda, M. Humayun, and N. Jhanjhi, “Diagnosing melanomas in dermoscopy images using deep learning,” *Diagnostics*, vol. 13, no. 10, p. 1815, 2023.
- [14] M. Nawaz, Z. Mehmood, T. Nazir, R. A. Naqvi, A. Rehman, M. Iqbal, and T. Saba, “Skin cancer detection from dermoscopic images using deep learning and fuzzy kmeans clustering,” *Microscopy research and technique*, vol. 85, no. 1, pp. 339–351, 2022.
- [15] J. Daghrir, L. Tlig, M. Bouchouicha, and M. Sayadi, “Melanoma skin cancer detection using deep learning and classical machine learning techniques: A hybrid

approach,” in *2020 5th international conference on advanced technologies for signal and image processing (ATSIP)*. IEEE, 2020, pp. 1–5.

- [16] Z. Civelek and M. Kfashi, “An improved deep cnn for an early and accurate skin cancer detection and diagnosis system,” *International Journal of Engineering Research and Development*, vol. 14, no. 2, pp. 721–734, 2022.
- [17] M. Zafar, M. I. Sharif, M. I. Sharif, S. Kadry, S. A. C. Bukhari, and H. T. Rauf, “Skin lesion analysis and cancer detection based on machine/deep learning techniques: A comprehensive survey,” *Life*, vol. 13, no. 1, p. 146, 2023.
- [18] H. Ghosh, I. S. Rahat, S. N. Mohanty, J. Ravindra, and A. Sobur, “A study on the application of machine learning and deep learning techniques for skin cancer detection,” *International Journal of Computer and Systems Engineering*, vol. 18, no. 1, pp. 51–59, 2024.
- [19] P. Hermosilla, R. Soto, E. Vega, C. Suazo, and J. Ponce, “Skin cancer detection and classification using neural network algorithms: A systematic review,” *Diagnostics*, vol. 14, no. 4, p. 454, 2024.
- [20] N. Mukherjee, N. Dolzake, H. Ubhare, S. Sahu, S. Sharma, and S. Shirdhankar, “Melanoma espial employing deep learning applied to mobilenet.”
- [21] M. Zia Ur Rehman, F. Ahmed, S. A. Alsuhibany, S. S. Jamal, M. Zulfiqar Ali, and J. Ahmad, “Classification of skin cancer lesions using explainable deep learning,” *Sensors*, vol. 22, no. 18, p. 6915, 2022.
- [22] K. M. Hosny, M. A. Kassem, and M. M. Foaud, “Skin cancer classification using deep learning and transfer learning,” in *2018 9th Cairo international biomedical engineering conference (CIBEC)*. IEEE, 2018, pp. 90–93.
- [23] P. Ly, D. Bein, and A. Verma, “New compact deep learning model for skin cancer recognition,” in *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2018, pp. 255–261.
- [24] A. Naeem and T. Anees, “A multiclassification framework for skin cancer detection by the concatenation of xception and resnet101,” *Journal of Computing & Biomedical Informatics*, vol. 6, no. 02, pp. 205–227, 2024.

- [25] T. Guergueb and M. A. Akhloufi, “Melanoma skin cancer detection using recent deep learning models,” in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2021, pp. 3074–3077.
- [26] S. Barman, M. R. Biswas, S. Marjan, N. Nahar, M. S. Hossain, and K. Andersson, “Transfer learning based skin cancer classification using googlenet,” in *International Conference on Machine Intelligence and Emerging Technologies*. Springer, 2022, pp. 238–252.
- [27] A. Naeem, M. S. Farooq, A. Khelifi, and A. Abid, “Malignant melanoma classification using deep learning: datasets, performance measurements, challenges and opportunities,” *IEEE access*, vol. 8, pp. 110 575–110 597, 2020.
- [28] P. Ghosh, S. Azam, R. Quadir, A. Karim, F. J. M. Shamrat, S. K. Bhowmik, M. Jonkman, K. M. Hasib, and K. Ahmed, “Skinnet-16: A deep learning approach to identify benign and malignant skin lesions,” *Frontiers in Oncology*, vol. 12, p. 931141, 2022.
- [29] M. Kaleem, M. A. Mushtaq, S. A. Ramay, S. K. Hussain, M. Zohaib, M. Y. Hassan, N. Azam, and N. Ahmad, “Initial prediction of skin cancer using deep learning techniques: A systematic review,” *Journal of Computing & Biomedical Informatics*, vol. 5, no. 02, pp. 327–337, 2023.
- [30] R. H. Jatmiko and Y. Pristyanto, “Investigating the effectiveness of various convolutional neural network model architectures for skin cancer melanoma classification,” *MATRIX: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 23, no. 1, pp. 1–16, 2023.
- [31] N. Abuared, A. Panthakkan, M. Al-Saad, S. A. Amin, and W. Mansoor, “Skin cancer classification model based on vgg 19 and transfer learning,” in *2020 3rd International conference on signal processing and information security (ICSPIS)*. IEEE, 2020, pp. 1–4.
- [32] A. Pandya, K. Pandya, T. Upadhyaya, and U. Patel, “International journal of intelligent systems and applications in engineering,” *International Journal of IN-*

- [33] A. G. Diab, N. Fayed, and M. M. El-Seddek, “Accurate skin cancer diagnosis based on convolutional neural networks,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 3, pp. 1429–1441, 2022.
- [34] A. Magdy, H. Hussein, R. F. Abdel-Kader, and K. Abd El Salam, “Performance enhancement of skin cancer classification using computer vision,” *IEEE Access*, 2023.
- [35] A. Imran, A. Nasir, M. Bilal, G. Sun, A. Alzahrani, and A. Almuhaimeed, “Skin cancer detection using combined decision of deep learners,” *IEEE Access*, vol. 10, pp. 118 198–118 212, 2022.
- [36] N. Kausar, A. Hameed, M. Sattar, R. Ashraf, A. S. Imran, M. Z. u. Abidin, and A. Ali, “Multiclass skin cancer classification using ensemble of fine-tuned deep learning models,” *Applied Sciences*, vol. 11, no. 22, p. 10593, 2021.
- [37] M. H. Imam, N. Nahar, M. A. Rahman, and F. Rabbi, “Enhancing skin cancer classification using a fusion of densenet and mobilenet models: a deep learning ensemble approach,” *Multidisciplinary Science Journal*, vol. 6, no. 7, pp. 2024 117–2 024 117, 2024.
- [38] N. Girdhar, A. Sinha, and S. Gupta, “Densenet-ii: An improved deep convolutional neural network for melanoma cancer detection,” *Soft computing*, vol. 27, no. 18, pp. 13 285–13 304, 2023.
- [39] K. Ali, Z. A. Shaikh, A. A. Khan, and A. A. Laghari, “Multiclass skin cancer classification using efficientnets—a first step towards preventing skin cancer,” *Neuroscience Informatics*, vol. 2, no. 4, p. 100034, 2022.
- [40] E. Yilmaz and M. Trocan, “A modified version of googlenet for melanoma diagnosis,” *Journal of Information and Telecommunication*, vol. 5, no. 3, pp. 395–405, 2021.

- [41] V. Anand, S. Gupta, A. Altameem, S. R. Nayak, R. C. Poonia, and A. K. J. Saudagar, “An enhanced transfer learning based classification for diagnosis of skin cancer,” *Diagnostics*, vol. 12, no. 7, p. 1628, 2022.
- [42] N. H. Quang *et al.*, “Automatic skin lesion analysis towards melanoma detection,” in *2017 21st Asia Pacific symposium on intelligent and evolutionary systems (IES)*. IEEE, 2017, pp. 106–111.
- [43] T. H. Rafi and R. M. Shubair, “A scaled-2d cnn for skin cancer diagnosis,” in *2021 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, 2021, pp. 1–6.
- [44] K. Aljohani and T. Turki, “Automatic classification of melanoma skin cancer with deep convolutional neural networks,” *Ai*, vol. 3, no. 2, pp. 512–525, 2022.
- [45] M. Yıldırım and A. Çınar, “Classification of skin cancer images with convolutional neural network architectures,” *Turkish Journal of Science and Technology*, vol. 16, no. 2, pp. 187–195, 2021.
- [46] R. O. Ogundokun, A. Li, R. S. Babatunde, C. Umezuruike, P. O. Sadiku, A. T. Abdulahi, and A. N. Babatunde, “Enhancing skin cancer detection and classification in dermoscopic images through concatenated mobilenetv2 and xception models,” *Bioengineering*, vol. 10, no. 8, p. 979, 2023.
- [47] E. Alabdulkreem, H. Elmannai, A. Saad, I. S. Kamil, and A. Elaraby, “Deep learning-based classification of melanoma and non-melanoma skin cancer.” *Traitemen du Signal*, vol. 41, no. 1, 2024.
- [48] S. A. ElGhany, M. R. Ibraheem, M. Alruwaili, and M. Elmogy, “Diagnosis of various skin cancer lesions based on fine-tuned resnet50 deep network.” *Computers, Materials & Continua*, vol. 68, no. 1, 2021.
- [49] E. Hassan, F. M. Talaat, S. Adel, S. Abdelrazek, A. Aziz, Y. Nam, and N. El-Rashidy, “Robust deep learning model for black fungus detection based on gabor filter and transfer learning.” *Computer Systems Science & Engineering*, vol. 47, no. 2, 2023.

- [50] P. Singh, M. Kumar, and A. Bhatia, “A comparative analysis of deep learning algorithms for skin cancer detection,” in *2022 6th international conference on intelligent computing and control systems (iciccs)*. IEEE, 2022, pp. 1160–1166.
- [51] A. A. Adegun and S. Viriri, “Fcnn-based densenet framework for automated detection and classification of skin lesions in dermoscopy images,” *IEEE Access*, vol. 8, pp. 150 377–150 396, 2020.
- [52] L. Di Biasi, F. De Marco, A. Auriemma Citarella, M. Castrillón-Santana, P. Barra, and G. Tortora, “Refactoring and performance analysis of the main cnn architectures: using false negative rate minimization to solve the clinical images melanoma detection problem,” *BMC bioinformatics*, vol. 24, no. 1, p. 386, 2023.
- [53] J. SM, M. P, C. Aravindan, and R. Appavu, “Classification of skin cancer from dermoscopic images using deep neural network architectures,” *Multimedia Tools and Applications*, vol. 82, no. 10, pp. 15 763–15 778, 2023.
- [54] A. Kharb and P. Chaudhary, “Designing efficient brain tumor classifier using hybrid efficientnet-faster r-cnn deep learning model,” *Engineering Research Express*, vol. 6, no. 3, p. 035216, 2024.
- [55] K. M. Sünnetci, S. B. Akben, M. M. Kara, and A. Alkan, “Face mask detection using googlenet cnn-based svm classifiers,” *Gazi University Journal of Science*, vol. 36, no. 2, pp. 645–658, 2023.
- [56] R. Ali, A. Manikandan, R. Lei, and J. Xu, “A novel spasa based hyper-parameter optimized fcedn with adaptive cnn classification for skin cancer detection,” *Scientific Reports*, vol. 14, no. 1, p. 9336, 2024.
- [57] Y. Zhang, “Cooperative control method of robot formation movement path based on machine vision,” *Journal of Computational Methods in Sciences and Engineering*, vol. 22, no. 6, pp. 2093–2105, 2022.
- [58] P. Malik, A. Dureja, A. Dureja, R. S. Rathore, and N. Malhotra, “Enhancing intracranial hemorrhage diagnosis through deep learning models,” *Procedia Computer Science*, vol. 235, pp. 1664–1673, 2024.

- [59] A. Rajvanshi and S.-W. Chin, “Early detection of crop diseases using cnn classification,” 2023.
- [60] S. Patro, “Normalization: A preprocessing stage,” *arXiv preprint arXiv:1503.06462*, 2015.
- [61] M. Hasani and H. Khotanlou, “An empirical study on position of the batch normalization layer in convolutional neural networks,” in *2019 5th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*. IEEE, 2019, pp. 1–4.
- [62] Y. Yang, “Data augmentation to improve the diagnosis of melanoma using convolutional neural networks,” in *Proceedings of the 2021 international conference on bioinformatics and intelligent computing*, 2021, pp. 151–158.
- [63] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, “Image data augmentation for deep learning: A survey,” *arXiv preprint arXiv:2204.08610*, 2022.
- [64] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, “A comprehensive survey of image augmentation techniques for deep learning,” *Pattern Recognition*, vol. 137, p. 109347, 2023.
- [65] Z. Zhao, L. Alzubaidi, J. Zhang, Y. Duan, U. Naseem, and Y. Gu, “Robust and explainable framework to address data scarcity in diagnostic imaging,” *arXiv preprint arXiv:2407.06566*, 2024.
- [66] M. K. Doma, S. K. Sahu, and G. Raju, “Enhanced skin cancer classification through a hybrid optimized approach: Deep echo network machine utilizing pelican-optimized deep kohonen features,” *Traitemen du Signal*, vol. 40, no. 6, p. 2455, 2023.
- [67] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.
- [68] R. Bravin, L. Nanni, A. Loreggia, S. Brahnam, and M. Paci, “Varied image data augmentation methods for building ensemble,” *IEEE Access*, vol. 11, pp. 8810–8823, 2023.

- [69] F. A. BOUAZA Youcef, “Image filtering design and implementation based on xilinx system generator with hardware co-simulation and vhdl with fpga ip core generator,” Ph.D. dissertation, 2023.
- [70] N. M. Ibrahim, A. Abou ElFarag, and R. Kadry, “Gaussian blur through parallel computing.” in *IMPROVE*, 2021, pp. 175–179.
- [71] S. Birada, “Few-shot learning for animal identification: Enhancing prototypical networks with convolutional neural networks,” 2024.
- [72] L. Zhang, R. Shi, and N. Youssefi, “Oral cancer diagnosis based on gated recurrent unit networks optimized by an improved version of northern goshawk optimization algorithm,” *Heliyon*, 2024.
- [73] D. Pathak and U. S. N. Raju, “Content-based image retrieval for super-resolutioned images using feature fusion: Deep learning and hand crafted,” *Concurrency and Computation: Practice and Experience*, vol. 34, no. 22, p. e6851, 2022.
- [74] I. Corley, C. Robinson, R. Dodhia, J. M. L. Ferres, and P. Najafirad, “Revisiting pre-trained remote sensing model benchmarks: resizing and normalization matters,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 3162–3172.
- [75] P. Parsania, P. V. Virparia *et al.*, “A review: Image interpolation techniques for image scaling,” *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 12, pp. 7409–7414, 2014.
- [76] P. Mudjirahardjo, “The performance of histogram processing in real time.”
- [77] D. Lakshmi, K. P. Thanaraj, and M. Arunmozhi, “Convolutional neural network in the detection of lung carcinoma using transfer learning approach,” *International journal of imaging systems and technology*, vol. 30, no. 2, pp. 445–454, 2020.
- [78] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” *Artificial intelligence review*, vol. 53, pp. 5455–5516, 2020.

- [79] T. Adhikari, “Designing a convolutional neural network for image recognition: A comparative study of different architectures and training techniques,” *Available at SSRN 4366645*, 2023.
- [80] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, “Review of deep learning: concepts, cnn architectures, challenges, applications, future directions,” *Journal of big Data*, vol. 8, pp. 1–74, 2021.
- [81] Aphex34, “Own work,” 2015, cC BY-SA 4.0. [Online]. Available: <https://commons.wikimedia.org/w/index.php?curid=45679374>
- [82] Z. Khashroum, H. Rahimighazvini, and M. Bahrami, “Applications of machine learning in power electronics: A specialization on convolutional neural networks,” *ENG TRANSACTIONS*, vol. 4, no. 1, pp. 1–5, 2023.
- [83] 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.
- [84] D. V. Godoy, “Dl visuals,” 2023, cC BY 4.0, <https://commons.wikimedia.org/w/index.php?curid=150823503>. [Online]. Available: <https://github.com/dvgodoy/dl-visuals/>
- [85] S. Mascarenhas and M. Agarwal, “A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification,” in *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)*, vol. 1. IEEE, 2021, pp. 96–99.
- [86] O. F. S. Daraghmeh, “A hybrid artificial intelligence approach for early detection of breast cancer and classification from mammogram images in palestine,” Ph.D. dissertation, Al-Quds University, 2024.
- [87] T. Sugata and C. Yang, “Leaf app: Leaf recognition with deep convolutional neural networks,” in *IOP Conference Series: Materials Science and Engineering*, vol. 273, no. 1. IOP Publishing, 2017, p. 012004.

- [88] J. Tao, Y. Gu, J. Sun, Y. Bie, and H. Wang, “Research on vgg16 convolutional neural network feature classification algorithm based on transfer learning,” in *2021 2nd China international SAR symposium (CISS)*. IEEE, 2021, pp. 1–3.
- [89] Y. Zheng, C. Yang, and A. Merkulov, “Breast cancer screening using convolutional neural network and follow-up digital mammography,” in *Computational Imaging III*, vol. 10669. SPIE, 2018, p. 1066905.
- [90] J. Kurek, K. Szymanowski, L. Chmielewski, and A. Orłowski, “Advancing chip-board milling process monitoring through spectrogram-based time series analysis with convolutional neural network using pretrained networks,” *Machine Graphics and Vision*, vol. 32, no. 2, pp. 89–108, 2023.
- [91] J. Jaworek-Korjakowska, P. Kleczek, and M. Gorgon, “Melanoma thickness prediction based on convolutional neural network with vgg-19 model transfer learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.
- [92] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A survey of convolutional neural networks: analysis, applications, and prospects,” *IEEE transactions on neural networks and learning systems*, vol. 33, no. 12, pp. 6999–7019, 2021.
- [93] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.
- [94] M. Hemmer, H. Van Khang, K. G. Robbersmyr, T. I. Waag, and T. J. Meyer, “Fault classification of axial and radial roller bearings using transfer learning through a pretrained convolutional neural network,” *Designs*, vol. 2, no. 4, p. 56, 2018.
- [95] D. Raval and J. N. Undavia, “A comprehensive assessment of convolutional neural networks for skin and oral cancer detection using medical images,” *Healthcare Analytics*, vol. 3, p. 100199, 2023.

- [96] H. Yin, “Research on image classification algorithm based on convolutional neural network,” in *2023 International Conference on Electronics and Devices, Computational Science (ICEDCS)*. IEEE, 2023, pp. 299–303.
- [97] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [98] C. Y. Lo, F. C. Lau, and C.-W. Sham, “Fixed-point implementation of convolutional neural networks for image classification,” in *2018 International Conference on Advanced Technologies for Communications (ATC)*. IEEE, 2018, pp. 105–109.
- [99] A. Su, X. He, and X. Zhao, “Jpeg steganalysis based on resnext with gauss partial derivative filters,” *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 3349–3366, 2021.
- [100] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [101] A. S.-a. M. M.-I. A. Ahsan, S. M. Alif, J. B. Kibria, and P. E. Gomes, “Detection of skin cancer using convolutional neural network,” Ph.D. dissertation, Brac University, 2019.
- [102] Z. Guo, Q. Chen, G. Wu, Y. Xu, R. Shibasaki, and X. Shao, “Village building identification based on ensemble convolutional neural networks,” *Sensors*, vol. 17, no. 11, p. 2487, 2017.
- [103] Y. Song, H. He, Z. Zhang, J. Li, Z. Liu, and S. Gao, “Googlednet: Googlenet with dendritic learning for image classification,” in *2023 15th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. IEEE, 2023, pp. 41–44.
- [104] P. Ghimire, S. Piya, and A. M. Gurung, “Comparative study of face mask recognition using deep learning and machine learning classifiers,” in *2021 International*

*Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES).* IEEE, 2021, pp. 1–9.

- [105] M. N. Qureshi, M. S. Umar, and S. Shahab, “A transfer-learning-based novel convolution neural network for melanoma classification,” *Computers*, vol. 11, no. 5, p. 64, 2022.
- [106] E. Yilmaz and M. Trocan, “Benign and malignant skin lesion classification comparison for three deep-learning architectures,” in *Asian conference on intelligent information and database systems*. Springer, 2020, pp. 514–524.
- [107] N. Davari, G. Akbarizadeh, and E. Mashhour, “Corona detection and power equipment classification based on googlenet-alexnet: An accurate and intelligent defect detection model based on deep learning for power distribution lines,” *IEEE Transactions on Power Delivery*, vol. 37, no. 4, pp. 2766–2774, 2021.
- [108] K. He, X. Zhang, S. Ren *et al.*, “Deep residual learning,” *Image Recognition*, vol. 7, 2015.
- [109] H. Alaeddine and M. Jihene, “Deep residual network in network,” *Computational Intelligence and Neuroscience*, vol. 2021, no. 1, p. 6659083, 2021.
- [110] A. Kanavos and P. Mylonas, “Deep learning analysis of histopathology images for breast cancer detection: A comparative study of resnet and vgg architectures,” in *2023 18th International Workshop on Semantic and Social Media Adaptation & Personalization (SMAP) 18th International Workshop on Semantic and Social Media Adaptation & Personalization (SMAP 2023)*. IEEE, 2023, pp. 1–6.
- [111] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [112] Y. Wang, D. C. Louie, J. Cai, L. Tchvialeva, H. Lui, Z. J. Wang, and T. K. Lee, “Deep learning enhances polarization speckle for in vivo skin cancer detection,” *Optics & Laser Technology*, vol. 140, p. 107006, 2021.
- [113] A. Pandey, S. Degadwala, and D. Vyas, “A review on transfer learning approaches for skin melanoma classification,” 2022.

- [114] D. Singh, C. Aravinda, M. Kaur, M. Lin, J. Shetty, V. R. Reddicherla, and H.-N. Lee, “Dknet: Deep kuzushiji characters recognition network,” *IEEE Access*, vol. 10, pp. 75 872–75 883, 2022.
- [115] H.-Y. Chen and C.-Y. Su, “An enhanced hybrid mobilenet,” in *2018 9th International Conference on Awareness Science and Technology (iCAST)*. IEEE, 2018, pp. 308–312.
- [116] A. Wibowo, C. A. Hartanto, and P. W. Wirawan, “Android skin cancer detection and classification based on mobilenet v2 model,” *International Journal of Advances in Intelligent Informatics*, vol. 6, no. 2, pp. 135–148, 2020.
- [117] Wikidocs. (2023) Title of the page. [Online]. Available: <https://wikidocs.net/165429>
- [118] S. Alzahrani, B. Al-Bander, and W. Al-Nuaimy, “A comprehensive evaluation and benchmarking of convolutional neural networks for melanoma diagnosis,” *Cancers*, vol. 13, no. 17, p. 4494, 2021.
- [119] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.
- [120] Z. Wang, J. Guo, and S. Zhang, “Lightweight convolution neural network based on multi-scale parallel fusion for weed identification,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 36, no. 07, p. 2250028, 2022.
- [121] W. W. Lo, X. Yang, and Y. Wang, “An xception convolutional neural network for malware classification with transfer learning,” in *2019 10th IFIP international conference on new technologies, mobility and security (NTMS)*. IEEE, 2019, pp. 1–5.
- [122] K. Srinivasan, L. Garg, D. Datta, A. A. Alaboudi, N. Jhanjhi, R. Agarwal, and A. G. Thomas, “Performance comparison of deep cnn models for detecting driver’s distraction,” 2021.

- [123] A. Panthakkan, S. Anzar, S. Al Mansoori, W. Mansoor, and H. Al Ahmad, “A systematic comparison of transfer learning models for covid-19 prediction,” *Intelligent Decision Technologies*, vol. 16, no. 3, pp. 557–574, 2022.
- [124] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
- [125] T. Zhou, X. Ye, H. Lu, X. Zheng, S. Qiu, and Y. Liu, “Dense convolutional network and its application in medical image analysis,” *BioMed Research International*, vol. 2022, no. 1, p. 2384830, 2022.
- [126] H. Alshazly, C. Linse, M. Abdalla, E. Barth, and T. Martinetz, “Covid-nets: deep cnn architectures for detecting covid-19 using chest ct scans,” *PeerJ Computer Science*, vol. 7, p. e655, 2021.
- [127] X. Feng, H. Yao, and S. Zhang, “An efficient way to refine densenet,” *Signal, Image and Video Processing*, vol. 13, pp. 959–965, 2019.
- [128] M. Tan, “Efficientnet: Rethinking model scaling for convolutional neural networks,” *arXiv preprint arXiv:1905.11946*, 2019.
- [129] D. Ene, V. Anireh, D. Matthias, and E. Bennett, “Leveraging efficientnet and amortized stochastic variational inference for improved transfer learning in vaes in mobile and resource-constrained environments.”
- [130] P. R. Oza, P. Sharma, and S. Patel, “A transfer representation learning approach for breast cancer diagnosis from mammograms using efficientnet models,” *Scalable Computing: Practice and Experience*, vol. 23, no. 2, pp. 51–58, 2022.
- [131] W. A. R. Abdalla, “Brain tumor classification using efficientnet-b1: A deep learning approach,” *African Journal of Advanced Pure and Applied Sciences (AJAPAS)*, pp. 603–613, 2024.
- [132] M. Tan and Q. V. Le. (2019) Efficientnet: Improving accuracy and efficiency through automl and model scaling. [Online]. Available: <https://research.google/blog/efficientnet-improving-accuracy-and-efficiency-through-automl-and-model-scaling/>

- [133] M. Bhargavi, R. Renugadevi, S. Sivabalan, P. Phani, J. Ganesh, and K. Bhanu, “Ensemble learning for skin lesion classification: A robust approach for improved diagnostic accuracy (elslc),” in *2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. IEEE, 2023, pp. 390–395.
- [134] O. Islam, M. Assaduzzaman, and M. Z. Hasan, “An explainable ai-based blood cell classification using optimized convolutional neural network,” *Journal of Pathology Informatics*, vol. 15, p. 100389, 2024.
- [135] X. Yang, “Quantum fuzzy neural network based on fuzzy number,” *Frontiers in Computing and Intelligent Systems*, vol. 3, no. 2, pp. 99–105, 2023.
- [136] P. M. M. da Silva, E. R. Vieira, E. Morya, and F. Azevêdo, “5.4 artigo 04: Machine learning-based on type 2 diabetes detection using spatiotemporal and pressure distribution gait parameters,” *Aprendizagem de máquina aplicada à execução da marcha em diabéticos tipo 2*, p. 96, 2023.
- [137] A. Hazra, N. Bera, A. Mandal *et al.*, “Predicting lung cancer survivability using svm and logistic regression algorithms,” *International Journal of Computer Applications*, vol. 174, no. 2, pp. 19–24, 2017.
- [138] N. N. Sultana and N. B. Puhan, “Recent deep learning methods for melanoma detection: a review,” in *Mathematics and Computing: 4th International Conference, ICMC 2018, Varanasi, India, January 9-11, 2018, Revised Selected Papers 4*. Springer, 2018, pp. 118–132.
- [139] P. Bansal, R. Garg, and P. Soni, “Detection of melanoma in dermoscopic images by integrating features extracted using handcrafted and deep learning models,” *Computers & Industrial Engineering*, vol. 168, p. 108060, 2022.
- [140] A. Bhattacharyya, D. Bhaik, S. Kumar, P. Thakur, R. Sharma, and R. B. Pachori, “A deep learning based approach for automatic detection of covid-19 cases using chest x-ray images,” *Biomedical Signal Processing and Control*, vol. 71, p. 103182, 2022.

- [141] D. E. Hill, “Errors of judgment and reporting in a law merchant system,” *Theory and decision*, vol. 56, no. 3, p. 239, 2004.
- [142] D. H. Lundgren, S.-I. Hwang, L. Wu, and D. K. Han, “Role of spectral counting in quantitative proteomics,” *Expert review of proteomics*, vol. 7, no. 1, pp. 39–53, 2010.
- [143] N. Ternès, F. Rotolo, and S. Michiels, “Empirical extensions of the lasso penalty to reduce the false discovery rate in high-dimensional cox regression models,” *Statistics in medicine*, vol. 35, no. 15, pp. 2561–2573, 2016.
- [144] M. Oumoulyte, A. O. Alaoui, Y. Farhaoui, A. El Allaoui, and A. Bahri, “Convolutional neural network-based approach for skin lesion classification,” *Data and Metadata*, vol. 2, pp. 171–171, 2023.
- [145] X. Li, Y. Lin, C. Gu, and Z. Li, “Srmdap: Simrank and density-based clustering recommender model for mirna-disease association prediction,” *BioMed research international*, vol. 2018, no. 1, p. 5747489, 2018.
- [146] X. Hou, C. Ju, and B. Wang, “Prediction of solar irradiance using convolutional neural network and attention mechanism-based long short-term memory network based on similar day analysis and an attention mechanism,” *Heliyon*, vol. 9, no. 11, 2023.
- [147] C. Li, Q. Min, Y. Cheng, Y. Yuan, and G. Wang, “Local semantic structure captured and instance discriminated by unsupervised hashing.” *Int. J. Softw. Informatics*, vol. 11, no. 1, pp. 55–67, 2021.
- [148] N. Ratyal, I. A. Taj, M. Sajid, A. Mahmood, S. Razzaq, S. H. Dar, N. Ali, M. Usman, M. J. A. Baig, and U. Mussadiq, “Deeply learned pose invariant image analysis with applications in 3d face recognition,” *Mathematical Problems in Engineering*, vol. 2019, no. 1, p. 3547416, 2019.
- [149] U. Hameed, M. Ur Rehman, A. Rehman, R. Damaševičius, A. Sattar, and T. Saba, “A deep learning approach for liver cancer detection in ct scans,” *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 11, no. 7, p. 2280558, 2024.

- [150] J. Chu, Q. Zhang, R. Gao, F. Wang, and X. Xin, “Resnet equalizer for oam mode divsion multiplexing obtical fiber communication system,” in *2023 IEEE 11th International Conference on Information, Communication and Networks (ICICN)*. IEEE, 2023, pp. 73–78.
- [151] X. Zheng and R. S. Cloutier, “A review of image classification algorithms in iot,” *EAI Endorsed Transactions on Internet of Things*, vol. 7, no. 28, 2022.
- [152] H. Yi, H. Wang, A. Shu, and J. Huang, “Changeable environment visual detection of grinding surface roughness based on lightweight network,” *Nondestructive Testing and Evaluation*, pp. 1–24, 2024.
- [153] T. Wang, Z. Liu, and L. Liu, “Investigating a three-dimensional convolution recognition model for acoustic emission signal analysis during uniaxial compression failure of coal,” *Geomatics, Natural Hazards and Risk*, vol. 15, no. 1, p. 2322483, 2024.
- [154] W. An, P. Zhang, J. Xu, H. Luo, L. Huang, and S. Zhong, “A novel machine learning aided antenna selection scheme for mimo internet of things,” *Sensors*, vol. 20, no. 8, p. 2250, 2020.
- [155] A. Abdulazeez *et al.*, “A review on utilizing machine learning classification algorithms for skin cancer,” *Journal of Applied Science and Technology Trends*, vol. 5, no. 2, pp. 60–71, 2024.
- [156] “Isic archive: Collection 212,” <https://api.isic-archive.com/collections/212/>, accessed: [insert the date you accessed the website].
- [157] I. Razzak, G. Shoukat, S. Naz, and T. M. Khan, “Skin lesion analysis toward accurate detection of melanoma using multistage fully connected residual network,” in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.
- [158] “Isic 2017: Skin lesion analysis towards melanoma detection challenge,” <https://challenge.isic-archive.com/landing/2017/>, accessed: [insert the date you accessed the website].

- [159] A. Yilmaz, M. Kalebasi, Y. Samoylenko, M. E. Guvenilir, and H. Uvet, “Benchmarking of lightweight deep learning architectures for skin cancer classification using isic 2017 dataset,” *arXiv preprint arXiv:2110.12270*, 2021.
- [160] B. Cassidy, C. Kendrick, A. Brodzicki, J. Jaworek-Korjakowska, and M. H. Yap, “Analysis of the isic image datasets: Usage, benchmarks and recommendations,” *Medical image analysis*, vol. 75, p. 102305, 2022.
- [161] §. Öztürk and U. Özkaya, “Skin lesion segmentation with improved convolutional neural network,” *Journal of digital imaging*, vol. 33, pp. 958–970, 2020.
- [162] S. P. Kothapalli, P. S. H. Priya, V. S. Reddy, B. Lahya, and P. Ragam, “Melanoma skin cancer detection using svm and cnn,” *EAI Endorsed Transactions on Pervasive Health and Technology*, vol. 9, 2023.

## **Final Year Design Project as a Complex Engineering Problem**

It is to certify here that the final year design project (FYDP) entitled,

*“AI Based Skin Cancer Detection Algorithms: Challenges, Countermeasures and Way forward”*

is categorized as a complex engineering problem (CEP) based on the preamble (in-depth engineering knowledge) and involvement of the following attributes.

1. Depth of knowledge required
2. Depth of analysis required
3. Range of conflicting requirements
4. Consequences
5. Interdependence

The above listed attributes are thoroughly assessed after conducting meeting on 3<sup>rd</sup> November 2023, with the following final year students, who proposed the idea of the titled FYDP.

1. Name: Tayyaba Arooj  
Reg No: ECI-IT-20-139
2. Name: Mahjabeen Sadeeq  
Reg No: ECI-IT-20-092

This project is going to be conducted in fall semester 2023 and spring semester 2024. Further, it is submitted that the proposed idea is worthy, and the required efforts are up to the level of a final year design project.

FYDP Advisor

## 1.7 Complex Engineering Problem

This project satisfies the attributes of the complex engineering problem, in the given context, this section presents the justification that how the presented work addresses different attributes of the complex engineering problem. The details are presented in Table Error! *No text of specified style in document...*

Table Error! *No text of specified style in document..1 CEP Attributes Mapping*

Sr. No	Attribute	Justification
1	Depth of knowledge required	The project requires in-depth knowledge of convolutional neural networks (CNNs), medical imaging analysis, data preprocessing, and key machine learning evaluation metrics relevant to healthcare applications.
2	Range of conflicting requirements	Achieving high accuracy while minimizing false positives and false negatives are conflicting requirements, as they affect the clinical applicability and reliability of the model in skin cancer diagnosis.
3	Depth of analysis required	Detailed analysis of CNN architectures is necessary to optimize the model's performance. Comparative studies on recall, precision, and specificity metrics are essential for selecting the best model.
4	Consequences	Improved early detection of skin cancer, leading to better patient outcomes, highlights the project's success. Reducing diagnostic errors can significantly impact public health and patient care quality.
5	Interdependence	Model performance, computational cost, and interpretability are interdependent factors. Optimizing one may impact the others, especially in the context of clinical deployment where resource constraints exist.

## Sustainable Development Goals

This section presents a brief overview of all the SDGs and mainly justifies the contribution of the project to the sustainable development goals (SDGs). Detailed justification of the mentioned points is presented in Table Error! *No text of specified style in document...*

Table Error! *No text of specified style in document..2 SGD Table of the Project*

Sr. No	Title	Compliance (Y/N)	Remarks/Justification
1	No poverty	No	Not applicable
2	Zero hunger	No	Not applicable
3	Good health/wellbeing	Yes	Early and accurate detection of skin cancer improves health outcomes and supports public health by reducing mortality and morbidity associated with late-stage diagnosis.
4	Quality education	Yes	The project promotes awareness about AI applications in healthcare, enhancing educational insights for medical professionals and students in diagnostic technologies.
5	Gender equality	No	Not applicable
6	Clean water and sanitation	No	Not applicable
7	Affordable and clean energy	No	Not applicable
8	Decent work and economic Growth	yes	Potential to reduce healthcare costs by enabling earlier detection, which could decrease treatment costs and improve

			the economic efficiency of healthcare systems.
9	Industry, innovation and Infrastructure	Yes	Contributes to healthcare innovation through AI and diagnostic technologies, enhancing medical infrastructure and supporting AI adoption in medical industries.
10	Reduced Inequalities	Yes	Improves accessibility to early skin cancer detection in underserved regions, potentially addressing disparities in healthcare access and outcomes.
11	Sustainable Cities and Communities	Yes	Improved healthcare accessibility and diagnostics contribute to the sustainability of healthcare services in communities, promoting healthier populations.
12	Responsible consumption and Production	Yes	The project encourages responsible healthcare by facilitating early diagnosis, reducing the need for more resource-intensive treatments at later stages.
13	Climate action	No	Not applicable
14	Life below water	No	Not applicable
15	Life on land	Yes	Supports healthy lives on land by enabling better management of skin health and potentially improving survival rates for skin cancer patients.
16	Peace, Justice and strong Institutions	No	Not applicable
17	Partnerships for the goals	Yes	Potential for partnerships with

			healthcare providers and AI research institutions to expand the impact and accessibility of AI in medical diagnostics.
--	--	--	--