

**Project/Thesis No:**

## **Skin Lesion Classification using Convolutional Neural Network**

**By**

**Abdullah Al Shafi**

Roll: 1807004



Department of Computer Science and Engineering

Khulna University of Engineering & Technology

Khulna 9203, Bangladesh

**February 2024**

# Skin Lesion Classification using Convolutional Neural Network

By

**Abdullah Al Shafi**

Roll: 1807004

A thesis submitted in partial fulfillment of the requirements for the degree of  
"Bachelor of Science in Computer Science and Engineering"

Supervisor:

Dr. Pintu Chandra Shill

Professor

Department of Computer Science and Engineering  
Khulna University of Engineering & Technology  
Khulna, Bangladesh



---

Signature

Department of Computer Science and Engineering  
Khulna University of Engineering & Technology  
Khulna 9203, Bangladesh

**February 2024**

## Acknowledgement

All the praise to the Almighty Allah, whose blessing and mercy succeeded me to complete this thesis work fairly. I gratefully acknowledge the valuable suggestions, advice and sincere co-operation of Dr. Pintu Chandra Shill, Professor, Department of Computer Science and Engineering, Khulna University of Engineering & Technology, under whose supervision this work was carried out. His intellectual advice, encouragement and guidance make me feel confident and scientific research needs much effort in learning and applying and need to have a broad view at problems from different perspective. I would like to convey my heartiest gratitude to all the faculty members, official and staffs of the Department of Computer Science and Engineering as they have always extended their co-operation to complete this work. Last but not least, I wish to thank my friends and family members for their constant supports.

**Author**

## Abstract

Skin cancer is one of the most common diseases that can be initially detected by visual observation and further with the help of dermoscopic analysis and other tests. The survival rate can be significantly increased if it is identified at an early stage. On the other hand, the classification of skin lesions is incredibly challenging. At an initial stage, visual observation gives the opportunity of utilizing artificial intelligence to intercept the different skin images, so several skin lesion classification methods using deep learning based on convolutional neural network (CNN) and annotated skin photos exhibit improved results, which is lifesaving in terms of diagnosis. The study investigates the application of deep learning techniques, specifically Convolutional Neural Networks (CNNs), in the automated classification of skin lesions for early detection of skin diseases. Leveraging CNNs for skin lesion classification can significantly improve the accuracy and efficiency of diagnosis. For the recognition network, lesions are classified into multiple classes using an ensemble of MobileNetV2, VGG19 and InceptionV3 against three datasets, namely, HAM10000, ISIC-2016 and ISIC-2019 datasets. Initially, rebalancing and augmentation of the skin images are performed. Then classification is done through ensemble of Convolutional Neural Networks. The proposed ensemble method demonstrates praiseworthy success in lesion recognition, providing an accuracy of 87.53%, 84.86%, and 85.92% respectively for those three datasets. The classification performance is measured with different well known measures designed for skin lesion detection and recognition mentioned in the result section and the result is appreciable.

## Contents

<b>Title Page</b>	<b>i</b>
<b>Acknowledgement</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>x</b>
<b>List of Abbreviations</b>	<b>xi</b>
<b>CHAPTER I: Introduction</b>	<b>1</b>
1.1 Introduction	1
1.2 Background	1
1.3 Motivation	3
1.4 Problem Statement	4
1.5 Objectives	4
1.6 Scope of the Thesis	5
1.7 Contribution	5
1.8 Project planning	6
1.9 Application of the work	6
1.10 Organization of the thesis	7
<b>CHAPTER II: Literature Review</b>	<b>8</b>
2.1 Introduction	8
2.2 Literature Review	8
2.3 Discussion	13
<b>CHAPTER III: Background Analysis</b>	<b>15</b>
3.1 Introduction	15
3.2 Skin	15

3.3	Skin Lesion	15
3.4	Skin Cancer Types	16
3.4.1	Malignant Skin Lesions	17
3.4.2	Benign Skin Lesion	17
3.5	ABCDE Rule	18
3.6	Data Augmentation Using Affine Image Transformations	19
3.6.1	Flip and Rotation	20
3.6.2	Translation	20
3.6.3	Scaling and Cropping	21
3.6.4	Shearing	21
3.7	Efficient training with transfer learning	21
3.8	Convolutional Neural Network	22
3.8.1	CNN architecture	23
3.9	Separable Convolution	25
3.9.1	Spatially Separable Convolutions	26
3.9.2	Depthwise Separable Convolutions	27
3.9.2.1	Depthwise convolution	27
3.9.2.2	Pointwise convolution	28
3.10	MobileNetV2 Model	28
3.10.1	Depthwise Separable Convolutions	29
3.10.2	Linear Bottleneck	29
3.10.3	Inverted residuals	31
3.10.4	MobileNetV2 convolutional Blocks	32
3.10.5	Architecture	33
3.11	VGG19 Model	34
3.11.1	Convolutional layers	35
3.11.2	Activation layers	36
3.11.3	Pooling layers	36
3.11.4	Fully connected layers	37
3.12	InceptionV3 Model	37
3.13	Technical Issue	38

<b>CHAPTER IV: Methodology</b>	<b>40</b>
4.1 Introduction	40
4.2 Proposed method	40
4.2.1 Image classification	40
4.2.2 Data Pre-processing	41
4.2.2.1 Rebalancing	42
4.2.2.2 Data augmentation	42
4.2.3 Convolutional Neural Network	43
4.3 Proposed Classification Network	44
4.4 Conclusion	46
<b>CHAPTER V: Implementation, Results and Discussions</b>	<b>47</b>
5.1 Introduction	47
5.2 Experimental Setup	47
5.2.1 Python	47
5.2.2 Google Colab	48
5.2.3 Pandas	48
5.2.4 NumPy	49
5.2.5 Matplotlib	49
5.2.6 Keras and Tensorflow	49
5.3 Evaluation Metrics	49
5.4 Datasets	52
5.5 Implementation and Results	54
5.5.1 Quantitative results	54
5.5.1.1 Training and validation graph	54
5.5.1.2 Confusion Matrix	54
5.5.1.3 Qualitative Classification Result	54
5.5.2 Qualitative results	59
5.5.3 Analysis of the result	59
5.6 Objective Achieved	61
5.7 Conclusion	61

<b>CHAPTER VI: Societal,Health,Environment,Safety,Ethical,Legal and Cultural Issues</b>	<b>62</b>
6.1 Intellectual Property Consideration	62
6.2 Ethical Consideration	62
6.3 Safety Consideration	63
6.4 Legal Consideration	63
6.5 Impact of the project on Societal, Health and Cultural Issues	64
6.5.1 Societal Impact	64
6.5.2 Health Impact	64
6.5.3 Cultural Impact	64
6.6 Impact of Project on the Environment and Sustainability	65
6.6.1 Environmental Impact	65
6.6.2 Sustainability Impact	65
6.6.3 Waste Reduction	65
<b>CHAPTER VII:Addressing Complex Engineering Problems and Activities</b>	<b>66</b>
7.1 Complex engineering problems associated with the current project/thesis	66
7.2 Complex engineering activities associated with the current project/thesis	67
<b>CHAPTER VIII:Conclusions</b>	<b>68</b>
8.1 Summary	68
8.2 Limitations	68
8.3 Recommendations and Future Works	68
<b>References</b>	<b>70</b>

## List of Figures

<b>Figure No.</b>	<b>Description</b>	<b>Page</b>
1.1	Dermatology statistics in 2020 and forecast for 2040 [2].	1
1.2	Work plan using the Gantt Chart.	6
3.1	Skin cancer affected skin image [30].	16
3.2	Classification of skin lesions [32].	17
3.3	Lesion diagnosis by dermatologists(ABCD criteria for lesion diagnosis focuses on finding the certain properties of lesions) [33].	19
3.4	An example of the transfer learning technique [36].	22
3.5	Simple CNN architecture[37].	23
3.6	Max pool with 2x2 filters and stride 2[38].	25
3.7	Convolutions with spatially separable kernels[39].	26
3.8	Depthwise Separable Convolutions [40].	27
3.9	MobileNetV2 [41].	28
3.10	The difference between residual block and inverted residual [21]. Diagonally hatched layers do not use non-linearities. We use thickness of each block to indicate its relative number of channels. Note how classical residuals connects the layers with high number of channels, whereas the inverted residuals connect the bottlenecks. Best viewed in color.	31
3.11	MobileNetV2 convolutional Blocks[41].	32
3.12	MobileNetV2 Architecture[41].	33
3.13	VGG19 Neural Network Architecture.[45].VGG-19 has 16 convolution layers grouped into 5 blocks. After every block, there is a Maxpool layer that decreases the size of the input image by 2 and increases the number of filters of the convolution layer also by 2. The dimensions of the last three dense layers in block 6 are 4096, 4096, and 1000 respectively.	34
3.14	ReLU6 Activation Function.	36
3.15	InceptionV3 architecture with 11 inception blocks.	38
3.16	Inception module with dimensionality reduction.	38

4.1	Overview of proposed method for skin lesions classification.	41
4.2	Applied different augmentations based on geometric transformations, such as rotation, and flipping.	42
4.3	The difference between the original skin lesion image (left) and the images after data augmentation (right).	43
4.4	Illustration of the backbone network (Vgg-19) used to classify skin cancer images. First, the image features are extracted by the CNN feature extractor. Next, these features are reduced by the pooling layer. Finally, the softmax layer serves as the output layer of the model to predict skin lesion classes.	44
5.1	Work plan using the Gantt Chart.	50
5.2	Overview of HAM10000 Dataset.	52
5.3	Overview of ISIC 2019 Dataset.	53
5.4	Training and validation (a) accuracy on MobileNetV2 (b) loss on MobileNetV2 (c) accuracy on VGG19 (d) loss on VGG19 (e) accuracy on InceptionV3 and (f) loss on InceptionV3 for HAM10000[58] dataset.	55
5.5	Training and validation (a) accuracy on MobileNetV2 (b) loss on MobileNetV2 (c) accuracy on VGG19 (d) loss on VGG19 (e) accuracy on InceptionV3 and (f) loss on InceptionV3 for ISIC-16[59] dataset.	56
5.6	Training and validation (a) accuracy on MobileNetV2 (b) loss on MobileNetV2 (c) accuracy on VGG19 (d) loss on VGG19 (e) accuracy on InceptionV3 and (f) loss on InceptionV3 for ISIC-19[60] dataset.	57
5.7	Confusion Matrix for (q) ISIC-16 (b) ISIC-19 (c) HAM10000.	58
5.8	Sample prediction from images of (a)HAM10000 [58] (b) ISIC-19 [60] dataset. The predicted labels are outside the parenthesis and the ground truth are inside the parenthesis.Blue indicates the predicted label and ground truth label are same i.e. right prediction and red indicates they are different i.e. wrong prediction.	60

## List of Tables

<b>Table No.</b>	<b>Description</b>	<b>Page</b>
2.1	A brief summary of skin lesion classification	14
3.1	Bottleneck residual block transforming from k to k' channels, with stride s, and expansion factor t [41]	33
4.1	Summary of various architectures	45
5.1	Summary of the datasets used in the experiments	52
5.2	Quantitative classification results on different test datasets using MobileNetV2,VGG-19, InceptionV3 and an ensemble method	59
5.3	Comparison of the classification results with the related methods	61
7.1	Complex engineering problems associated with the current project/thesis	66
7.2	Complex engineering activities associated with the current project/thesis	67

## List of Abbreviations

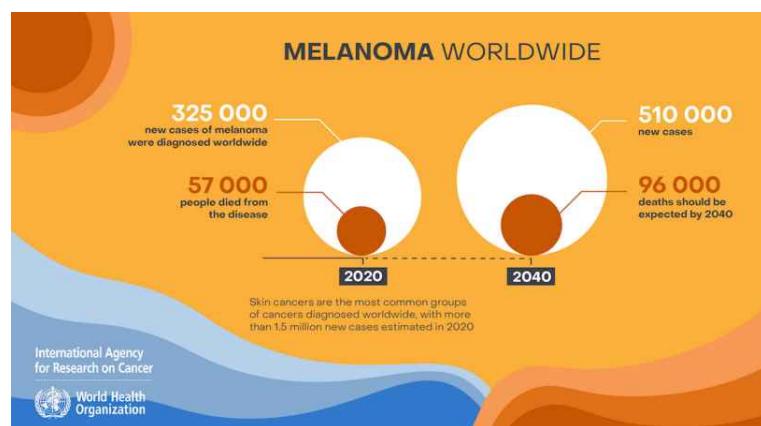
<b>CNN</b>	Convolutional Neural Network
<b>TL</b>	Transfer Learning
<b>ISIC</b>	International Skin Imaging Collaboration
<b>VGG</b>	Visual Geometry Group
<b>SLC</b>	Skin Lesion Classification
<b>w/o</b>	WithOut
<b>DNN</b>	Deep Neural Network

## Chapter I

### Introduction

#### 1.1 Introduction

Skin cancer is one of the most widespread and fatal cancer types globally. It generally develops due to exposure to ultraviolet (UV) rays from the sun, which harms the DNA of skin cells [1]. Some artificial sources of light, in particular tanning beds and sunlamps, increase the risk of developing this disease. In 2022, an estimated 99,780 adults (57,180 men and 42,600 women) in the United States will have been diagnosed with invasive melanoma of the skin [2]. Worldwide, there approximately 324,635 people were newly diagnosed with melanoma in 2020 and 57,000 people died from the disease [2]. A new study by scientists from the International Agency for Research on Cancer (IARC) and partners predicts that the number of new cases of cutaneous melanoma per year will increase by more than 50% from 2020 to 2040 [2].



**Figure 1.1:** Dermatology statistics in 2020 and forecast for 2040 [2].

#### 1.2 Background

Skin cancer is a common type of cancer that originates in the skin's epidermis layer by the irregular cells due to ultraviolet radiation exposure [3]. Cancer forms vary, in which skin cancer has the highest chance of occurrences with a significant malignancy risk [4]. As

the most common cancer type, skin cancer is generally classified into melanoma and non-melanoma categories [5]. Skin lesion classification is a critical task in dermatology, with accurate diagnosis being essential for timely and effective treatment of skin conditions. The increasing prevalence of skin cancer, combined with the growing demand for cost-effective and efficient healthcare solutions, has driven the development of computer-aided diagnosis systems for skin lesion classification.

However, image-based automated (CAS) systems are highly challenging for the following hurdles: Wide range of intra-class variance in colors, textures, edges, and shapes and homogeneity in inter-classes. Sometimes, low contrasts and unclear boundaries (edges) in the malignant and other class images. Color skin images contain features related to skin lesion (ROI) and background (noise and skin surface) which make the process of extracting ROI only a hard and tedious task. Lesion ROI frequently shares similar visual characteristics and subtle distinctions due to lighting, perspective, and spatial information within an image. The appearance of different artifacts, such as natural (hairs, veins) or synthetic (air bubbles, ruler lines, color balance charts, marker signs, paint, ink color, artificial objects, etc.), LED lighting, and darker borders. Unavailability of a large number of manually annotated images, which is the core requirement of supervised learning systems. An automated system to detect skin lesion type consists of three main stages: segmentation, feature extraction, and classification. In segmentation, the cancerous part is separated from the background of the dermoscopic image. In the next stage, the features of this segmented foreground are extracted then classification is performed with the help of extracted features to detect true skin lesion type.

However, traditional methods for skin lesion classification, such as the need for manual feature engineering, the limited availability of annotated medical images, and difficulty in handling large variations in lesion appearance. The problem of skin lesion classification is an important task in dermatology, as it can help with early detection and diagnosis of various skin conditions. With the increasing availability of medical imaging data, there has been a growing interest in using machine learning techniques to assist healthcare professionals in making more accurate and faster diagnoses.

The use of deep learning models for skin lesion classification, such as Convolutional Neural Networks (CNNs), have shown impressive performance in a range of computer vision tasks,

including image classification and segmentation. In this context, the use of CNNs for skin lesion classification has several advantages, including automatic feature extraction, improved accuracy and the ability to handle large variations in lesion appearance.

However, the limited availability of annotated medical images is a major challenge in the development of deep learning models for skin lesion classification. To overcome this challenge, transfer learning, a technique that leverages knowledge from pre-trained models, has been widely used in the development of deep learning models for skin lesion classification. Transfer learning allows deep learning models to be trained on large datasets, even in the absence of annotated medical images, by fine-tuning pre-trained models on smaller datasets which can assist healthcare professionals in making more informed decisions, by providing additional information and improving the accuracy of diagnoses.

### 1.3 Motivation

The automated classification of skin lesions is still a challenging task because of the great visual similarity between melanoma and benign lesions. The first problems are the intra-class differences and interclass similarities of skin lesions, which cause many difficulties in the identification of malignant and benign skin lesions. Since variation in the appearances of skin lesions is small, local, and subtle, fine-grained global context information is essential for skin lesion recognition.

Skin cancer is on the rise without a corresponding increase in the number of dermatologists. The good news is that diagnosis of skin diseases plus CAD can greatly assist non-expert diagnosis. Medical diagnosis in remote areas is becoming increasingly urgent, so remote diagnosis will also become a trend. Deep learning methods have also grown dramatically in the field of dermatology in recent years, but most of them are based on large volumes of data. However, collecting high-quality medical images is not easy in hospitals, and gaining research approval of applications for research is even more difficult. I initially targeted small datasets; over time, I eventually targeted large data sets and multiple classifications, where the extreme imbalance of data also affects the accuracy of classification. Improving the accuracy of skin disease classification is an endless pursuit. All the abovementioned considerations motivated us to carry out this research project.

## 1.4 Problem Statement

Skin cancer is one of the most prevalent forms of cancer in the United States as well as in the world. Melanoma is a lethal form of skin cancer, and it is among the most common cancer types. However, survival rates are high if detected and diagnosed early. Traditional machine learning and deep learning can play a significant role in skin cancer Diagnosis. Though image segmentation and classification itself is a difficult task, the skin lesions image segmentation and classification task has its unique challenges as it has high inter-class similarities. As it is a part of real-life patient diagnosis, precision has to be as high as possible. However, the number of well-established biomedical image datasets is less than needed. There often exists high class imbalance problems in the training data. One has to tackle all those challenges to solve a biomedical image segmentation and classification problem successfully.

## 1.5 Objectives

The Convolution Neural Network(CNN) is used in this thesis for skin lesion classification.

The study will be carried out with the following precise aims to achieve the goal to:

- Research models in the biomedical field, such as the convolutional neural network model.
- Study the data preprocessing on skin lesion images.
- Design a robust system for skin lesions image classification model.
- Classify properly using an ensemble of CNN architectures as it is hard to distinguish due to inter-class similarities.
- Investigate the model's performance in terms of accuracy, sensitivity, and specificity.
- Compare the results with the performance of dermatologists in a controlled study.

## 1.6 Scope of the Thesis

This study covers both theoretical and implementational aspects of developing a computer aided diagnosis system for efficient biomedical image classification. The study aims to improve the accuracy of skin lesion classification through the use of an ensemble of convolutional neural networks and to contribute to the field of medical image analysis. So that it can be used in real-time and in sophisticated environments.

## 1.7 Contribution

In this work, I propose an ensemble Convolutional Neural Network with transfer learning for skin lesion classification. The proposed model leverages transfer learning and an ensemble approach to enhance its accuracy. It has shown state-of-the art performance on benchmark datasets and has potential applications in computer-aided diagnosis systems. The contributions of this work can be summarized as follows:

- Develop an ensembling model using Convolutional Neural Networks with transfer learning.
- Apply the proposed method for classifying seven different skin lesions with the HAM10000 dataset and eight different skin lesions with the ISIC-2019 dataset.
- Apply geometry-based image augmentations, the class rebalancing techniques to protect the classifier from being biased towards any particular class with more sample.
- Demonstrate state-of-the-art lesion detection and recognition results, to our best knowledge, on two IEEE International Symposium on Biomedical Imaging (ISBI) datasets, such as ISIC-2019(includes ISIC-2018 and ISIC-2017) and HAM10000 having a different number of classes

## 1.8 Project planning

Task Name	1st term							2nd term						
	1-2	3-4	5-6	7-8	9-10	11-12	13	1-2	3-4	5-6	7-8	9-10	11-12	13
<i>Topic Selection</i>	■													
<i>Thesis Planning</i>		■												
<i>Literature Review</i>			■	■										
<i>Dataset collection &amp; preprocessing</i>					■									
<i>Predefense Report Manuscript</i>						■								
<i>Predefense Presentation</i>							■		■					
<i>Segmentation &amp; augmentation</i>								■	■					
<i>CNN model construction</i>									■	■				
<i>Model Finalizing</i>										■				
<i>Final Model Evaluation</i>											■			
<i>Thesis report manuscript</i>											■			
<i>Thesis Defense</i>												■		

**Figure 1.2:** Work plan using the Gantt Chart.

Figure 5.1 shows the work plan with the help of gantt chart.

In the first term,I selected the topic of the thesis after analyzing several topics.In the following two weeks,I planned about how I can do the thesis.In the next four week,I collected several research paper on skin lesion classification and then I spent two weeks for dataset collection and data preprocessing.I spent next two weeks for preparing predefense report manuscript.I utilized the next following week for pre-defense presentation.In the second term,I did the data augmentation in the first two weeks.Then I used next four weeks for CNN model construction.I then utilized the next two weeks for final model evaluation.I spent next four weeks to prepare thesis report manuscript.I utilized the last week for defense presentation.

## 1.9 Application of the work

skin cancer tends to be deadly. The survival rate can be significantly increased if the skin lesions are identified in dermoscopic images at an early stage. On the other hand, the classification of skin lesions is incredibly challenging. Skin lesion classification using deep learning approaches has provided better results in classifying skin diseases than those of dermatologists, which is lifesaving in terms of diagnosis.The study investigates the application of deep learning techniques, specifically Convolutional Neural Networks (CNNs), in the automated

classification of skin lesions for early detection of skin diseases. Leveraging CNNs for skin lesion classification can significantly improve the accuracy and efficiency of diagnosis.

### **1.10 Organization of the thesis**

The rest of the thesis is organized as follows:

- **Chapter-II:** In this chapter, some previous works related to the skin lesion classification problem are reviewed. It includes the previously proposed solutions as well as the modern solutions.
- **Chapter-III:** In this chapter, some theoretical prospects for developing the method are presented. The drawbacks of existing methods and how to overcome them are also discussed.
- **Chapter-IV:** In this chapter, the proposed methodology and how the system works are demonstrated. The implementation aspect of our proposed system are also described.
- **Chapter-V:** In this chapter, our proposed methodology and individual model with different experiments were analyzed and the results were demonstrated.
- **Chapter-VI:** In this chapter, societal, health, environment, safety, ethical, legal and cultural issues related to the thesis are described.
- **Chapter-VII:** In this chapter, complex engineering problems and activities associated with the thesis were discussed.
- **Chapter-VIII:** In this chapter, the whole thesis is summarized, the limitation and future plan are discussed, and finally the thesis is concluded.

## Chapter II

### Literature Review

#### 2.1 Introduction

In this chapter, Some similar classification-based works are also examined. This chapter further explains how the suggested solution overcomes the limitations of existing methods.

#### 2.2 Literature Review

Several skin lesion classification techniques exist in the literature—using either conventional or deep methods.

Chowdhury et al. [6] utilized pre-trained deep learning models, namely MobileNetV2 and DenseNet201, to categorize skin diseases into two classes: benign and malignant. This was achieved by utilizing the International Skin Imaging Collaboration (ISIC) archive dataset [6]. The models were modified by appending convolutional layers at the end of the both models to detect the skin cancer effectively. Furthermore, the final classification layer and the classification head were also modified. The modified models exhibited superior performance compared to the original pre-trained MobileNetV2 and DenseNet201 models. Both the classes can be detected using the modified models. The Modified DenseNet201 model attains an accuracy of 95.50%, sensitivity of 93.96%, and specificity of 97.03%. With minor modifications, the model can be employed for diagnosing skin cancer across multiple categories.

Khan et al. [8] presented a novel technique based on probabilistic distribution and feature selection for skin lesion detection and classification. Normal and uniform distributions are implemented to segment the lesion area. Later, the features are extracted from the segmented images, which are finally fused using a parallel fusion strategy. For feature selection, the entropy-based technique is combined with Bhattacharyya distance and variance formulation. The proposed technique is evaluated on three publicly available datasets, including combined ISBI 2016 and ISBI 2017, ISIC, and PH2, achieving an accuracy of 93.2% 97.75% and 97.5% respectively.

Tschandl et al. [7] trained a fully convolutional neural network (CNN) on the ISIC 2017 dataset and reused the ResNet34 layers to segment skin lesions. Pre-training and fine-tuning of ResNet34 improves the segmentation performance; therefore, the mentioned steps were embedded.

Chowdhury et al. [8] employed a custom Convolutional Neural Network (CNN) to recognize 7 classes of skin diseases through the HAM10000 dataset [2]. Class Activation Mapping (CAM) was used as an XAI method. This study aimed to explore whether the features extracted from deep models like convolutional networks, self-attention models, and attention as activation models have a correlation with clinically significant features. Three main categories of raw pixel-oriented machine learning algorithms, which include convolution, spatial self-attention, and attention as activation, were examined and contrasted with the Asymmetry, Border, Color, Diameter (ABCD) skin lesion clinical characteristics-based machine learning algorithms, employing both qualitative and quantitative interpretations. A visual analysis was done to see if the activation maps of deep models were similar to the segmentation maps used for clinical feature extraction. The maximum achieved accuracy is 82.7% and 78% of precision.

Nunnari et al. [9] presented on how saliency maps can be used to identify areas of interest in the diagnosis of skin cancer. VGG16 and ResNet-50 are used to classify 8 skin classes and Gradient-weighted Class Activation Mapping (Grad-CAM) was used as an explainable method. The two neural architectures distinguished by varying layers and distinct resolution of the final convolutional layer, measure how effectively thresholded Grad-CAM saliency maps can detect visual attributes of skin cancer. Research indicates that saliency maps subjected to a threshold can achieve a success rate of nearly 50%. The optimal threshold for obtaining the highest Jaccard index differs significantly depending on the features. Furthermore, the Jaccard index attained was as high as 0.143, which is approximately half the performance of cutting-edge architectures designed for predicting feature masks at the pixel level, like U-Net. The models attain an accuracy of 72.2% and 76.7% respectively.

Kassem et al. [10] put forth a scheme for categorizing eight types of skin diseases by utilizing the ISIC 2019 dataset. Timely and exact identification of skin abnormalities can be a matter of life and death. To accomplish this, the model uses GoogleNet pre-trained model and transfer learning. The model effectively categorized Melanoma, melanocytic nevus, basal

cell carcinoma, actinic keratosis, benign keratosis, dermatofibroma, and Squamous cell carcinoma. The accuracy, sensitivity, specificity, and precision percentages of the classification were 94.92%, 79.8%, 97%, and 80.36%, respectively. The model can also detect images that do not belong to any of the eight classes.

Mahbod et al. [11] explored the image resizing effect on pre-trained CNN models to classify skin lesions. The images were resized on six different scales to investigate the classification results of three CNN architectures, namely SeReNeXt-50, EfficientNetB0, and EfficientNetB1. They also developed and evaluated a multi-scale multi-CNN (MSM-CNN) fusion technique based on the ensemble method. This approach utilized the three CNN models trained on cropped images of different sizes. The MSM-CNN technique achieved an 86.2accuracy on the ISIC 2018 dataset. They also concluded that image cropping yields better results as compared to image resizing.

To classify the skin lesion images into seven classes, the researchers in [12] proposed a deep CNN architecture. They implemented GoogleNet and Inception-V3 to perform binary classification. They improved the accuracy by up to 7% for the multiclass problem.

Chaturvedi et al. [13] presented an automated classification system for multiclass skin cancer. They performed extensive experiments on pre-trained CNN models including Xception, NASNetLarge, Inception-V3, InceptionResNet-V2, and ResNetXt-101 and the ensembles of these models. These CNN architectures were fine-tuned on seven classes of the HAM10000 dataset using transfer learning. This proposed model achieved an accuracy of 93.2ResNetXt-101 model. The accuracy achieved on the ensemble of InceptionResNet-V2 and ResNetXt-101 was 92.83

Al-Masni et al. [14] presented a hybrid model for multiple skin lesion classification and segmentation. A full-resolution convolutional network (FrCN) was utilized for the segmentation of lesion parts. Deep CNN classification was performed on the segmented skin lesions. The presented technique was validated on three challenging skin datasets, ISIC 2016, ISIC 2017, and ISIC 2018, with proper data normalization of these datasets and achieved improved results.

Xie et al. [15] presented a mutual bootstrapping deep convolutional neural network (MB-DCNN) for efficient detection and classification of skin lesions. A coarse segmentation

network was utilized for enhanced segmentation of lesions and a mask-guided network was implemented for classification. Bootstrapping coarse segmentation networks and enhanced segmentation networks played a vital role in segmentation and classification. The features of both networks were concatenated for efficient detection and classification. The proposed technique was validated on the ISIC 2017 and PH2 datasets and achieved mean area under the curve (AUC) values of 93.8% and 97.7% respectively.

Jayapariya et al. [16] introduced a fully convolutional network-based model for melanoma detection. The VGG16 [17] and GoogleNet [18] deep CNN models were used for the segmentation of lesions, followed by the feature extraction step. The deep CNN-extracted features were fused for accurate segmentation. Later, they extracted handcrafted features and concatenated them with a deep vector. The SVM was added at the end for a final classification. The presented model was tested on the challenging ISIC 2016 and ISIC 2017 datasets, with an accuracy of 0.8892 and 0.853.

Xie et al. [19] focused on spatial features for lesions' segmentation by introducing a high resolution CNN model. In this model, they extracted deep features without affecting the spatial attributes and decreasing the noise effect. The proposed model robustly segmented the lesions by overcoming the artifacts and hair distraction.

Miglani et al. [20] compared the performance of deep CNN models for the robust classification of skin lesions. Transfer learning was performed using ResNet-50 [21] and EfficientNet [22] by fine-tuning their parameters. The HAM1000 dataset was utilized to validate the performance of the deep CNNs. EfficientNet-Bo outperformed ResNet-50 by achieving macro and micro AUC value of 0.93 and 0.97 for skin lesion classification.

Mahbod et al. [23] proposed using multiple pre-trained CNNs with different architectures that are fine-tuned on dermoscopic skin lesion images. The deep features acquired from each CNN were used to train different SVM classifiers. Finally, the prediction probability classification vectors were fused to provide a final prediction. The proposed method achieved an 87.3% AUC using the skin lesion images from the ISIC 2017 dataset. Finally, Mahbod et al. [24] analyzed the impact of various segmentation masks but observed no significant difference between using manually or automatically created segmentation masks on the images from the ISIC 2017 dataset.

Bisla et al. [25] proposed a new deep-learning system for lesion segmentation and classification. The first network, a U-Net, segments the lesion region in the image, while the second network, a ResNet-18[26], classifies the lesion. They obtained an ROC AUC of 0.88 for melanoma classification against ISIC 2017 Dataset.

Khan et al. [27] proposed a method for skin lesion segmentation and multiclass classification using deep learning features and improved Moth Flame Optimization (MOFO). The performance of the proposed method on two publicly available datasets: ISIC 2018 and PH2. The results show that the proposed method achieved an average Dice Similarity Coefficient (DSC) of 0.92 and 0.93 for lesion segmentation on the ISIC 2018 and PH2 datasets, respectively, which outperforms the state-of-the-art methods by a significant margin. In terms of classification, the proposed method achieved an average accuracy of 95.73ISIC 2018 dataset and 97.06% on the PH2 dataset, which again outperforms other methods. The authors conclude that the combination of deep learning features and improved MOFO provides a promising approach for skin lesion segmentation and classification.

Thapar et al. [28] presented multiple deep learning techniques to accurately segment and classify skin lesions which involve combining a U-Net-based segmentation network and a ResNet-based classification network. The approach was evaluated on two publicly available datasets - ISIC 2018 and PH2, and the results showed that the proposed method achieved an average dice coefficient of 0.87 for skin lesion segmentation and an average accuracy of 90.0% for lesion classification. These results were found to be superior compared to other state-of-the-art approaches, demonstrating the effectiveness of the hybrid deep learning approach for skin lesion analysis.

Anindya et al.[29] highlighted the potential of transfer learning for improving the performance of deep learning models in medical image analysis.UNet with a pre-trained VGG16 model used for magnetic resonance imaging (MRI) based brain tumor segmentation and found 96.1% accuracy in the learning dataset.

### 2.3 Discussion

The discussed survey is summarized in Table 2.1. It illustrates that despite technical advances and numerous research works being carried on in the field of image processing, a considerable scope of improvement still exists in the field of skin lesion image processing.

Further, timely detection and categorization of skin lesions to offer timely medical attention also requires highly accurate classification methods. To achieve this, the classification methods used by various researchers have been discussed in this section that guides and motivates the author for the involvement of ensemble of CNN to improve the overall skin lesion classification.

**Table 2.1:** A brief summary of skin lesion classification

Authors	Techniques	Dataset(s)	Performance Measure	Remarks
Tschandl et al. [7]	ResNet34 for segmentation and CNN for classification	ISIC 2017	class predictions of the CNN (blue line; 76.9% )	limited accuracy and small range of Data availability.
Mahbod et al. [11]	multi-CNN (MSM-CNN) fusion technique based on the ensemble method	ISIC 2018	MSM-CNN technique achieved an 86.2% accuracy	Limited Dataset and Lack of comparison with other methods
Al-Masni et al. [14]	FrCN for segmentation and Deep CNN classification	ISIC 2016, ISIC 2017, and ISIC 2018	81.27% fo ISIC 2016, 81.34% fo ISIC 2017, and 88.70% fo ISIC 2018	low accuracy, may not perform well for rare or uncommon skin lesions
Xie et al. [15]	Mutual bootstrapping DCNN	ISIC 2017 and PH2 datasets	8area under the curve (AUC) values of 93.8% and 97.7% respectively	Limited Dataset and only 2 or 3 classes of lesion
Jayapariya et al. [16]	VGG16 and GoogleNet for segmentation and SVM for final classification	ISIC 2016 and ISIC 2017	accuracy of 0.8892 and 0.853, respectively.	Limited Dataset and low accuracy
Miglani et al. [19]	Transfer learning using ResNet-50 and EfficientNet-B0	HAM1000	AUC value of 0.93 and 0.97 for skin lesion classification	No segmentation method applied
Mahbod et al. [21]	pre-trained CNNs and SVM classifiers	ISIC 2017	achieved an 87.3% AUC	Limited imbalanced Dataset and cover a smaller range of lesion classes.
bisla et al. [23]	U-Net for segmenting the lesion region and ResNet-18 for classifying the lesion	ISIC 2017	obtained ROC AUC of 0.88 for melanoma classification	limited imbalanced dataset and cover a smaller range of lesion classes.
Khan et al. [24]	multiclass classification using deep learning features and improved by Moth Flame Optimization (MOFO)	ISIC 2018 and PH2	Dice Similarity Coefficient (DSC) of 0.92 and 0.93 for lesion segmentation	Has great performance in multimodal biomedical images but computationally costly

## Chapter III

### Background Analysis

#### 3.1 Introduction

In this chapter, the motivations and theoretical considerations of our proposed method are discussed. The basic notion of skin lesion and its classification are introduced in this chapter. The CNN and used models are then briefly discussed.

#### 3.2 Skin

The skin is the outer covering of the body and is the largest organ of the integumentary system. The skin has up to seven layers of ectodermal tissue guarding muscles, bones, ligaments and internal organs. Human skin is similar to most of the other mammals' skin, and it is very similar to pig skin. Though nearly all human skin is covered with hair follicles, it can appear hairless. There are two general types of skin, hairy and glabrous skin (hairless). The adjective cutaneous literally means "of the skin" (from Latin *cutis*, skin).

Skin plays an important immunity role in protecting the body against pathogens and excessive water loss. Its other functions are insulation, temperature regulation, sensation, synthesis of vitamin D, and the protection of vitamin B folates. Severely damaged skin will try to heal by forming scar tissue. This is often discoloured and depigmented.

In humans, skin pigmentation (affected by melanin) varies among populations, and skin type can range from dry to non-dry and from oily to non-oily. Such skin variety provides a rich and diverse habitat for bacteria that number roughly 1000 species from 19 phyla, present on the human skin.

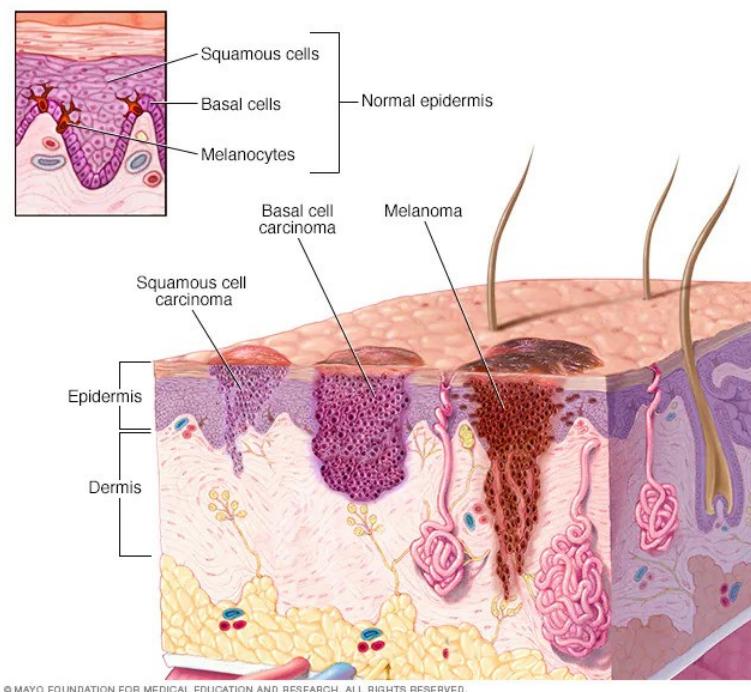
#### 3.3 Skin Lesion

A skin lesion refers to any skin area that has different characteristics from the surrounding skin, including color, shape, size, and texture. Skin lesions are very common and often appear as a result of a localized damage to the skin, like sunburns or contact dermatitis. Others, however, can be manifestations of underlying disorders, such as infections, diabetes,

and autoimmune or genetic disorders. Although most skin lesions are benign and harmless, some of them can be malignant or premalignant, meaning they have the potential to evolve into skin cancer.

### 3.4 Skin Cancer Types

The occurrence of skin cancer is associated with UV ray exposure from the sun, contributing to skin cells' DNA impairment. Gene mutations would trigger when DNA is damaged, whereby skin cells multiply excessively, leading to tumors' formation. Fig. 3.1 illustrates



**Figure 3.1:** Skin cancer affected skin image [30].

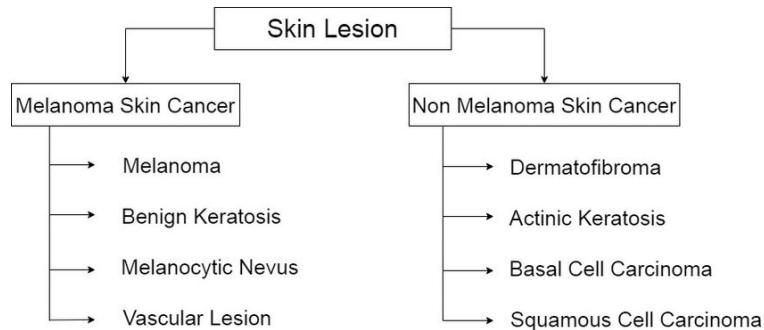
the effect of skin cancer on the skin. Lesions on the skin could be due to different causes such as allergies, cancerous cells, etc [31]. Lesions on the skin could be due to different causes such as allergies, cancerous cells, etc. However, among these, skin lesions due to cancerous cells are perilous. In the worst case, certain manifestations of cancerous skin lesions are fatal. Among cancer-related lesions, melanoma contributes to an 8rate, which is considered a significantly high rate[32].

Skin cancer is classified into two categories:

- Melanoma Skin Cancer (MSC).

- Non-Melanoma Skin Cancer (NMSC).

The following are eight categories of cancerous skin lesions shown in Fig. 3.2:



**Figure 3.2:** Classification of skin lesions [32].

### 3.4.1 Malignant Skin Lesions

A malignant skin lesion is, by definition, skin cancer. The two main types of skin cancer are keratinocyte carcinoma and melanoma. Each type of skin cancer has unique characteristics, but general signs of skin cancer can include rapidly growing skin lesions, changes in the color or size of a preexisting lesion, or a scabbing sore that doesn't heal with time.

Keratinocyte carcinoma arises from skin cells called keratinocytes, and includes basal cell carcinoma and squamous cell carcinoma. Basal cell carcinoma can appear as a pearly, flesh-colored skin lesion, with superficial blood vessels called telangiectasias on top. Basal cell carcinoma may present as a superficial scaling plaque, or a non-healing sore, which may bleed or form a crust. Conversely, squamous cell carcinoma commonly appears as a thick, crusty sore, with a reddish, inflamed base that can ulcer (appear as an open sore) and bleed. Melanoma arises from skin cells called melanocytes. Melanoma typically looks like an abnormal or irregular mole. The main warning signs of melanoma can be assessed using the ABCDE rule. ABCDE stands for Asymmetry, Border irregularities, Color heterogeneity, Diameter over 6 mm, and Evolution, which refers to changes in size, color, or shape over time. The presence of one or more of these features indicates a higher chance of malignancy.

### 3.4.2 Benign Skin Lesion

A benign skin lesion is a non-cancerous skin abnormality, growth, or tumor that can occur anywhere on the body. Benign lesions can manifest in a number of different ways, depending

on their cause and tissue of origin. Common benign skin lesions include most melanocytic nevi, better known as moles, seborrheic keratoses, skin tags, cherry angiomas, and lipomas, among others. Most of the time, these lesions are harmless and don't require treatment, unless they cause symptoms such as discomfort or itching.

Unlike malignant lesions, benign skin lesions are generally symmetrical, well-circumscribed, have a uniform appearance, and are stable or grow slowly over time. However, in certain cases, it can be difficult to distinguish between benign and malignant lesions; in those cases, a biopsy or surgical removal of the affected area can be performed to rule out malignancy.

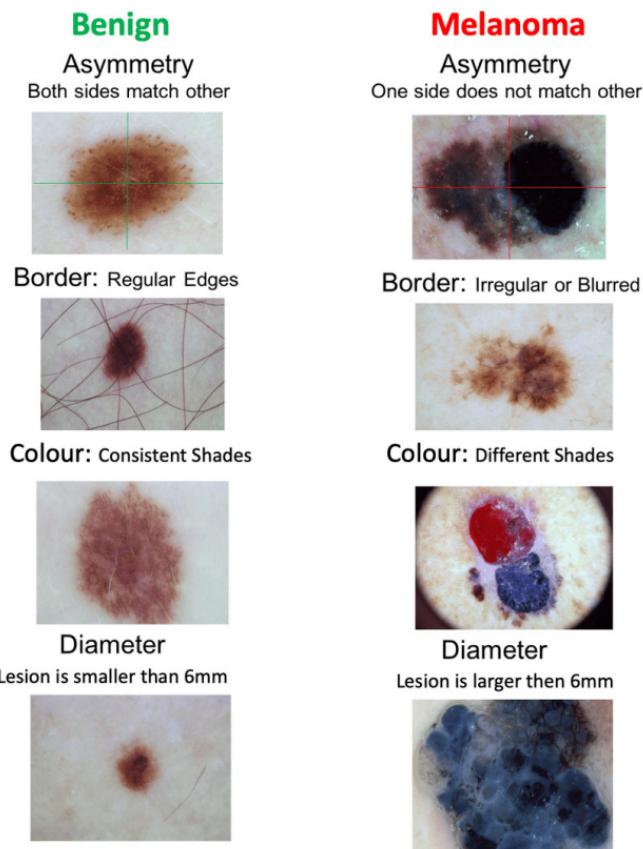
Benign lesions should also be distinguished from premalignant lesions such actinic keratosis or lentigo maligna, which present an increased risk of developing into different types of skin cancer. Both actinic keratosis and lentigo maligna occur as a result of long term unprotected sun exposure. Actinic keratosis appears as dry, scaly patches of skin over sun-exposed areas, like the nose and forehead, whereas lentigo maligna takes the appearance of localized dark-brown or black lesions, predominantly on the face and trunk.

### 3.5 ABCDE Rule

ABCDE rule has been generally utilized to diagnose skin cancers, as following [33] clinically:

- **A: Asymmetry property.** Two halves of skin lesions are assessed for similarity in aspects of edges, shape, and color.
- **B: Border property.** Skin lesion edges are assessed in appearance to see if they are well-defined and smooth. Otherwise, the lesion is likely to be melanoma if the edges are jagged, fuzzy, and uneven.
- **C: Color property.** Melanoma visibly shows color contrast between different skin regions with shades varying from black, red, brown, and tan.
- **D: Diameter property.** Skin lesion diameter that exceeds 6mm generally tells a sign of melanoma.
- **E: Evolution property.** Melanoma will often change characteristics, such as size, shape or color. Unlike most benign moles, melanoma tends to change over time. If you have a mole or skin growth, watch it for signs of changes.

The ABCDE rule is typically carried out by conducting a physical assessment of skin with the naked eye. Despite this, there have been cases whereby melanoma appears identical to benign lesions while following the ABDCE rule (see Fig. 3.3), leading to erroneous judgment. On the other hand, different clinical assessments such as biopsy are prone to errors. Specialists' accuracy in predicting whether a skin lesion is malignant or otherwise falls in the range of 49% to 81%, where a third of melanomas have been inaccurately declared as benign lesions. These low accuracies have prompted the application of dermoscopy [33].



**Figure 3.3:** Lesion diagnosis by dermatologists(ABCD criteria for lesion diagnosis focuses on finding the certain properties of lesions) [33].

### 3.6 Data Augmentation Using Affine Image Transformations

Data augmentation is a popular technique which helps improve generalization capabilities of deep neural networks, and can be perceived as implicit regularization. It plays a pivotal role in scenarios in which the amount of high-quality ground-truth data is limited, and acquiring

new examples is costly and time-consuming. In the affine approaches, actual image data undergo different operations (rotation, zooming, cropping, flipping, or translations) to increase the number of training examples [34]. Shin et al. [35] pointed out that such traditional data augmentation techniques fundamentally produce very correlated images, therefore can offer very few improvements for the deep-network training process and future generalization over the unseen test data (such examples do not regularize the problem sufficiently).

### 3.6.1 Flip and Rotation

Random flipping creates a mirror reflection of an original image along one (or more) selected axis. Usually, natural images can be flipped along the horizontal axis, which is not the case for the vertical one because up and down parts of an image are not always “interchangeable.” For example, rotating an image by an angle  $\phi$  around the center pixel can be exploited in this context. This operation is followed by an appropriate interpolation to fit the original image size. The rotation operation denoted as  $R$  is often coupled with zero-padding applied to the missing pixels:

$$R_\phi = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.1)$$

### 3.6.2 Translation

The translation operation shifts the entire image by a given number of pixels in a chosen direction while applying padding accordingly. It allows the network to not become focused on features present mainly in one particular spatial region, but it forces the model to learn spatially-invariant features instead.

$$T = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \quad (3.2)$$

The translating factors are given as  $t_x$  and  $t_y$  for the x and y directions, respectively

### 3.6.3 Scaling and Cropping

Introducing scaled versions of the original images into the training set can help the deep network learn valuable deep features independently of their original scale. This operation  $S$  can be performed independently in different directions (for brevity, we have only two dimensions here):

$$S = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.3)$$

and the scaling factors are given as  $s_x$  and  $s_y$  for the x and y directions, respectively.

### 3.6.4 Shearing

The shear transformation ( $H$ ) displaces each point in an image in a selected direction. This displacement is proportional to its distance from the line which goes through the origin and is parallel to this direction:

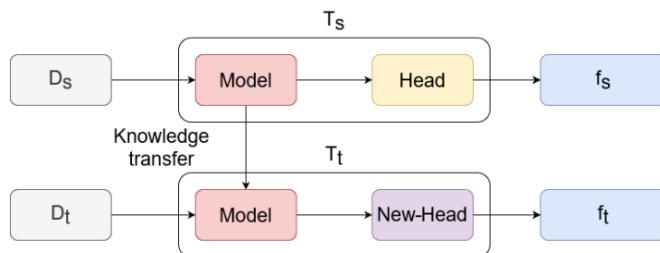
$$H = \begin{pmatrix} 1 & h_x & 0 \\ h_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.4)$$

where  $h_x$  and  $h_y$  denote the shear coefficient in the x and y directions, respectively.

## 3.7 Efficient training with transfer learning

Machine learning models are frequently designed to solve specific tasks and work alone. In these cases, the model must be built and trained from scratch, which necessitates a large amount of data. Millions of samples may be required depending on the type of problem to be solved or the type of model. In general, complex problems necessitate a large amount of data, but the situation becomes more complicated when the data must be labeled for supervised learning. It is extremely difficult to label a large amount of data, which may necessitate the use of time, tools, and people, and even once completed, there is no guarantee that all of the labels are correct. To solve this problem, researchers came up with the idea of transfer learning. Transfer learning is a research problem in machine learning (ML) that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. That means transfer learning is a technique that stores the knowledge gained from

one task and uses it to solve a related but different problem. Let, a base domain is  $D_s$  and its corresponding learning task is  $T_s$ , a target domain is  $D_t$  and its corresponding learning task is  $T_t$ . Transfer learning aims to help improve the learning of the target predictive function, say,  $f_t$  in  $T_t$  using the knowledge in  $D_s$  and  $T_s$  [36]. Figure 3.4 shows a visualization of such event. Here, a mathematical model (the Model added with the Head) is used to learn the base task  $T_s$ . Then the pre-trained Model part is isolated and added with the New-Head to create a new model and learn the target task  $T_t$ . Transfer learning can enhance the robustness of the model and helps a deep learning model to learn faster. In our case, we used this technique to learn primitive-level feature extraction, like edge or structure of real life objects, from a vast resource of images. This knowledge is then used to initialize the further learning of segmenting biomedical images.



**Figure 3.4:** An example of the transfer learning technique [36].

### 3.8 Convolutional Neural Network

A Convolutional Neural Network (CNN) is a type of Deep Learning neural network architecture commonly used in Computer Vision.[37] Computer vision is a field of Artificial Intelligence that enables a computer to understand and interpret the image or visual data. When it comes to Machine Learning, Artificial Neural Networks perform really well. Neural Networks are used in various datasets like images, audio, and text. Different types of Neural Networks are used for different purposes, for example for predicting the sequence of words we use Recurrent Neural Networks more precisely an LSTM, similarly for image classification we use Convolution Neural networks. In this blog, we are going to build a basic building block for CNN.

In a regular Neural Network there are three types of layers:

**Input Layers:** It's the layer in which we give input to our model. The number of neurons in this layer is equal to the total number of features in our data (number of pixels in the case of

an image).

**Hidden Layer:** The input from the Input layer is then feed into the hidden layer. There can be many hidden layers depending upon our model and data size. Each hidden layer can have different numbers of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of output of the previous layer with learnable weights of that layer and then by the addition of learnable biases followed by activation function which makes the network nonlinear.

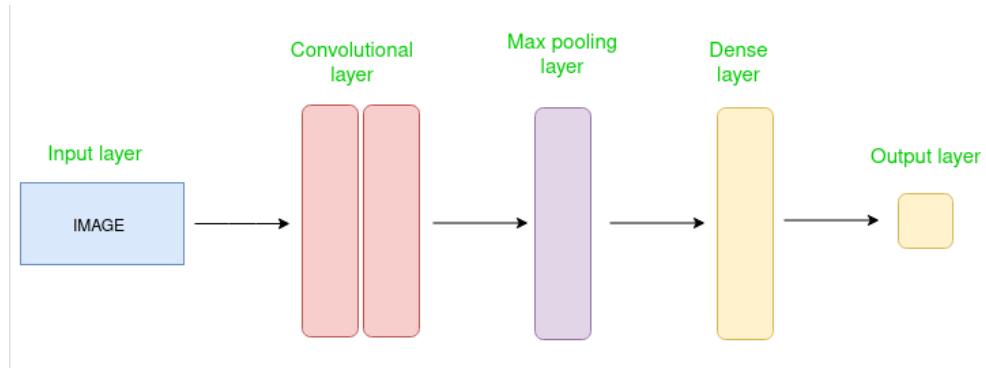
**Output Layer:** The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into the probability score of each class.

The data is fed into the model and output from each layer is obtained from the above step is called feedforward, we then calculate the error using an error function, some common error functions are cross-entropy, square loss error, etc. The error function measures how well the network is performing. After that, we backpropagate into the model by calculating the derivatives. This step is called Backpropagation which basically is used to minimize the loss.

Convolutional Neural Network (CNN) is the extended version of artificial neural networks (ANN) which is predominantly used to extract the feature from the grid-like matrix dataset. For example visual datasets like images or videos where data patterns play an extensive role.

### 3.8.1 CNN architecture

Convolutional Neural Network consists of multiple layers like the input layer, Convolutional layer, Pooling layer, and fully connected layers. The Convolutional layer applies filters to



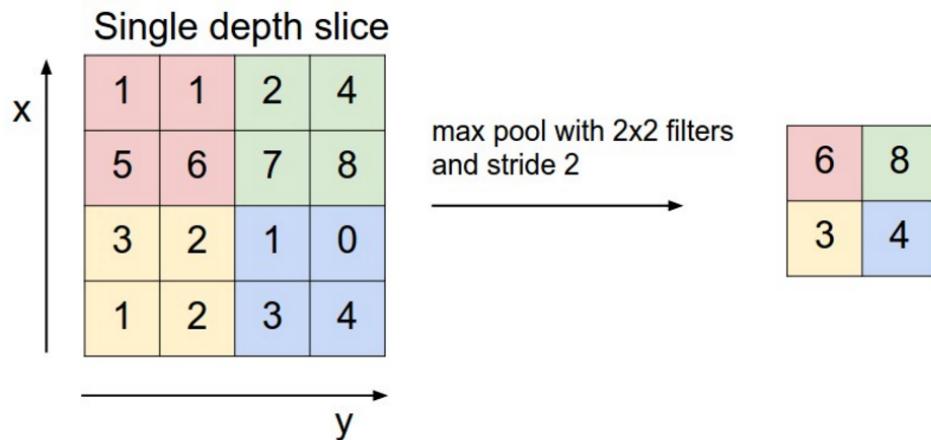
**Figure 3.5:** Simple CNN architecture[37].

the input image to extract features, the Pooling layer downsamples the image to reduce computation, and the fully connected layer makes the final prediction. The network learns the optimal filters through backpropagation and gradient descent.

A complete Convolution Neural Networks architecture is also known as covnets. A covnets is a sequence of layers, and every layer transforms one volume to another through a differentiable function.

Let's take an example by running a covnets on of image of dimension 32 x 32 x 3.

- **Input Layers:** It's the layer in which we give input to our model. In CNN, Generally, the input will be an image or a sequence of images. This layer holds the raw input of the image with width 32, height 32, and depth 3.
- **Convolutional Layers:** This is the layer, which is used to extract the feature from the input dataset. It applies a set of learnable filters known as the kernels to the input images. The filters/kernels are smaller matrices usually  $2\times 2$ ,  $3\times 3$ , or  $5\times 5$  shape. it slides over the input image data and computes the dot product between kernel weight and the corresponding input image patch. The output of this layer is referred ad feature maps. Suppose we use a total of 12 filters for this layer we'll get an output volume of dimension 32 x 32 x 12.
- **Activation Layer:** By adding an activation function to the output of the preceding layer, activation layers add nonlinearity to the network. it will apply an element-wise activation function to the output of the convolution layer. Some common activation functions are RELU:  $\max(0, x)$ , Tanh, Leaky RELU, etc. The volume remains unchanged hence output volume will have dimensions 32 x 32 x 12.
- **Pooling layer:** This layer is periodically inserted in the covnets and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents overfitting. Two common types of pooling layers are max pooling and average pooling. If we use a max pool with  $2 \times 2$  filters and stride 2, the resultant volume will be of dimension 16x16x12.



**Figure 3.6:** Max pool with 2x2 filters and stride 2[38].

- **Flattening:** The resulting feature maps are flattened into a one-dimensional vector after the convolution and pooling layers so they can be passed into a completely linked layer for categorization or regression.
- **Fully Connected Layers:** It takes the input from the previous layer and computes the final classification or regression task.
- **Output Layer:** The output from the fully connected layers is then fed into a logistic function for classification tasks like sigmoid or softmax which converts the output of each class into the probability score of each class.

### 3.9 Separable Convolution

A Separable Convolution is a process in which a single convolution can be divided into two or more convolutions to produce the same output. A single process is divided into two or more sub-processes to achieve the same effect.

Mainly there are two types of Separable Convolutions:

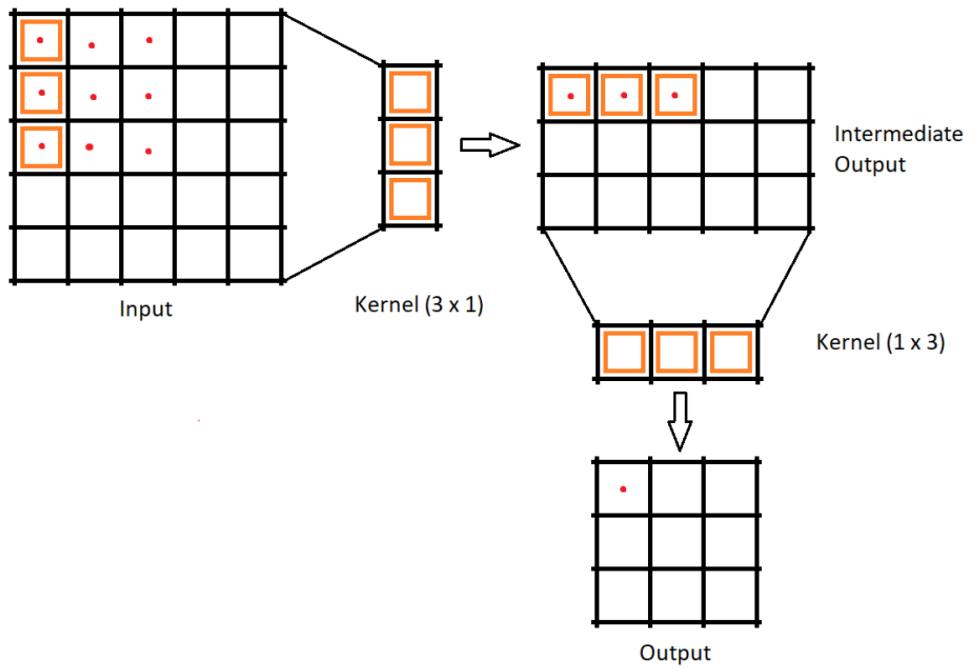
- Spatially Separable Convolutions.
- Depth-wise Separable Convolutions.

### 3.9.1 Spatially Separable Convolutions

In images height and width are called spatial axes. The kernel that can be separated across spatial axes is called the spatially separable kernel. The kernel is broken into two smaller kernels and those kernels are multiplied sequentially with the input image to get the same effect of the full kernel. For example,

$$\begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -1 \end{pmatrix} \quad (3.5)$$

In the conventional convolutions, if the kernel is  $3 \times 3$  then the number of parameters would be 9. In spatially separable convolution we divide the kernel into two kernels of shapes  $3 \times 1$  and  $1 \times 3$ . The input is first convolved with  $3 \times 1$  kernel and then with  $1 \times 3$ , then the number of parameters would be  $3 + 3 = 6$ . So less matrix multiplication is required. An important



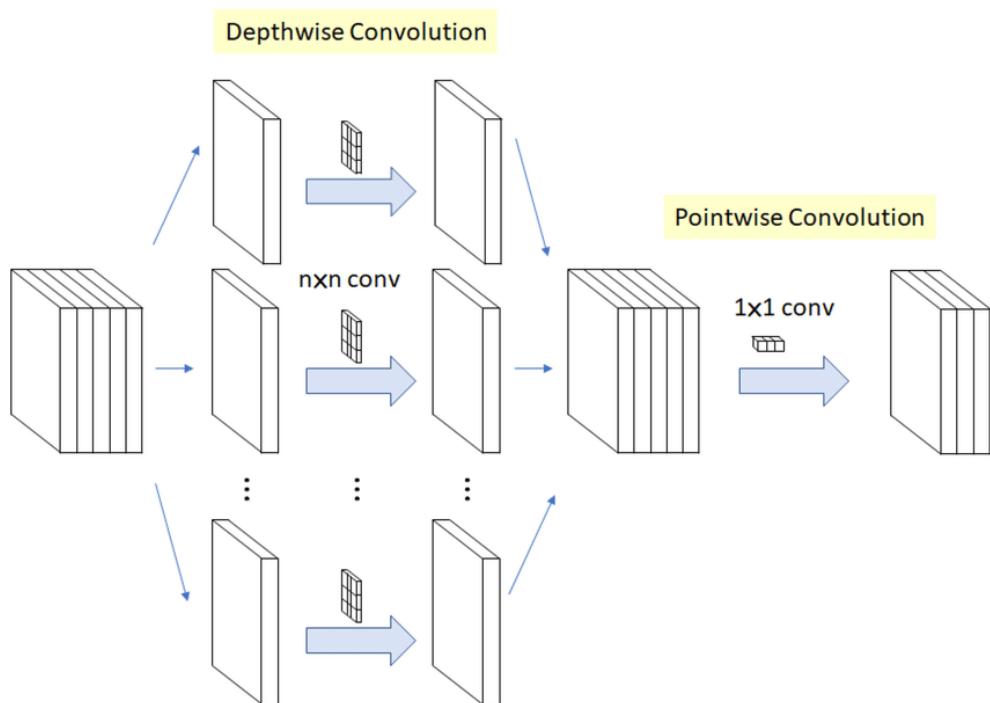
**Figure 3.7:** Convolutions with spatially separable kernels[39].

thing to note here is that not every kernel can be separated. Because of this drawback, this method is used lesser compared to Depthwise separable convolutions.

### 3.9.2 Depthwise Separable Convolutions

When we call `tf.keras.layers.SeparableConv2D` we would be calling a Depthwise separable convolution layer itself. Here you can use even those kernels which can not be spatially separable. Similar to spatial convolution, here also a regular convolution is divided into two convolutions namely

- Depthwise convolution
- Pointwise convolution



**Figure 3.8:** Depthwise Separable Convolutions [40].

#### 3.9.2.1 Depthwise convolution

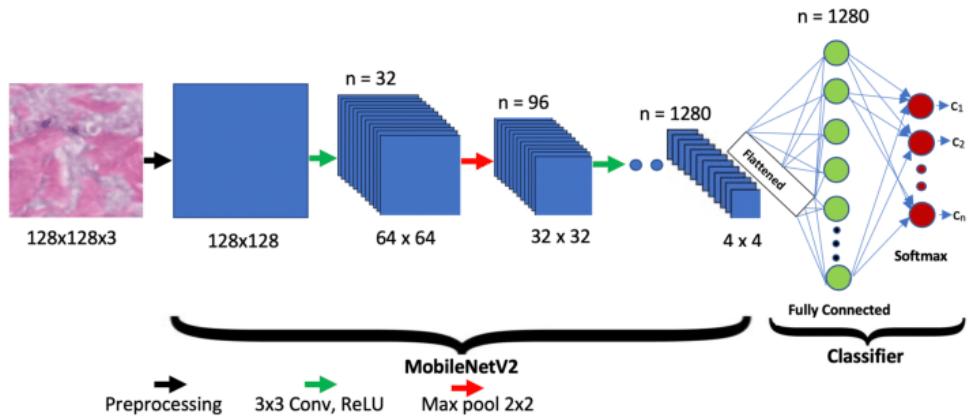
Let us assume we have an image input of shape  $7 \times 7 \times 3$ . We make sure after the depthwise convolution the intermediate image has the same depth. This is done by convoluting with 3 kernels with shape  $3 \times 3 \times 1$ . Each kernel iterates on only one channel of the image-producing an intermediate output of shape  $5 \times 5 \times 1$  which are stacked together to create an output of shape  $5 \times 5 \times 3$ .

### 3.9.2.2 Pointwise convolution

After the depthwise convolution, we have an intermediate output of shape  $5 \times 5 \times 3$ . Now we need to increase the depth of the output, which is done by convoluting with a kernel with shape  $1 \times 1 \times \text{depth}$  which is called a pointwise convolution. Let us assume the depth is 32. Then after pointwise convolution, the output would have the shape  $5 \times 5 \times 32$ . Which is equivalent to convolution with 32 filters of shape  $5 \times 5$ . In this example, the regular convolution would have to do 32  $3 \times 3 \times 3$  kernels that move  $5 \times 5$  times that is 21600 multiplication but in this method, we would have to do  $3 \times 3 \times 3 \times 5 \times 5 + 32 \times 1 \times 1 \times 3 \times 5 \times 5 = 3075$  multiplication. So we can see that this method can eliminate a large chunk of multiplication.

## 3.10 MobileNetV2 Model

In this section, MobileNetV2, by Google, is briefly reviewed. In MobileNetV1, Depthwise Separable Convolution is introduced which dramatically reduce the complexity cost and model size of the network, which is suitable to Mobile devices, or any devices with low computational power. In MobileNetV2, a better module is introduced with inverted residual structure. Non-linearities in narrow layers are removed this time. With MobileNetV2 as backbone for feature extraction, state-of-the-art performances are also achieved for object detection and semantic segmentation.



**Figure 3.9:** MobileNetV2 [41].

Key Features:

- Depthwise Separable Convolutions

- Linear Bottleneck
- Inverted residuals

### 3.10.1 Depthwise Separable Convolutions

Depthwise Separable Convolutions are a key building block for many efficient neural network architectures [42, 43] and we use them in the present work as well. The basic idea is to replace a full convolutional operator with a factorized version that splits convolution into two separate layers. The first layer is called a depthwise convolution, it performs lightweight filtering by applying a single convolutional filter per input channel. The second layer is a  $1 \times 1$  convolution, called a pointwise convolution, which is responsible for building new features through computing linear combinations of the input channels.

Standard convolution takes an  $h_i \times w_i \times d_i$  input tensor  $L_i$ , and applies convolutional kernel  $K \in R^{k \cdot k \cdot d_i \cdot d_j}$  to produce an  $h_i \times w_i \times d_j$  output tensor  $L_j$ . Standard convolutional layers have the computational cost of  $h_i \cdot w_i \cdot d_i \cdot d_j \cdot k \cdot k$ .

Depthwise separable convolutions are a drop-in replacement for standard convolutional layers. Empirically they work almost as well as regular convolutions but only cost:

$$h_i w_i d_i (k^2 + d_j) \quad (3.6)$$

which is the sum of the depthwise and  $1 \times 1$  pointwise convolutions. Effectively depthwise separable convolution reduces computation compared to traditional layers by almost a factor of  $k^2$ . MobileNetV2 uses  $k = 3$  ( $3 \times 3$  depthwise separable convolutions) so the computational cost is 8 to 9 times smaller than that of standard convolutions at only a small reduction in accuracy[42].

### 3.10.2 Linear Bottleneck

Consider a deep neural network consisting of  $n$  layers  $L_i$  each of which has an activation tensor of dimensions  $h_i \times w_i \times d_i$ . Throughout this section we will be discussing the basic properties of these activation tensors, which we will treat as containers of  $h_i \times w_i$  “pixels” with  $d_i$  dimensions. Informally, for an input set of real images, we say that the set of layer activations (for any layer  $L_i$ ) forms a “manifold of interest”. It has been long assumed that

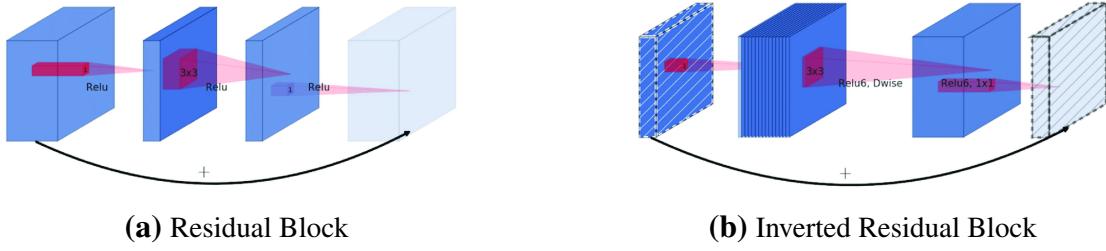
manifolds of interest in neural networks could be embedded in low-dimensional subspaces. In other words, when we look at all individual d-channel pixels of a deep convolutional layer, the information encoded in those values actually lie in some manifold, which in turn is embeddable into a low-dimensional subspace.

At a first glance, such a fact could then be captured and exploited by simply reducing the dimensionality of a layer thus reducing the dimensionality of the operating space. This has been successfully exploited by MobileNetV1 [42] to effectively trade off between computation and accuracy via a width multiplier parameter, and has been incorporated into efficient model designs of other networks as well [44]. Following that intuition, the width multiplier approach allows one to reduce the dimensionality of the activation space until the manifold of interest spans this entire space. However, this intuition breaks down when we recall that deep convolutional neural networks actually have non-linear per coordinate transformations, such as ReLU. For example, ReLU applied to a line in 1D space produces a 'ray', where as in  $R^n$  space, it generally results in a piece-wise linear curve with n-joints.

It is easy to see that in general if a result of a layer transformation  $\text{ReLU}(Bx)$  has a non-zero volume  $S$ , the points mapped to interior  $S$  are obtained via a linear transformation  $B$  of the input, thus indicating that the part of the input space corresponding to the full dimensional output, is limited to a linear transformation. In other words, deep networks only have the power of a linear classifier on the non-zero volume part of the output domain. We refer to supplemental material for a more formal statement.

On the other hand, when ReLU collapses the channel, it inevitably loses information in that channel. However if we have lots of channels, and there is a structure in the activation manifold that information might still be preserved in the other channels. In supplemental materials, we show that if the input manifold can be embedded into a significantly lower-dimensional subspace of the activation space then the ReLU transformation preserves the information while introducing the needed complexity into the set of expressible functions.

To summarize, we have highlighted two properties that are indicative of the requirement that the manifold of interest should lie in a low-dimensional subspace of the higher-dimensional activation space:



**Figure 3.10:** The difference between residual block and inverted residual [21]. Diagonally hatched layers do not use non-linearities. We use thickness of each block to indicate its relative number of channels. Note how classical residuals connects the layers with high number of channels, whereas the inverted residuals connect the bottlenecks. Best viewed in color.

- If the manifold of interest remains non-zero volume after ReLU transformation, it corresponds to a linear transformation.
- ReLU is capable of preserving complete information about the input manifold, but only if the input manifold lies in a low-dimensional subspace of the input space.

These two insights provide us with an empirical hint for optimizing existing neural architectures: assuming the manifold of interest is low-dimensional we can capture this by inserting linear bottleneck layers into the convolutional blocks. Experimental evidence suggests that using linear layers is crucial as it prevents non-linearities from destroying too much information.

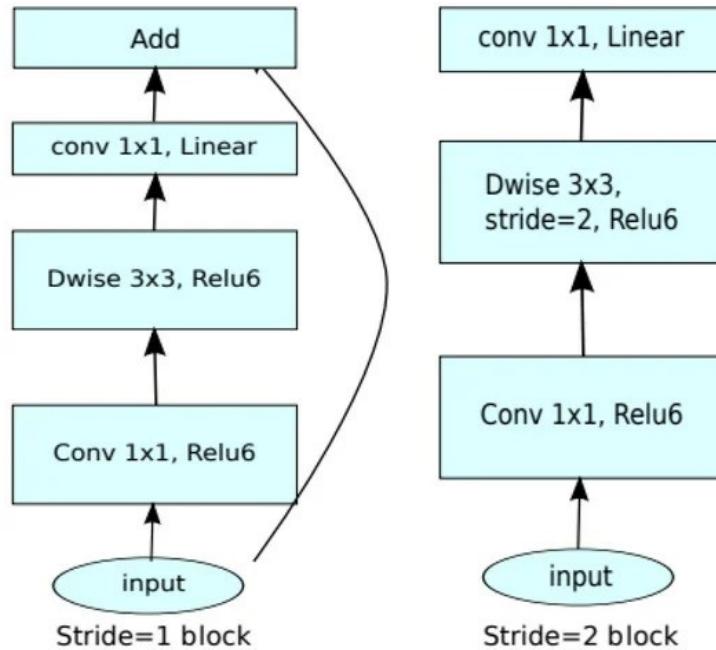
### 3.10.3 Inverted residuals

The bottleneck blocks appear similar to residual block where each block contains an input followed by several bottlenecks then followed by expansion [21]. However, inspired by the intuition that the bottlenecks actually contain all the necessary information, while an expansion layer acts merely as an implementation detail that accompanies a non-linear transformation of the tensor, we use shortcuts directly between the bottlenecks.

Figure 3.11 provides a schematic visualization of the difference in the designs. The motivation for inserting shortcuts is similar to that of classical residual connections: we want to improve the ability of a gradient to propagate across multiplier layers. However, the inverted design is considerably more memory efficient as well as works slightly better in our

experiments.

### 3.10.4 MobileNetV2 convolutional Blocks



**Figure 3.11:** MobileNetV2 convolutional Blocks[41].

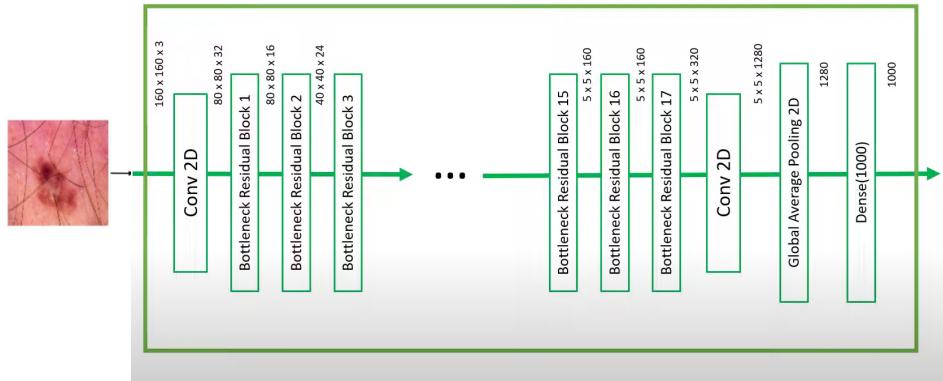
In MobileNetV2,

- There are two types of blocks. One is residual block with stride of 1. Another one is block with stride of 2 for downsizing.
- There are 3 layers for both types of blocks.
- This time, the first layer is  $1 \times 1$  convolution with ReLU6.
- The second layer is the depthwise convolution.
- The third layer is another  $1 \times 1$  convolution but without any non-linearity. It is claimed that if ReLU is used again, the deep networks only have the power of a linear classifier on the non-zero volume part of the output domain.
- And there is an expansion factor  $t$ . And  $t=6$  for all main experiments.
- If the input got 64 channels, the internal output would get  $64 \times t = 64 \times 6 = 384$  channels.

**Table 3.1:** Bottleneck residual block transforming from  $k$  to  $k'$  channels, with stride  $s$ , and expansion factor  $t$  [41]

Input	Operator	Dataset(Output)
$h \times w \times k$	$1 \times 1$ conv2d, ReLU6	$h \times w \times (tk)$
$h \times w \times (tk)$	$3 \times 3$ dwise, ReLU 6	$(h/s) \times (w/s) \times (tk)$
$(h/s) \times (w/s) \times (tk)$	linear $1 \times 1$ conv2d	$(h/s) \times (w/s) \times K'$

### 3.10.5 Architecture



**Figure 3.12:** MobileNetV2 Architecture[41].

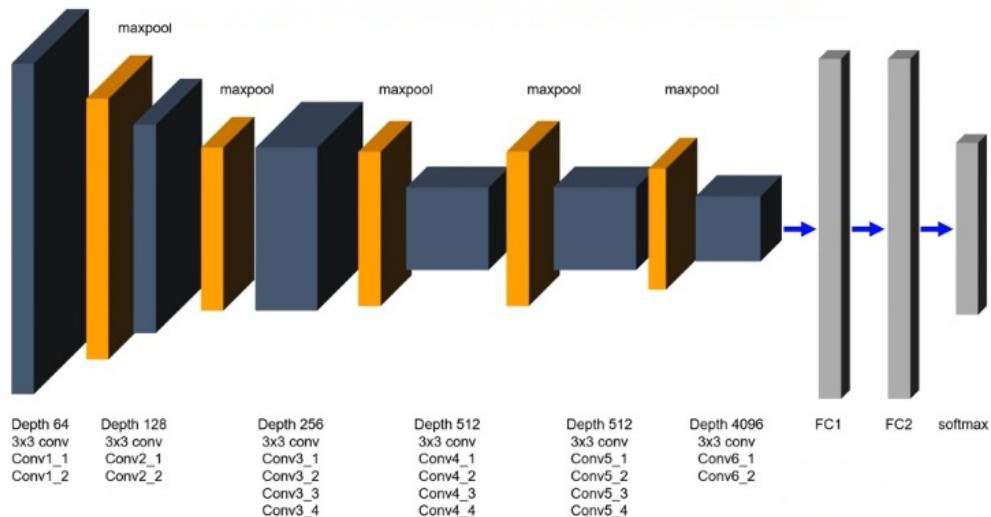
The basic building block is a bottleneck depth-separable convolution with residuals. The detailed structure of this block is shown in figure 3.12. The architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers. We use ReLU6 as the non-linearity because of its robustness when used with low-precision computation. We always use kernel size  $3 \times 3$  as is standard for modern networks, and utilize dropout and batch normalization during training.

With the exception of the first layer, we use constant expansion rate throughout the network. In our experiments we find that expansion rates between 5 and 10 result in nearly identical performance curves, with smaller networks being better off with slightly smaller expansion rates and larger networks having slightly better performance with larger expansion rates.

For all our main experiments we use expansion factor of 6 applied to the size of the input tensor. For example, for a bottleneck layer that takes 64-channel input tensor and produces a tensor with 128 channels, the intermediate expansion layer is then  $64 \cdot 6 = 384$  channels.

The use of MobileNetV2 for image classification offers several advantages. Firstly, its lightweight architecture allows for efficient deployment on mobile and embedded devices with limited computational resources. Secondly, MobileNetV2 achieves competitive accuracy compared to larger and more computationally expensive models. Lastly, the model's small size enables faster inference times, making it suitable for real-time applications.

### 3.11 VGG19 Model



**Figure 3.13:** VGG19 Neural Network Architecture.[45].VGG-19 has 16 convolution layers grouped into 5 blocks. After every block, there is a Maxpool layer that decreases the size of the input image by 2 and increases the number of filters of the convolution layer also by 2. The dimensions of the last three dense layers in block 6 are 4096, 4096, and 1000 respectively.

The precise structure of the VGG-16 network shown in figure 3.13 is described as follows:

- The first and second convolutional layers are comprised of 64 feature kernel filters and size of the filter is  $3 \times 3$ . As input image (RGB image with depth 3) passed into first and second convolutional layer, dimensions changes to  $224 \times 224 \times 64$ . Then the resulting output is passed to max pooling layer with a stride of 2.

- The third and fourth convolutional layers are of 124 feature kernel filters and size of filter is  $3 \times 3$ . These two layers are followed by a max pooling layer with stride 2 and the resulting output will be reduced to  $56 \times 56 \times 128$ .
- The fifth, sixth and seventh layers are convolutional layers with kernel size  $3 \times 3$ . All three use 256 feature maps. These layers are followed by a max pooling layer with stride 2.
- Eighth to thirteen are two sets of convolutional layers with kernel size  $3 \times 3$ . All these sets of convolutional layers have 512 kernel filters. These layers are followed by max pooling layer with stride of 1.
- Fourteen and fifteen layers are fully connected hidden layers of 4096 units followed by a softmax output layer (Sixteenth layer) of 1000 units.

The equation for the VGG-19 model can be represented as follows:

$$y = f(W, x) \quad (3.7)$$

Where  $y$  is the predicted class label,  $f$  is the function implemented by the VGG-16 model,  $W$  is the set of parameters (weights and biases) and  $x$  is the input image. The function  $f$  can be decomposed into the following sequence of operations:

### 3.11.1 Convolutional layers

In the convolutional layer, a matrix named kernel is passed over the input matrix to create a feature map for the next layer. Convolution is a specialized kind of linear operation which is widely used in a variety of domains including image processing, statistics, and physics. The convoluted image is calculated as follows in  $i$ -th convolutional layer with an image input,  $I$ , and a kernel filter,  $K$ :

$$Conv_i = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (3.8)$$

The output of the  $i$ -th convolutional layer  $h_i$  is expressed as :

$$h_i = Conv_i(h_{i-1}) + b_i \quad (3.9)$$

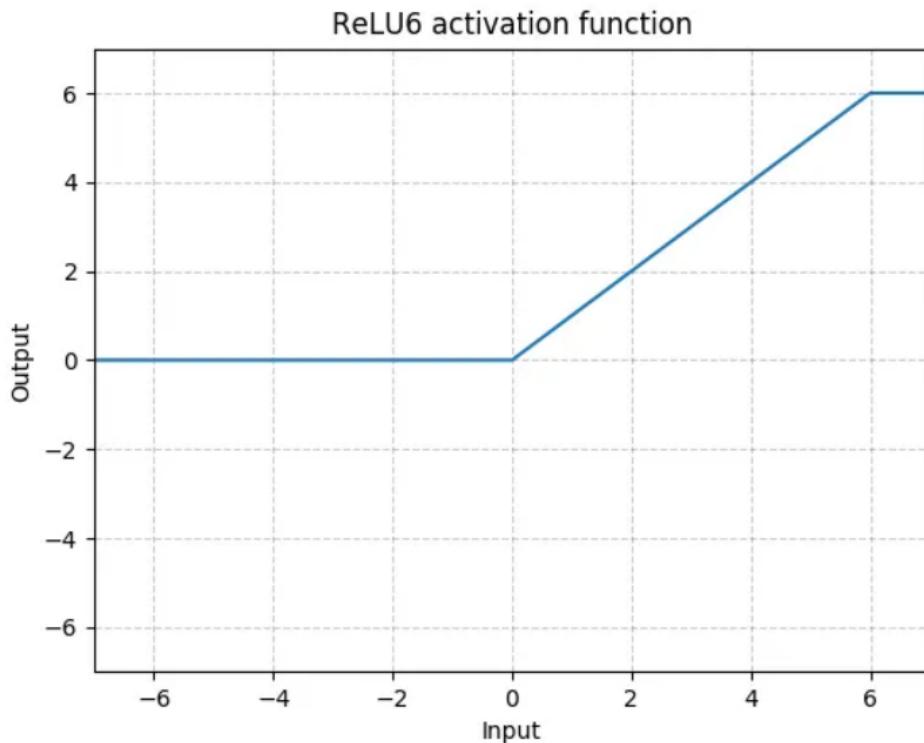
where  $\text{Conv}_i$  is the convolution operation, and  $b_i$  is the bias term.

### 3.11.2 Activation layers

An activation function is a node that comes after the convolutional layer and the activation function is the nonlinear transformation that we do over the input signal. The rectified linear unit activation function (ReLU) is a piecewise linear function that will output the input if it is positive, otherwise, it will output zero. The output of the i-th Activation layer  $h_i$  is:

$$h_i = \text{ReLU}(h_i) \quad (3.10)$$

where ReLU is the rectified linear activation function.



**Figure 3.14:** ReLu6 Activation Function.

### 3.11.3 Pooling layers

The drawback of the feature map output of the convolutional layer is that it records the precise position of features in the input. This means during cropping, rotation or any other minor changes to the input image will completely result in a different feature

map. To counter this problem, we approach the downsampling of convolutional layers. Downsampling can be achieved by applying a pooling layer after the nonlinearity layer. Pooling helps to make the representation become approximately invariant to small translations of the input. Invariance to translation means that if we translate the input by a small amount, the values of most of the pooled outputs do not change. The output of the i-th Pooling layer  $h_i$  is:

$$h_i = \text{Pool}_i(h_i) \quad (3.11)$$

where  $\text{Pool}_i$  is the pooling operation (e.g. max pooling).

### 3.11.4 Fully connected layers

At the end of a convolutional neural network, the output of the last Pooling Layer acts as input to the Fully Connected Layer. There can be one or more of these layers. Fully connected means that every node in the first layer is connected to every node in the second layer. The output of the i-th Fully connected layer  $h_i$  is:

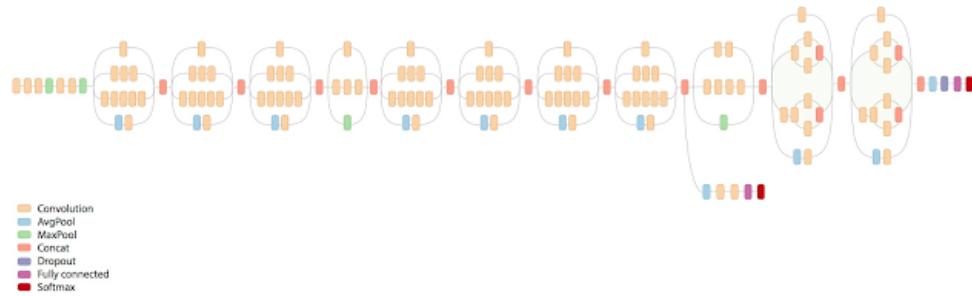
$$h_i = \text{FC}_i(h_{i-1}) + b_i \quad (3.12)$$

where  $\text{FC}_i$  is the fully connected operation, and  $b_i$  is the bias term.

This equation represents the forward pass of the VGG-19 model, which takes an input image  $x$  and outputs the predicted class label  $y$ . During training, the parameters  $W$  are optimized using a loss function, such as cross-entropy, to minimize the difference between the predicted and ground-truth labels.

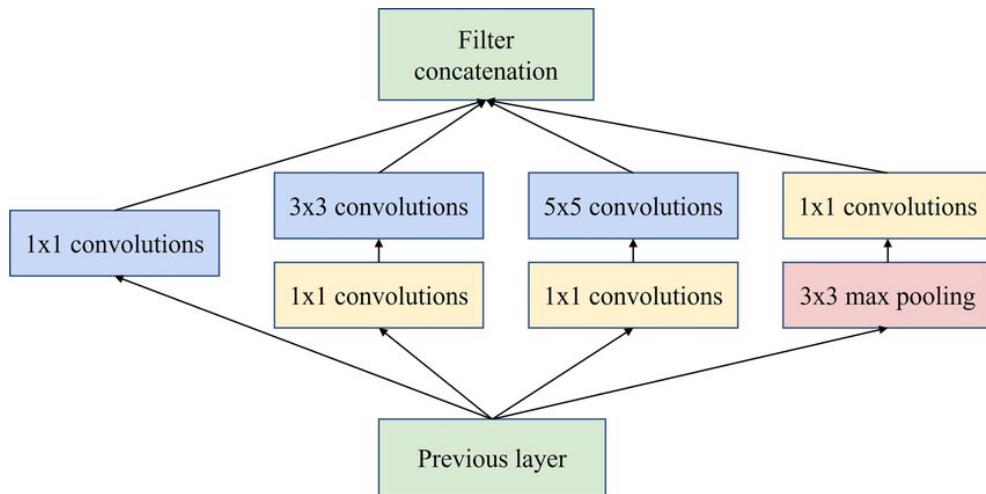
## 3.12 InceptionV3 Model

Inception V3 was the top performers on ImageNet with 0.937 accuracy for top-5 and 0.779 for top-1. The namesake of Inception v3 is the Inception modules it uses, which are basically mini models inside the bigger model. The inspiration comes from the idea that you need to make a decision as to what type of convolution you want to make at each layer:  $3 \times 3$ ? Or a  $5 \times 5$ ? The idea is that one don't need to know ahead of time if it was better to do, for example, a  $3 \times 3$  then a  $5 \times 5$ . Instead, just do all the



**Figure 3.15:** InceptionV3 architecture with 11 inception blocks.

convolutions and let the model pick what's best. Additionally, this architecture allows the model to recover both local feature via smaller convolutions and high abstracted features with larger convolutions. The larger convolutions are more computationally expensive, so it suggests first doing a  $1 \times 1$  convolution reducing the dimensionality of its feature map, passing the resulting feature map through a ReLU, and then doing the larger convolution (in this case,  $5 \times 5$  or  $3 \times 3$ ). The  $1 \times 1$  convolution is key because it will be used to reduce the dimensionality of its feature map.



**Figure 3.16:** Inception module with dimensionality reduction.

### 3.13 Technical Issue

Given InceptionV3 trained on ImageNet with 11 inception blocks or VGG19, 2 kinds of experiment can be performed:

- Fine-tuning Inception V3 from the last 2 inception blocks.

- Fine-tune the whole pretrained model.

The reason for this is the implementation of Batch Norm in Keras. The way Keras implemented Batch Norm is as follow. During training the network will always use the mini-batch statistics either the BN layer is frozen or not; also during inference it will use the previously learned statistics of the frozen BN layers. As a result, if we fine-tune the top layers, their weights will be adjusted to the mean/variance of the new dataset. Nevertheless, during inference they will receive data which are scaled differently because the mean/variance of the original ImageNet dataset will be used. Consequently, if I followed exactly what were did when fine-tuning VGG19 with InceptionV3, we will have very bad validation accuracy. One temporary solution to this issue is to set all Batch Normalization layer to trainable, so during inference, batch norm layers will use statistics of the mini-batch from our training set.

## Chapter IV

### Methodology

#### 4.1 Introduction

In this chapter, the proposed methodology is explained. .

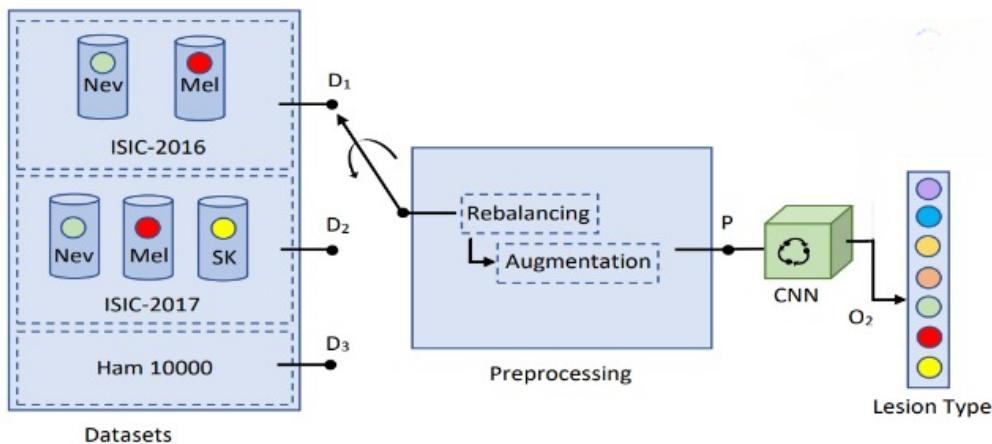
Artificial intelligence (AI) revolution, brings new promises to clinical practice using medical images[33]. Deep learning (DL) attempts to model high-level abstractions in data using multiple processing layers[39] . Recently, DL models have been achieving remarkable results in skin cancer classification tasks. In particular, convolutional neural networks (CNNs) have become the standard approach to handling computer vision problems. Compared with the traditional machine learning algorithms, which require complex feature engineering, DL can automatically learn robust feature representation and adapt to different fields and applications more easily. In this research, I used a DL algorithm in an attempt to develop an automated classification system using dermoscopic images of different established skin disorders.

#### 4.2 Proposed method

The overall architecture of our proposed system is depicted in Fig. 4.1. We utilize three different datasets (either D1 or D2 or D3), where D1 is a binary class Dataset and D2 or D3 is a multi-class Dataset. An input, either D1 or D2, or D3, generates output classification (O). The outputs O is then processed to provide lesion detection and recognition results. However, different preprocessing and convolutional Neural Networks of our proposed system are explained in the following sections.

##### 4.2.1 Image classification

Image classification is the task of assigning a label or class to an entire image. Images are expected to belong to only one class each. Image classification models take an image as input and return a prediction about the class to which the image belongs.



**Figure 4.1:** Overview of proposed method for skin lesions classification.

Classifying skin disease images with deep learning has the following steps:

- Examine and understand data
- Build an input pipeline
- Build the model
- Train the model
- Test the model
- Improve the model and repeat the process

Choosing a good backbone model is an extremely important part of achieving good experimental results.

#### 4.2.2 Data Pre-processing

The first step in the computerized analysis of skin lesion images is the pre-processing of an image. The pre-processing techniques will be different for different application based on the desired dataset of an image. The main aim of pre-processing techniques is image enhancement and image restoration.

Class rebalancing and different image augmentations is applied as a preprocessing, which are concisely explained as follows:

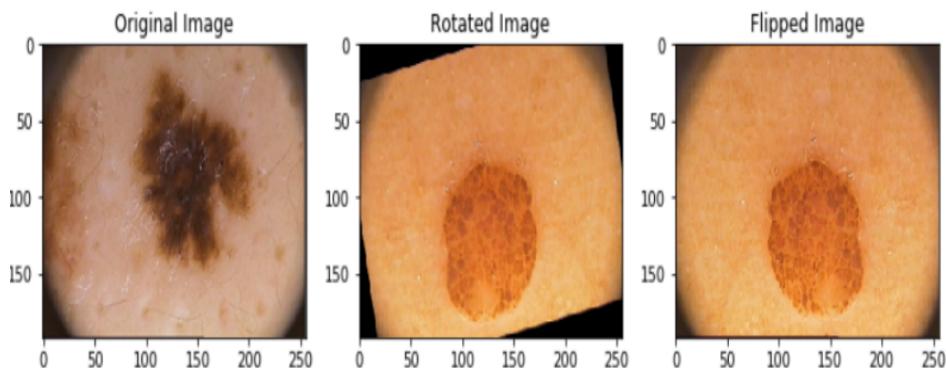
#### 4.2.2.1 Rebalancing

Class imbalance is a common phenomenon in the medical imaging domain as manually annotated images are very complex and arduous to achieve. Such a class imbalance can be partially overcome using two commonly used approaches, such as the data-level method and the algorithmic-level method[46]. We have combined additional images to the underrepresented class from the ISIC archive and weighted the loss function. For weighing the loss function, we apply  $W_i = N_i / N$ , where  $W_i$ ,  $N$ , and  $N_i$  are the weight for ith class, the total sample numbers, and the sample numbers in the ith class, respectively

#### 4.2.2.2 Data augmentation

Data augmentation is a manipulation applied to images to create different versions of similar content in order to expose the model to a wider range of training examples. For example, randomly altering the rotation, brightness, or scale of an input image requires that a model consider what an image subject looks like in a variety of situations.

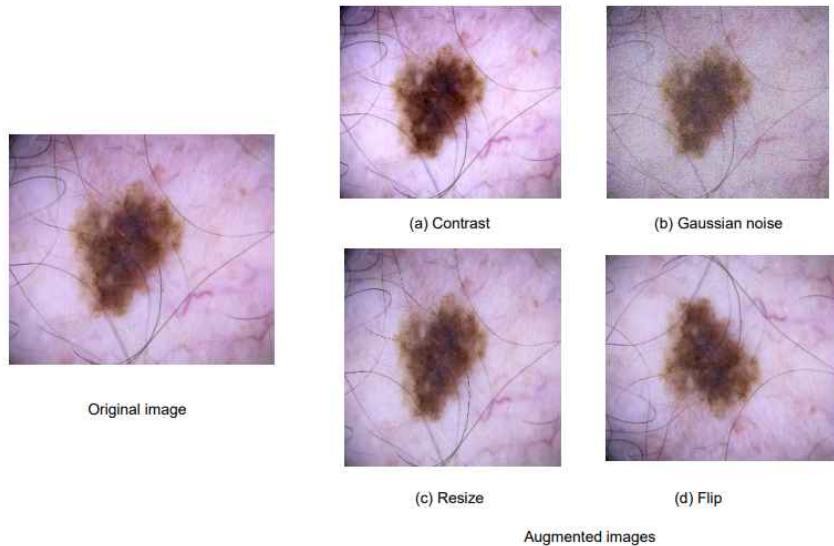
Data augmentation manipulations are forms of image preprocessing, but there is a critical difference: while image preprocessing steps are applied to both training and test sets, image augmentation is only applied to the training dataset. Thus, a transformation that could be an augmentation manipulation in some situations may best be a preprocessing step in others.



**Figure 4.2:** Applied different augmentations based on geometric transformations, such as rotation, and flipping.

One of the crucial challenges in the medical imaging domain is coping with small datasets, such as the ISIC-16 datasets. However, we have applied different augmentations based on geometric transformations, such as rotation, and flipping shown in Fig 4.2.

However, the input D1 or D2, or D3 produces the lesion classification output by applying preprocessing type: the P: rebalancing and augmentation.



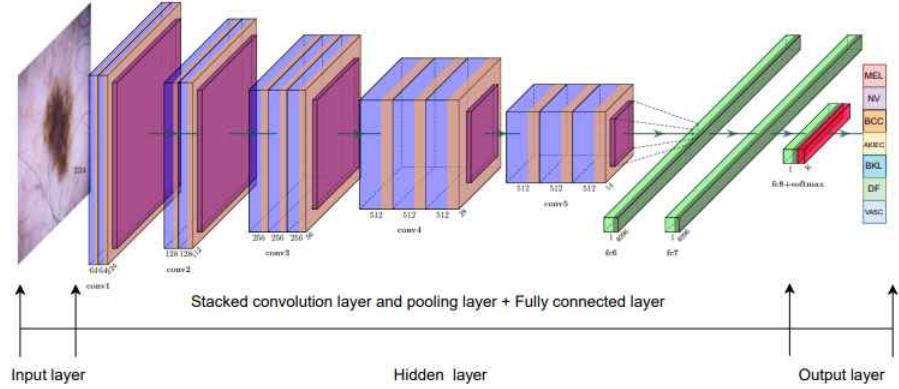
**Figure 4.3:** The difference between the original skin lesion image (left) and the images after data augmentation (right).

It is impossible to truly capture an image that accounts for every real world scenario a model may encompass. This is where augmentation can help. By augmenting the images, I can increase the sample size of the training data and add new cases that might be hard to find in the real world. Augmenting existing training data to generalize to other situations allows the model to learn from a wider range of situations. This is particularly important when collected datasets may be small. A deep learning model will (over)fit to the examples shown in training, so creating variation in the input images enables the generation of new, useful training examples.

#### 4.2.3 Convolutional Neural Network

CNNs have the advantage of non-linear mapping by automatically adjusting the training weights between neurons. The standard CNN is basically composed of three layers: the convolutional, pooling, and fully connected layers.

The convolutional layer is the most important layer and performs most of the computation. It consists of kernels, which are composed of weights that learn visual features from the input images. Each kernel is convolved across the whole image and produces a feature map, which is the output of this layer. The pooling layer is basically used to reduce the feature map size. Consequently, it reduces the number of training parameters for the next layers, helps control overfitting, and, along with a non-linear activation filter, incorporates non-linearity into the network. The fully connected layer is a standard neural network that is connected to the last feature map provided by the previous layer. In summary, the composition of the convolutional and pooling layers is known as the feature extractor, and the fully-connected layer is the classifier.



**Figure 4.4:** Illustration of the backbone network (Vgg-19) used to classify skin cancer images. First, the image features are extracted by the CNN feature extractor. Next, these features are reduced by the pooling layer. Finally, the softmax layer serves as the output layer of the model to predict skin lesion classes.

### 4.3 Proposed Classification Network

The task is to classify the detected region. In the machine learning paradigm, Convolutional Neural Networks are complex feed-forward neural networks. CNNs are used for image classification and recognition because of their high accuracy. CNN-based classification models are best for classifying medical images. There are several CNN-based model architectures for classification problems. Some of them which were evaluated using benchmark dataset IMAGENET[17] are listed in Table 4.1.

**Table 4.1:** Sumarry of various architectures

<b>Model Name</b>	<b>Number of parameters (Millions)</b>	<b>Accuracy (Percentage)</b>
Alexnet	60 M	63.3
Inception	5 M	69.8
VGG 19	138 M	74.4
ResNet-50	26 M	77.15
MobileNet	280 M	80.4

So we will ensemble VGG19,MobileNet and InceptionV3 for better classification output and to keep a total number of parameters as lower as possible for a lightweight model architecture. There are several benefits of using an ensemble of VGG19,MobileNet and InceptionV3 models:

- **Improved accuracy:** The ensemble model is able to capture the strengths of each individual model and, as a result, can lead to improved accuracy compared to using a single model.
- **Increased robustness:** The ensemble approach helps to reduce the impact of overfitting and increase the robustness of the model. This is because the ensemble model considers multiple perspectives of the same input data, which helps to reduce the impact of noise or outliers.
- **Increased generalization:** The ensemble approach can also lead to improved generalization performance, as the model can learn from the different representations learned by the individual models.
- **Reduced variance:** Ensemble models can also reduce the variance of the model predictions, which is the measure of the spread of the predictions. This is because the individual models have different ways of approaching the same task, which helps to reduce the variance in their predictions.

The ensemble of VGG19,MobileNet and InceptionV3 models is a technique in deep learning where multiple pre-trained models are combined to form a single, more robust, and accurate model. The idea behind this approach is that the individual models may have strengths in different areas and combining their predictions can lead to improved results. In this study, we first train multiple pre-trained models, such as

VGG19, MobileNet and InceptionV3, on a segmented lesion. After training, the models are combined in an ensemble by taking a weighted average of their predictions. The overall ensemble method is shown in algorithm 1.

---

**Algorithm 1** Ensemble of VGG19, MobileNetV2 and InceptionV3 models

---

**Input:** Training data  $X_{train}$ , target labels  $Y_{train}$ , validation data  $X_{val}$ , target labels  $Y_{val}$ , test data  $X_{test}$ , target labels  $Y_{test}$ .

**Output:** Ensemble accuracy (acc)

- 1: **procedure** Load the pre-trained VGG19, MobileNetV2 and InceptionV3 models as  $\text{vgg19} = \text{LoadVGG19Model}()$ ,  $\text{MobileNetV2} = \text{LoadMobileNetV2Model}()$  and  $\text{InceptionV3} = \text{LoadInceptionV3Model}()$
  - 2: Freeze the weights of the pre-trained models to prevent overfitting.
  - 3: Add a global average pooling layer to the outputs of the pre-trained models like  $w = \text{GlobalAveragePooling}(\text{MobileNetV2.output})$   
 $x = \text{GlobalAveragePooling}(\text{vgg19.output})$  and  
 $y = \text{GlobalAveragePooling}(\text{InceptionV3.output})$
  - 4: Concatenate the outputs of the two models to form  $z = \text{Concatenate}(w, x, y)$ .
  - 5: Add a dense layer for classification with a ReLU activation function.
  - 6: Combine the three models into a single model:  $\text{ensemble}_\text{model} = \text{CombineModels}(\text{MobileNetV2.input}, \text{vgg19.input}, \text{InceptionV3.input})$
  - 7: Compile the ensemble model using categorical cross-entropy as the loss function and Adam as the optimizer with a learning rate of 0.0001.
  - 8: Train the ensemble model using the training data  $X_{train}$  and target labels  $Y_{train}$ , with a batch size of 32 and 10 epochs. Validate the model using the validation data  $X_{val}$  and target labels  $Y_{val}$ .
  - 9: Make predictions using the ensemble model on the test data  $X_{test}$ .
  - 10: Average the predictions from the two models as  $\text{average}_\text{predictions} = \text{AveragePredictions}(\text{ensemble}_\text{predictions})$ .
  - 11: Calculate the accuracy of the averaged predictions using the target labels  $Y_{test}$ .
  - 12: **return** acc. ▷ the ensemble accuracy
- 

#### 4.4 Conclusion

The chapter discussed the proposed methodology for the classification network. The algorithm for ensemble network is demonstrated. The next chapter shows the implementation and result.

## Chapter V

### Implementation, Results and Discussions

#### 5.1 Introduction

The efficiency and performance of the offered methods are investigated in this chapter. Several tests are carried out to evaluate the scalability of the offered methodologies. This chapter includes a simulation analysis for a deeper comprehension of the proposed method's performance. Experimentation data sets and simulation results are covered in this chapter to analyze the efficacy of the proposed strategies.

#### 5.2 Experimental Setup

This section mentions the technologies installed and used for implementing models and visualizations. Technical programming is performed in ‘Python’ using its AI libraries.

##### 5.2.1 Python

Python is a high-level, interpreted programming language that was first released in 1991 by Guido van Rossum.[47] It is known for its readability and ease of use, making it a popular language for beginners and experienced programmers alike. Python is widely used in various fields, including web development, scientific computing, data analysis, artificial intelligence, and many others.

One of the key advantages of Python is its large and supportive community, which has developed numerous libraries and tools that extend the functionality of the language. These include libraries for data analysis (such as NumPy and Pandas) [48], machine learning (such as TensorFlow [49] and PyTorch [50]), and visualization (such as Matplotlib and Seaborn) [51]. This makes Python a versatile language that can be used for a wide range of tasks and applications.

Python is also highly extensible and can be easily integrated with other programming languages and tools. This makes it a popular choice for research and development projects, as well as for commercial applications. The language is maintained by the Python Software Foundation, which ensures that it is constantly being improved and updated to meet the needs of its users.

Python is an Object-Oriented Programming language with a high degree of abstraction. It is a general-purpose programming language with extensive support for machine learning and statistical models in its libraries. We utilized Python 3.8 for execution since the Python community and documentation is extensive. The models were trained using the Google Colab platform.

### **5.2.2 Google Colab**

Colab is a web-based Python editor that allows anybody to create and run arbitrary Python code. It's notably useful for machine learning, data analysis, and teaching. The specifications of the Google Colab platform are[52]:

- 1xTesla K80 (2496 CUDA cores)
- 1xsingle core hyperthreaded Xeon Processors @2.3Ghz
- 13 GB RAM
- 108 GB Run time HDD
- OS: Linux Kernel

### **5.2.3 Pandas**

Pandas, a Python library, is one of the most dependable libraries for dealing with enormous data sets. Its efficiency and ease of use have made it one of the most popular data analysis packages. Other libraries may exist, however, 'pandas' is relatively simple to use and work with [53].

### 5.2.4 NumPy

NumPy is a Python library that adds support for large, multi-dimensional arrays and matrices, as well as a large number of high-level mathematical functions to work on these arrays [54].

### 5.2.5 Matplotlib

Matplotlib is a graphing package for Python with NumPy, the Python numerical mathematics extension. We utilized it considerably for data visualization and analysis [55].

### 5.2.6 Keras and Tensorflow

TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML, and developers easily build and deploy ML-powered applications. Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. We used TensorFlow various models and evaluation metric functions during our study [56, 57]. Short experiments like validating the code's functionality were performed on a desktop computer. After choosing the platform to run the python notebook, we configured the environment with the necessary datasets and libraries.

## 5.3 Evaluation Metrics

We ran numerous tests with varying compositions using benchmark datasets we acquired from various sources to evaluate the classification accuracy and computational efficiency of convolutional neural networks. Various error measurements are utilized for evaluation, such as MAE, MSE, RMSE, and so on. True positive (TP) is properly classified as positive, whereas false negative (FN) is incorrectly classified (FN). True negative (TN) refers to an occurrence that is negative and accurately identified, whereas false-positive refers to the opposite (FP).

**Accuracy:** Accuracy is a performance metric in machine learning and computer vision that evaluates the overall accuracy of a model in making predictions. It measures the

		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP	FP
	Negative	FN	TN

**Figure 5.1:** Work plan using the Gantt Chart.

number of correct predictions made by the model, divided by the total number of predictions made. It can be expressed as a proportion or as a percentage.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.1)$$

where TP, TN, FP, and FN denote True Positive, True Negative, False Positive and False Negative respectively.

True Positives are the number of instances that the model correctly predicted as positive, and True Negatives are the number of instances that the model correctly predicted as negative.

Accuracy is a commonly used evaluation metric for both binary and multiclass classification problems. It provides a simple and straightforward way to measure the performance of a model, but it can be misleading in some cases, especially in imbalanced datasets where one class is more prevalent than the others. In such cases, precision, recall, and F1 Score are often used in conjunction with accuracy to get a more complete picture of a model's performance.

**Precision:** Precision is a performance metric in machine learning and computer vision that evaluates the fraction of true positive predictions among all positive predictions made by a model. It measures the ability of a model to avoid false positives, i.e. the number of instances where the model predicted the positive class while the ground truth label was negative.

$$Precision = \frac{TP}{TP + FP} \quad (5.2)$$

where TP and FP denote True Positive, False Positive respectively.

A high precision score indicates that the model is able to correctly identify positive instances with a low number of false positives, while a low precision score means that the model is making many false positive predictions. Precision is commonly used in binary classification tasks and is often used in conjunction with a recall to get a more complete picture of a model's performance.

**Sensitivity:** Sensitivity, also known as True Positive Rate (TPR), Recall or Hit Rate, is a performance metric in machine learning and computer vision that evaluates the fraction of actual positive instances that are correctly identified as positive by a model. It measures the ability of a model to detect positive instances, i.e. the number of instances where the model predicted the positive class and the ground truth label was also positive.

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (5.3)$$

where TP and FN denote True Positive, False Negative respectively.

A high sensitivity score indicates that the model is able to correctly identify a high proportion of positive instances, while a low sensitivity score means that the model is missing many positive instances. Sensitivity is commonly used in binary classification tasks and is often used in conjunction with precision to get a more complete picture of a model's performance.

### F-score

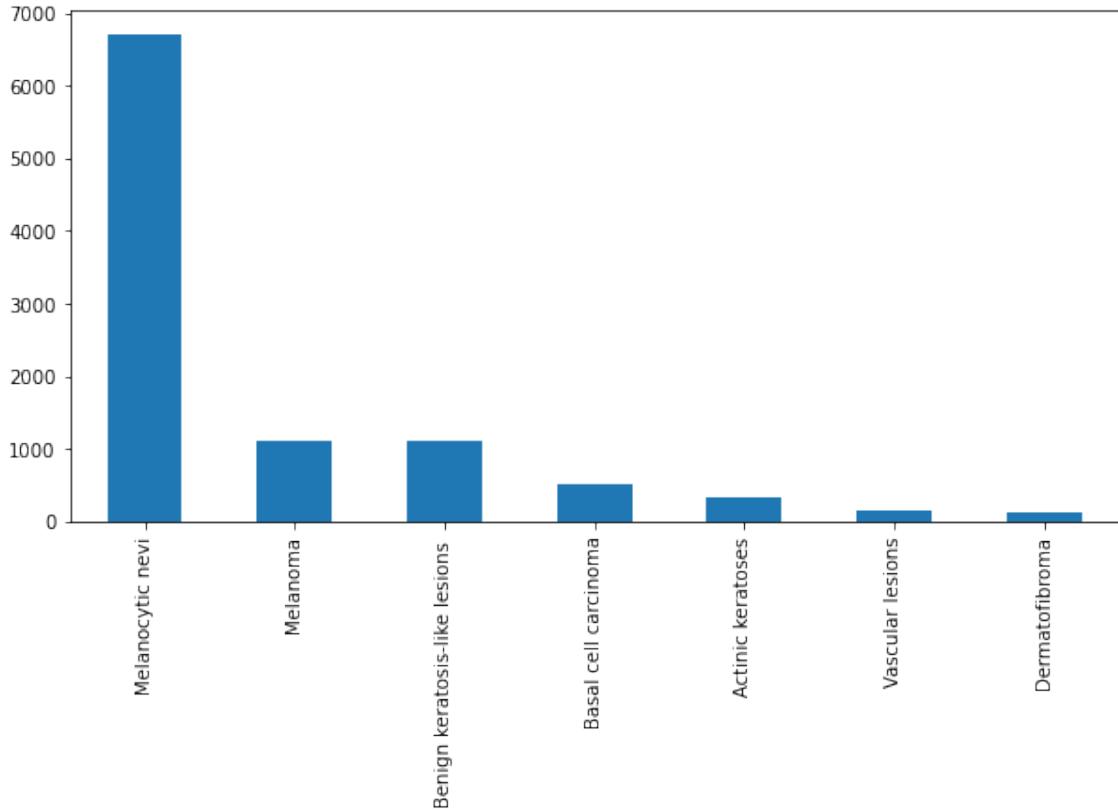
The F-score, also known as the F1-score, is a metric for how accurate a model is on a given dataset. The F-score, which is specified as the harmonic mean of the model's precision and recall, is a technique of combining the model's precision and recall. It's possible to tweak the F-score such that precision takes precedence over recall, or vice versa.

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5.4)$$

## 5.4 Datasets

Despite technological advances, however, the lack of valid clinical datasets has limited the application of deep learning research in medicine. I summarize twelve fairly popular dermoscopic datasets, that are commonly used for skin cancer classification. The most famous one is the International Skin Imaging Collaboration (ISIC) dataset, used in an annual skin cancer classification competition since 2016.

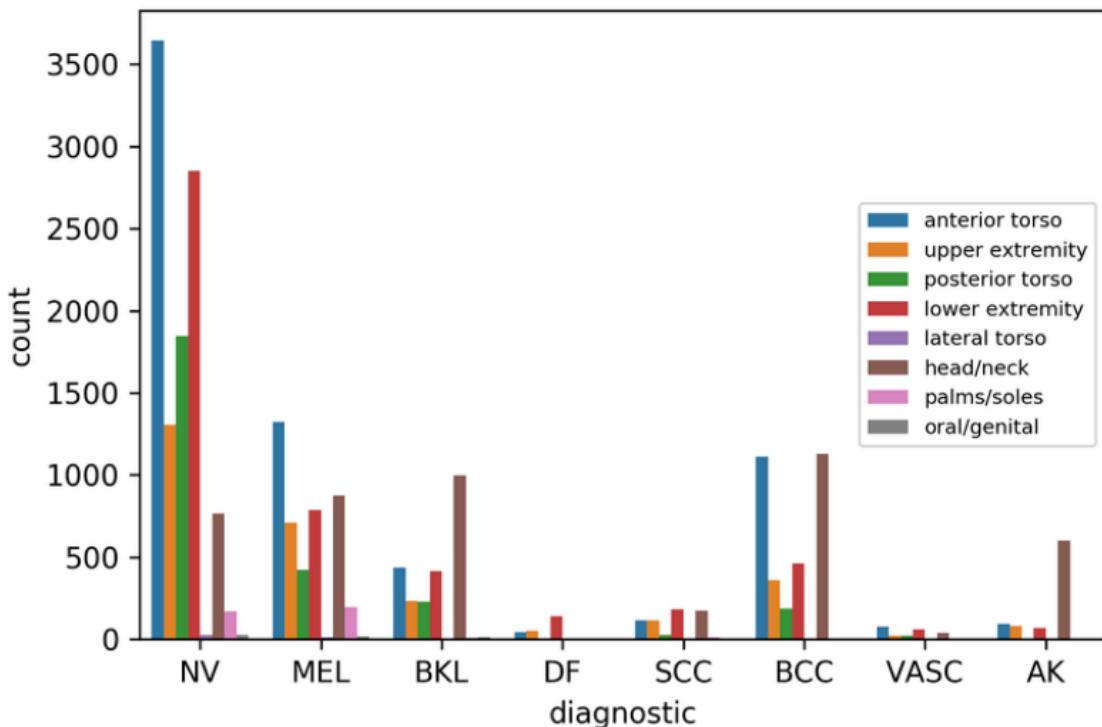
Three different datasets, such as HAM-10000[58],ISIC-2016 [59] and ISIC-2019 [60] are used to validate our proposed pipeline,whose class-wise distribution are presented in fig 5.2, ?? , 5.3 respectively.



**Figure 5.2:** Overview of HAM10000 Dataset.

**Table 5.1:** Summary of the datasets used in the experiments

Dataset	Lesion Types	Number of images	# of binary mask	resolution
HAM-10000[58]	7 Classes	10015	10015	128 × 128
ISIC-2016 [59]	2 Classes	900	900	128 × 128
ISIC-2019 [60]	8 Classes	25331	25331	128 × 128



**Figure 5.3:** Overview of ISIC 2019 Dataset.

The ISIC-2016 contains a binary class, aiming to classify as either Nevus (Nev) or Melanoma (Mel), explicating that class samples are imbalanced (4.2: 1 for Nev: Mel). On the other hand, the ISIC-2019 is a multi-class categorization task, intending to classify as either Nevus (Nev), Seborrheic Keratosis (SK), or Melanoma (Mel). The distribution of ISIC-2019 also tells that class samples are highly imbalanced, where Nev: SK: Mel in the Dataset is 5.4: 1.5: 1. The distribution of HAM10000 also tells that class samples are highly imbalanced shown in Fig 5.2. This unbalanced distribution situation consists of data belonging to 6705 nevi classes, which are considered harmless, whereas the remaining data Dermatofibroma (Df), Vascular skin (Vasc), Actinic keratosis(Akiec), Basal cell carcinoma (Bcc), Benign keratosis (Bkl), and Melanoma (Mel) consists of 3310. The training samples' imbalanced distribution makes the classifier biased toward the particular class with more samples, mitigated in the proposed pipeline by adopting two techniques. Table 4.1 demonstrates a concise summary of the datasets used in the experiment.

## 5.5 Implementation and Results

The evaluation of classification has been performed using qualitative and quantitative measures. The dataset splits into the 80% training portion and 20% images on the validation side.

### 5.5.1 Quantitative results

Firstly, the quantitative classification results is exhibited applying the ensemble method of the pre-trained MobileNetV2,VGG19 and InceptionV3 models.

#### 5.5.1.1 Training and validation graph

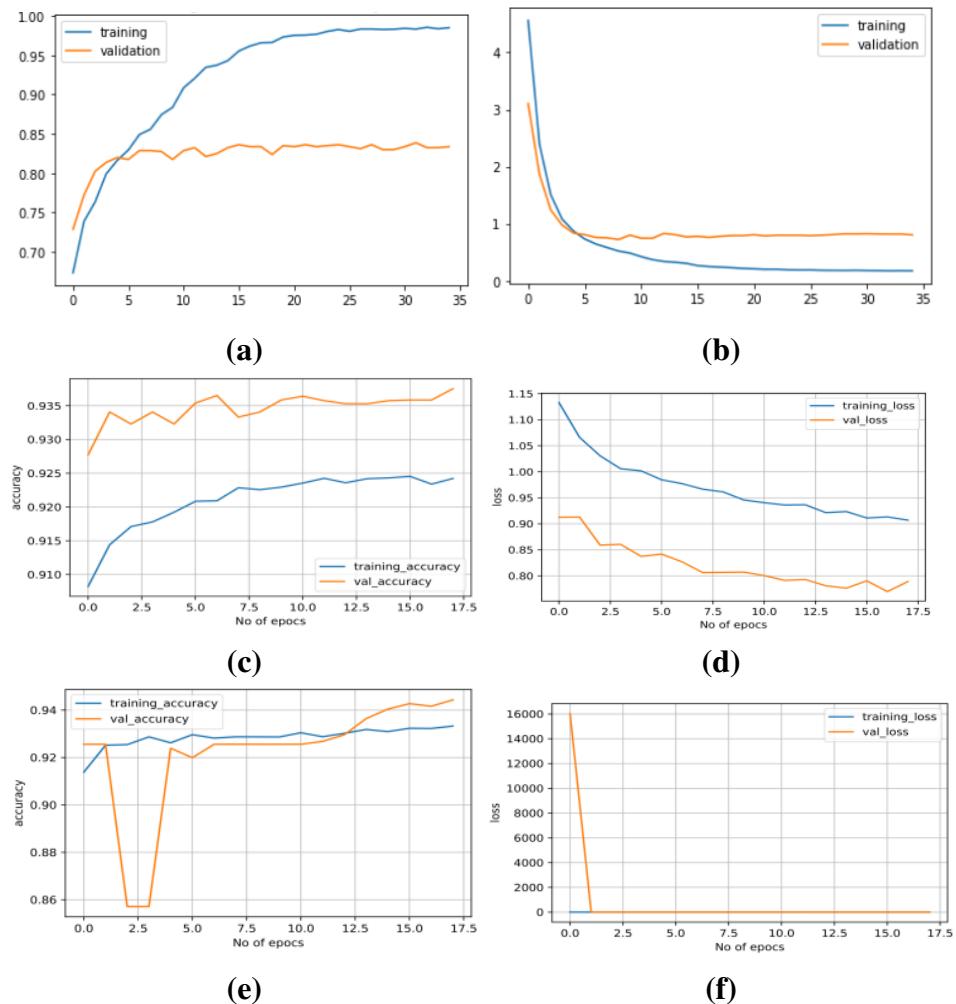
Training and validation graphs of MobileNetV2,VGG19 and InceptionV3 on HAM10000 [58],ISIC-16 [59] and ISIC-19 [60] have been shown in Figures 5.4, 5.5, 5.6 respectively.

#### 5.5.1.2 Confusion Matrix

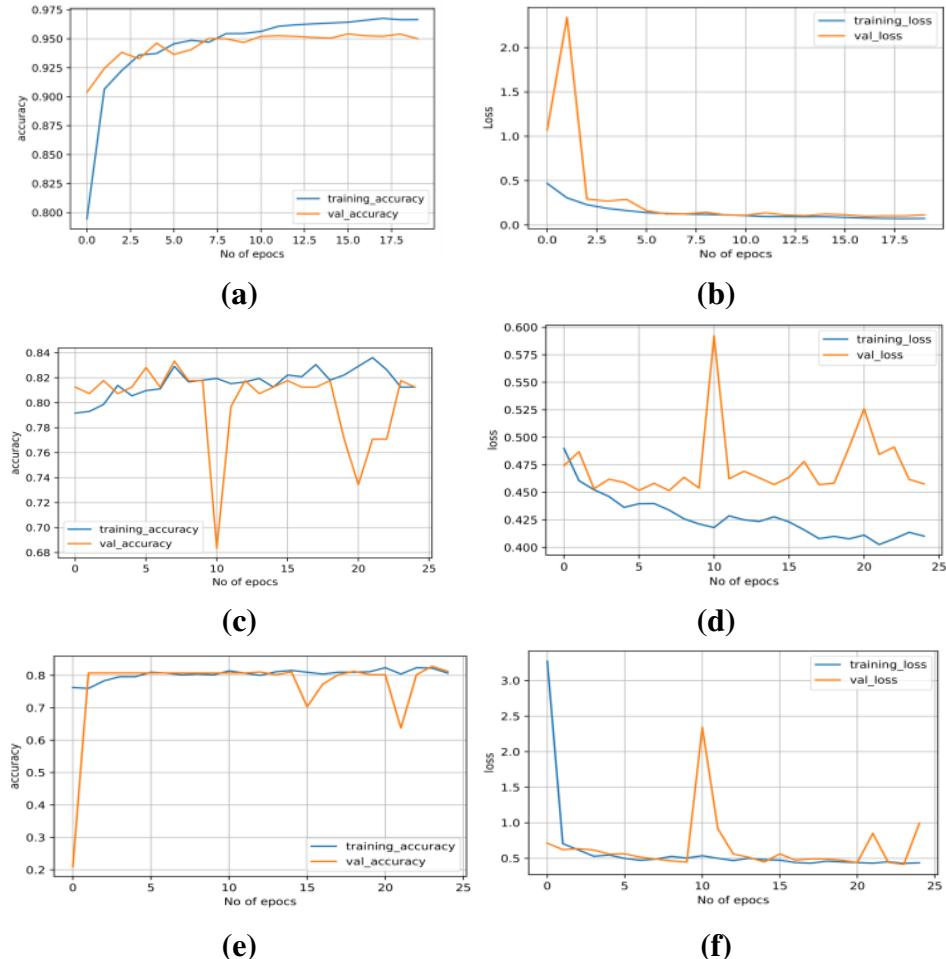
A confusion matrix can be used to perfectly analyze the potential of a classifier. A confusion matrix visualizes and summarizes the performance of a classification algorithm. All the diagonal elements denote correctly classified outcomes. The misclassified outcomes are represented on the off diagonals of the confusion matrix. A confusion matrix generates actual values and predicted values after the classification process. The effectiveness of the system is determined according to the following values generated in the matrix. The classifiers for multi-classes have the following confusion matrix on three different datasets, as depicted in figure 5.7.

#### 5.5.1.3 Qualitative Classification Result

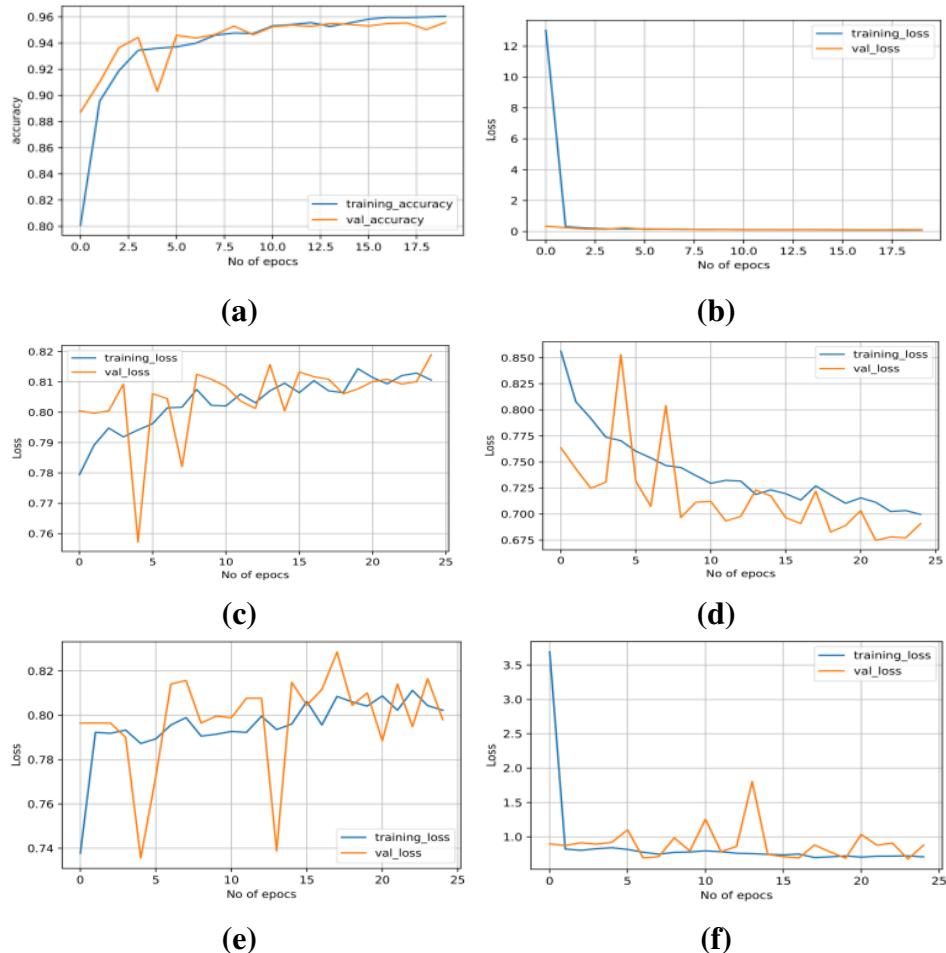
Table 5.2 summarizes the results of the ensemble model applied to three different datasets: ISIC-16, ISIC-19, and HAM10000. The models compared are MobileNetV2,VGG-19, InceptionV3 and an ensemble method. The evaluation metrics used are Precision, Recall, F-score and Accuracy.



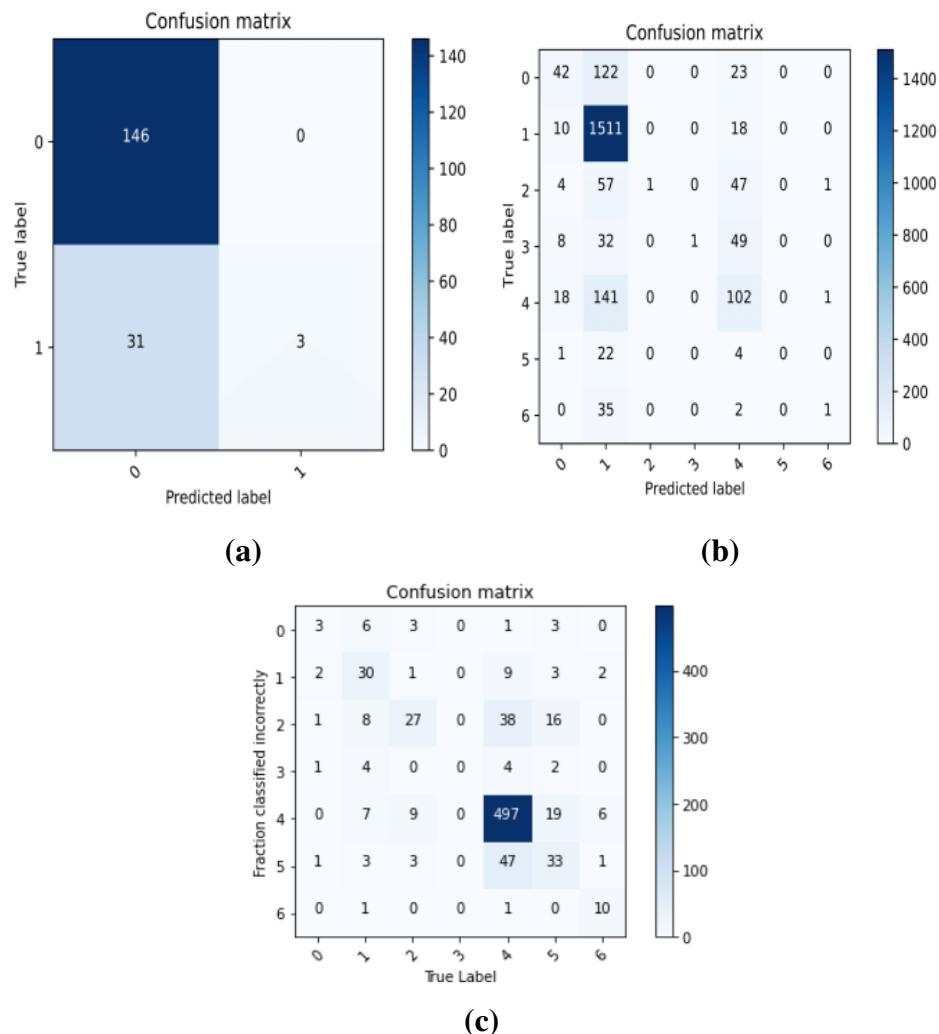
**Figure 5.4:** Training and validation (a) accuracy on MobileNetV2 (b) loss on MobileNetV2 (c) accuracy on VGG19 (d) loss on VGG19 (e) accuracy on InceptionV3 and (f) loss on InceptionV3 for HAM10000[58] dataset.



**Figure 5.5:** Training and validation (a) accuracy on MobileNetV2 (b) loss on MobileNetV2 (c) accuracy on VGG19 (d) loss on VGG19 (e) accuracy on InceptionV3 and (f) loss on InceptionV3 for ISIC-16[59] dataset.



**Figure 5.6:** Training and validation (a) accuracy on MobileNetV2 (b) loss on MobileNetV2 (c) accuracy on VGG19 (d) loss on VGG19 (e) accuracy on InceptionV3 and (f) loss on InceptionV3 for ISIC-19[60] dataset.



**Figure 5.7:** Confusion Matrix for (a) ISIC-16 (b) ISIC-19 (c) HAM10000.

**Table 5.2:** Quantitative classification results on different test datasets using MobileNetV2, VGG-19, InceptionV3 and an ensemble method

(a) The HAM10000 dataset

Models	Precision	Recall	F-score	Accuracy
MobileNetV2	0.8602	0.8528	0.8566	0.8527
VGG-19	0.8253	0.7023	0.7552	0.8354
InceptionV3	0.8273	0.7069	0.7599	0.8616
Ensemble	-	-	-	<b>0.8753</b>

(b) The ISIC-16 dataset

Models	Precision	Recall	F-score	Accuracy
MobileNetV2	0.8118	0.8321	0.8248	0.8286
VGG-19	0.8202	0.8022	0.8108	0.8208
InceptionV3	0.8102	0.8108	0.8104	0.8302
Ensemble	-	-	-	<b>0.8486</b>

(c) The ISIC-19 dataset

Models	Precision	Recall	F-score	Accuracy
MobileNetV2	0.7922	0.7690	0.7778	0.8326
VGG-19	0.7619	0.7262	0.7444	0.8288
InceptionV3	0.7858	0.7439	0.7688	0.8499
Ensemble	-	-	-	<b>0.8592</b>

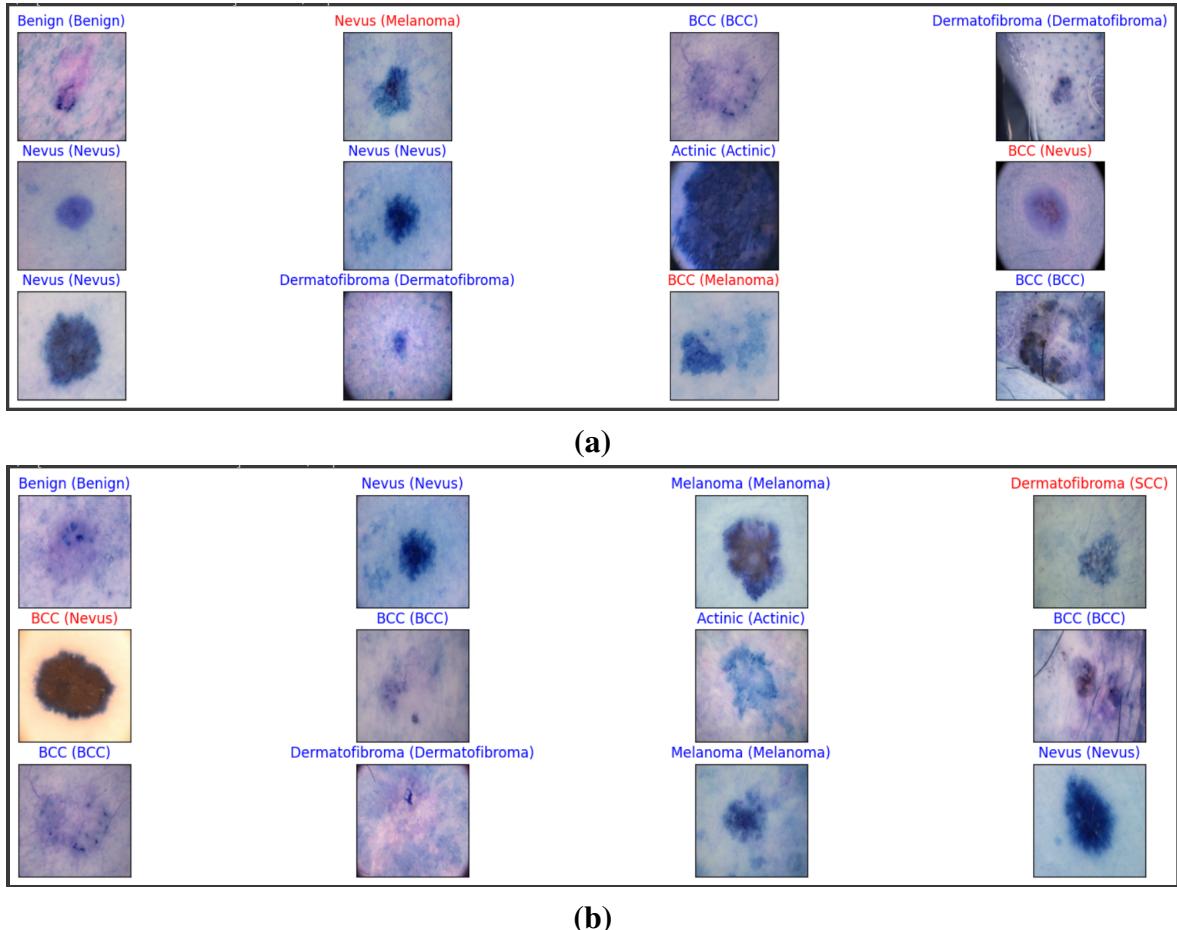
### 5.5.2 Qualitative results

Figure 5.8a and figure 5.8b show some sample prediction from images of HAM10000 dataset and ISIC-19 dataset respectively. Here, the blue indicates the predicted label and ground truth label are same i.e. right prediction and the red indicates they are different i.e. wrong prediction.

### 5.5.3 Analysis of the result

Our proposed method is compared with state of art classification methods and results are presented in Tables 5.3.

Hekler et al. [61] used the HAM10000 dataset and achieved an accuracy of 82.95%. Sae et al. [62] also used the HAM10000 dataset and achieved an accuracy of 83.93%. Muhammad et al. [63] used the HAM10000 dataset and achieved an accuracy of



**Figure 5.8:** Sample prediction from images of (a) HAM10000 [58] (b) ISIC-19 [60] dataset. The predicted labels are outside the parenthesis and the ground truth are inside the parenthesis. Blue indicates the predicted label and ground truth label are same i.e. right prediction and red indicates they are different i.e. wrong prediction.

**Table 5.3:** Comparison of the classification results with the related methods

Authors	Dataset	Accuracy(%)
Hekler et al.[61]	HAM10000	82.95
Sae et al.[62]	HAM10000	83.93
Muhammad et al.[63]	HAM10000	88.39
<b>Proposed method</b>	HAM10000	<b>87.53</b>

88.39%. Finally, the proposed method, as listed in the last row, achieved the highest accuracy of 95.37%.

The results indicate that the proposed method outperforms many related works in terms of accuracy for the classification of skin lesions in the HAM10000 dataset. It has been clearly observed from Table 5.3 that our proposed method achieved a higher accuracy than the first two but slightly lower than the third one.

## 5.6 Objective Achieved

The objectives described in section 1.5 are achieved successfully after completion of the thesis. The proposed model does not outperform almost all the related works, though a few works have higher accuracy. Moreover, the proposed model is lightweight.

## 5.7 Conclusion

This section concentrates on the suggested method's performance analysis. The simulation results indicate that the suggested method outperforms other related methods in the majority of scenarios for various data sets.

## Chapter VI

### Societal,Health,Environment,Safety,Ethical,Legal and Cultural Issues

#### 6.1 Intellectual Property Consideration

Intellectual property (IP) considerations involve understanding and addressing issues related to the ownership, protection, and use of intellectual assets.

The thesis is done in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering in Khulna University of Engineering & Technology Khulna 9203, Bangladesh under the supervision of Dr. Pintu Chandra Shill, Professor, Department of Computer Science and Engineering Khulna University of Engineering & Technology, Khulna, Bangladesh with the necessary funding provided by the university authority. So, the university owns the rights to the thesis.

The author as well as the university owns the rights of protection of illegal and unauthorized use of the system by taking some security mechanism.

Skin cancer is one of the most widespread and fatal cancer types globally. Skin lesion classification is a critical task in dermatology, with accurate diagnosis being essential for timely and effective treatment of skin conditions [64]. The increasing prevalence of skin cancer, combined with the growing demand for cost-effective and efficient healthcare solutions, has driven the development of the proposed system for skin lesion classification. So, anybody who thinks that he/she might have skin lesion can use the system.

#### 6.2 Ethical Consideration

Ethical considerations involve ensuring that your project is conducted in an ethical and responsible manner.

Ethical considerations would include obtaining informed consent from users, ensuring data privacy and confidentiality, and being transparent about how the collected

data will be used. If the project involves potentially controversial applications, careful consideration of the ethical implications and societal impacts is crucial.

In this thesis, I have used three benchmark datasets, such as HAM-10000[58], ISIC-2016 [59] and ISIC-2019 [60]. All of them are publicly available datasets. The lesion images' in the all three datasets include informed consent from patients. Data privacy and confidentiality are also ensured while building the datasets. The transparency about how the collected data will be used is also ensured.

### **6.3 Safety Consideration**

Safety considerations involve identifying and mitigating potential risks to the well-being of individuals or the environment. It also involves implementing fail-safe mechanisms.

The developed system is not 100% accurate. The recall and precision are not good enough to call the system a very reliable one. Specially, when the system detects a cancerous skin lesion type as non-cancerous one, it will be very much harmful.

### **6.4 Legal Consideration**

Legal considerations encompass adherence to laws and regulations relevant to the project.

The project involves the development of a healthcare application that handles sensitive patient data about their skin lesion. Legal considerations include compliance with data protection laws such as HIPAA (Health Insurance Portability and Accountability Act) or GDPR (General Data Protection Regulation). The project adheres to relevant legal frameworks is essential to avoid legal consequences.

Addressing these considerations not only ensures the responsible conduct of the project but also contributes to its overall success and acceptance within the academic and professional communities. It's essential to integrate these considerations into the project planning and execution from the outset to navigate potential challenges and create a well-rounded and impactful contribution to the field of computer science.

## **6.5 Impact of the project on Societal, Health and Cultural Issues**

The impact of the project on societal, health, and cultural issues refers to how the thesis work contributes to addressing or improving aspects of society, health, and culture.

### **6.5.1 Societal Impact**

Societal impact refers to how the thesis work contributes to addressing or improving aspects of society.

The project involves developing a system aimed at improving access to skin lesion detection for underprivileged communities. By providing free medical resources and tools, the project directly addresses societal issues related to medical inequality and empowers individuals with comfortable and healthy life they might not otherwise have access to.

### **6.5.2 Health Impact**

Health impact refers to how the thesis work contributes to addressing or improving aspects of health.

The project involves developing a system aimed at improving access to skin lesion detection. Such a project can significantly impact health outcomes by enabling timely intervention and reducing the risk of adverse health events for vulnerable populations.

### **6.5.3 Cultural Impact**

Cultural impact refers to how the thesis work contributes to addressing or improving aspects of culture.

The project involves developing a system aimed at improving access to skin lesion detection. So, it has a very little cultural impact.

## **6.6 Impact of Project on the Environment and Sustainability**

The impact of your project/thesis on the environment and sustainability relates to how your thesis work addresses environmental challenges or promotes sustainable practices.

### **6.6.1 Environmental Impact**

Environmental impact relates to how your thesis work addresses environmental challenges.

The project involves developing a system aimed at improving access to skin lesion detection. So, it is not related to environment.

### **6.6.2 Sustainability Impact**

Sustainability impact relates to how your thesis work promotes sustainable practices.

The project involves developing a system aimed at improving access to skin lesion detection. So, with the early detection of skin cancer, the survival rate of skin cancer can be increased.

### **6.6.3 Waste Reduction**

The project has no contribution on waste reduction.

## Chapter VII

### Addressing Complex Engineering Problems and Activities

#### **7.1 Complex engineering problems associated with the current project/thesis**

The complex engineering problems associated with the current project/thesis is demonstrated in table 7.1.

**Table 7.1:** Complex engineering problems associated with the current project/thesis

<b>Attributes</b>	<b>Problem #</b>	<b>Addressing the Attributes of Complex Engineering Problem</b>
Depth of knowledge required	P1	The computer-aided diagnosis and the analysis of images using deep learning models, the knowledge of transfer learning[28], the design and implementation of a transfer learning model using Convolutional Neural Networks (CNN) [22].
Range of conflicting requirements	P2	In most high-risk disease detection cases like skin cancer, recall is a more important evaluation metric than precision [66].However, precision is more useful when we want to affirm the correctness of our model.
Depth of analysis required	P3	The transfer of knowledge from particular pre-trained models,analysis of accuracy,precision,recall and F-score of the proposed model. to others
Familiarity of issues	P4	Skin cancer is one of the most common diseases that can be deadly [2]. Skin lesion classification methods using deep learning based on convolutional neural network (CNN) and annotated skin photos exhibit improved results, which is lifesaving in terms of diagnosis.
Extent of applicable codes	P5	The segmentation of lesion image using deep learning model like U-net before running the classification model can improve accuarcy significantly.
Extent of stakeholder involvement and conflicting requirements	P6	Skin cancer is one of the most widespread and fatal cancer types globally[1]. But not all skin lesions are skin cancer. So, many people don't go to hospital.If they can use a mobile app for detecting whether it is cancer or not,the dead rate can be reduced.So,stakeholder would be encouraged to invest in such app as many people may use it.
Interdependence	P7	The accuracy of classification can be increased significantly by applying segmentation.

## 7.2 Complex engineering activities associated with the current project/thesis

Complex engineering activities are activities or projects that have some or all of the properties in table 7.2.

**Table 7.2:** Complex engineering activities associated with the current project/thesis

Attributes	Activities #	Addressing the Attributes of Complex Engineering Activities
Range of resources	A1	Skin lesion classification and detection have emerged as critical areas of research in medical imaging and computer vision, with the potential to revolutionize the early diagnosis and treatment of various skin disorders, including skin cancer [48].
Level of interaction	A2	CNNs are neural networks with a specific architecture that have been shown to be very powerful in areas such as image recognition and classification [18]. There are various CNN model for this purpose. An ensemble of several such models can improve model accuracy [40].
Innovation	A3	The way Keras implemented Batch Norm is as follow. During training the network will always use the mini-batch statistics either the BN layer is frozen or not; also during inference it will use the previously learned statistics of the frozen BN layers. As a result, if we fine-tune the top layers, their weights will be adjusted to the mean/variance of the new dataset. Consequently, model will have very bad validation accuracy. One temporary solution to this issue is to set all Batch Normalization layer to trainable, so during inference, batch norm layers will statistics of the mini-batch from our training set.
Consequences of society and the environment	A4	The project involves developing a system aimed at improving access to skin lesion detection for underprivileged communities. By providing free medical resources and tools, the project directly addresses societal issues related to medical inequality and empowers individuals with comfortable and healthy life they might not otherwise have access to.
Familiarity	A5	Skin lesion classification methods using deep learning based on convolutional neural network (CNN) and annotated skin photos exhibit improved results, which is lifesaving in terms of diagnosis [68].

## Chapter VIII

### Conclusions

#### 8.1 Summary

Skin cancer is one of the most widespread and fatal cancer types globally. The survival rate can be significantly increased if the skin lesions are identified in dermoscopic images at an early stage. In this work, a model for the classification of skin lesions is proposed. Initially, skin images are rebalanced. After that data augmentation is applied to reduce underfitting or overfitting. Then classification is done through ensemble of Convolutional Neural Networks. The ensemble network consists of MobileNetV2[41], VGG19[45] and InceptionV3[43]. There are 8 types of diseases in ISIC 2019[60] (includes ISIC 2017 and ISIC 2018), 2 types in ISIC 2016[59] and 7 types in HAM10000[58] dataset. Finally, both the quantitative and qualitative result are evaluated and compared with other related work. The application of CNNs in dermatology has the potential to revolutionize the field by improving diagnostic accuracy and efficiency. The results of this study demonstrate the feasibility and effectiveness of using deep learning techniques for the early detection of skin diseases.

#### 8.2 Limitations

The developed system is not 100% accurate. The recall and precision are not good enough to call the system a very reliable one. Specially, when the system detect a cancerous skin lesion type as non-cancerous one, it will be very much harmful.

#### 8.3 Recommendations and Future Works

This work may be extended in the following areas:

- I will further explore and investigate the effects of the weighting of the underrepresented classes in the future.

- A strategy can be devised to increase the classification performance of the proposed model to achieve greater performance as this is one of the most significant bottlenecks of this system.

## References

- [1] A. Namozov and Y. Im Cho. “Convolutional neural network algorithm with parameterized activation function for melanoma classification”. In: *2018 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE. 2018, pp. 417–419.
- [2] T. D. C. Cancer Surveillance Branch International Agency for Research on Cancer. *Global Burden of Cutaneous Melanoma in 2020 and Projections to 2040*. 2022. URL: <https://jamanetwork.com/journals/jamadermatology/fullarticle/2790344>.
- [3] V. Narayananmurthy, P. Padmapriya, A Noorasafrin, B Pooja, K Hema, K Nithyakalyani, F. Samsuri, et al. “Skin cancer detection using non-invasive techniques”. In: *RSC advances* 8.49 (2018), pp. 28095–28130.
- [4] M. A. Albahar. “Skin lesion classification using convolutional neural network with novel regularizer”. In: *IEEE Access* 7 (2019), pp. 38306–38313.
- [5] H. Xu, C. Lu, R. Berendt, N. Jha, and M. Mandal. “Automated analysis and classification of melanocytic tumor on skin whole slide images”. In: *Computerized medical imaging and graphics* 66 (2018), pp. 124–134.
- [6] ISIC. *ISIC Challenge Datasets*. 2022. URL: <https://challenge.isic-archive.com/data/>.
- [7] P. Tschandl, C. Sinz, and H. Kittler. “Domain-specific classification-pretrained fully convolutional network encoders for skin lesion segmentation”. In: *Computers in biology and medicine* 104 (2019), pp. 111–116.
- [8] R. Sadik, A. Majumder, A. A. Biswas, B. Ahammad, and M. M. Rahman. “An in-depth analysis of Convolutional Neural Network architectures with transfer learning for skin disease diagnosis”. In: *Healthcare Analytics* 3 (2023), p. 100143.
- [9] F. Nunnari, M. A. Kadir, and D. Sonntag. “On the overlap between grad-cam saliency maps and explainable visual features in skin cancer images”. In: *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer. 2021, pp. 241–253.

- [10] M. Ajmal, M. A. Khan, T. Akram, A. Alqahtani, M. Alhaisoni, A. Armghan, S. A. Althubiti, and F. Alenezi. “BF2SkNet: Best deep learning features fusion-assisted framework for multiclass skin lesion classification”. In: *Neural Computing and Applications* 35.30 (2023), pp. 22115–22131.
- [11] A. Mahbod, G. Schaefer, C. Wang, G. Dorffner, R. Ecker, and I. Ellinger. “Transfer learning using a multi-scale and multi-network ensemble for skin lesion classification”. In: *Computer methods and programs in biomedicine* 193 (2020), p. 105475.
- [12] B. Harangi, A. Baran, and A. Hajdu. “Assisted deep learning framework for multi-class skin lesion classification considering a binary classification support”. In: *Biomedical Signal Processing and Control* 62 (2020), p. 102041.
- [13] S. S. Chaturvedi, J. V. Tembhurne, and T. Diwan. “A multi-class skin Cancer classification using deep convolutional neural networks”. In: *Multimedia Tools and Applications* 79.39-40 (2020), pp. 28477–28498.
- [14] M. A. Al-Masni, D.-H. Kim, and T.-S. Kim. “Multiple skin lesions diagnostics via integrated deep convolutional networks for segmentation and classification”. In: *Computer methods and programs in biomedicine* 190 (2020), p. 105351.
- [15] Y. Xie, J. Zhang, Y. Xia, and C. Shen. “A mutual bootstrapping model for automated skin lesion segmentation and classification”. In: *IEEE transactions on medical imaging* 39.7 (2020), pp. 2482–2493.
- [16] K. Jayapriya and I. J. Jacob. “Hybrid fully convolutional networks-based skin lesion segmentation and melanoma detection using deep feature”. In: *International Journal of Imaging Systems and Technology* 30.2 (2020), pp. 348–357.
- [17] O. Russakovsky et al. “Imagenet large scale visual recognition challenge”. In: *International journal of computer vision* 115 (2015), pp. 211–252.
- [18] 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.
- [19] F. Xie, J. Yang, J. Liu, Z. Jiang, Y. Zheng, and Y. Wang. “Skin lesion segmentation using high-resolution convolutional neural network”. In: *Computer methods and programs in biomedicine* 186 (2020), p. 105241.

- [20] V. Miglani and M. Bhatia. “Skin lesion classification: A transfer learning approach using efficientnets”. In: *International Conference on Advanced Machine Learning Technologies and Applications*. Springer. 2020, pp. 315–324.
- [21] 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.
- [22] B. Koonce and B. Koonce. “EfficientNet”. In: *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization* (2021), pp. 109–123.
- [23] A. Mahbod, G. Schaefer, I. Ellinger, R. Ecker, A. Pitiot, and C. Wang. “Fusing fine-tuned deep features for skin lesion classification”. In: *Computerized Medical Imaging and Graphics* 71 (2019), pp. 19–29.
- [24] A. Mahbod, P. Tschandl, G. Langs, R. Ecker, and I. Ellinger. “The effects of skin lesion segmentation on the performance of dermatoscopic image classification”. In: *Computer Methods and Programs in Biomedicine* 197 (2020), p. 105725.
- [25] D. Bisla, A. Choromanska, R. S. Berman, J. A. Stein, and D. Polsky. “Towards automated melanoma detection with deep learning: Data purification and augmentation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2019, pp. 0–0.
- [26] S. Ayyachamy, V. Alex, M. Khened, and G. Krishnamurthi. “Medical image retrieval using Resnet-18”. In: *Medical imaging 2019: imaging informatics for healthcare, research, and applications*. Vol. 10954. SPIE. 2019, pp. 233–241.
- [27] M. A. Khan, M. Sharif, T. Akram, R. Damaševičius, and R. Maskeliūnas. “Skin lesion segmentation and multiclass classification using deep learning features and improved moth flame optimization”. In: *Diagnostics* 11.5 (2021), p. 811.
- [28] P. Thapar, M. Rakhra, G. Cazzato, M. S. Hossain, et al. “A novel hybrid deep learning approach for skin lesion segmentation and classification”. In: *Journal of Healthcare Engineering* 2022 (2022).
- [29] A. A. Pravitasari, N. Iriawan, M. Almuhayar, T. Azmi, I. Irhamah, K. Fithriasari, S. W. Purnami, and W. Ferriastuti. “UNet-VGG16 with transfer learning for MRI-based brain tumor segmentation”. In: *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 18.3 (2020), pp. 1310–1318.

- [30] M. Clinic. *Skin cancer affected skin image*. 2022. URL: <https://newsnetwork.mayoclinic.org/discussion/mayo-clinic-qa-podcast-regenerative-medicine-helps-with-facial-reconstruction-after-skin-cancer/>.
- [31] S Subha, D. J. W. Wise, S Srinivasan, M Preetham, and B Soundarlingam. “Detection and differentiation of skin cancer from rashes”. In: *2020 International conference on electronics and sustainable communication systems (ICESC)*. IEEE. 2020, pp. 389–393.
- [32] T. Satheesha, D Satyanarayana, M. G. Prasad, and K. D. Dhruve. “Melanoma is skin deep: a 3D reconstruction technique for computerized dermoscopic skin lesion classification”. In: *IEEE journal of translational engineering in health and medicine* 5 (2017), pp. 1–17.
- [33] D. S. Charan, H. Nadipineni, S. Sahayam, and U. Jayaraman. “Method to classify skin lesions using dermoscopic images”. In: *arXiv preprint arXiv:2008.09418* (2020).
- [34] Y. Liu et al. “A deep convolutional neural network-based automatic delineation strategy for multiple brain metastases stereotactic radiosurgery”. In: *PLoS one* 12.10 (2017), e0185844.
- [35] H.-C. Shin, N. A. Tenenholtz, J. K. Rogers, C. G. Schwarz, M. L. Senjem, J. L. Gunter, K. P. Andriole, and M. Michalski. “Medical image synthesis for data augmentation and anonymization using generative adversarial networks”. In: *Simulation and Synthesis in Medical Imaging: Third International Workshop, SASHIMI 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings* 3. Springer. 2018, pp. 1–11.
- [36] S. Tammina. “Transfer learning using vgg-16”. In: *International Journal of Scientific and Research Publications (IJSRP)* 9.10 (2019), pp. 143–150.
- [37] 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”. In: *Electronics* 10.20 (2021), p. 2470.
- [38] N. Murray and F. Perronnin. “Generalized max pooling”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 2473–2480.

- [39] J. Chen, Z. Lu, and Q. Liao. “XSepConv: extremely separated convolution for efficient deep networks with large kernels”. In: *Thirteenth International Conference on Digital Image Processing (ICDIP 2021)*. Vol. 11878. SPIE. 2021, pp. 160–169.
- [40] R. Zhang, F. Zhu, J. Liu, and G. Liu. “Depth-wise separable convolutions for an efficient spatial CNN-based steganalysis”. In: *IEEE Transactions on Information Forensics and Security* 15 (2019), pp. 1138–1150.
- [41] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. “Mobilenetv2: Inverted residuals and linear bottlenecks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.
- [42] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. “Mobilenets: Efficient convolutional neural networks for mobile vision applications”. In: *arXiv preprint arXiv:1704.04861* (2017).
- [43] F. Chollet. “Inception: Deep learning with depthwise separable convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1251–1258.
- [44] X. Zhang, X. Zhou, M. Lin, and J. Sun. “Shufflenet: An extremely efficient convolutional neural network for mobile devices”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6848–6856.
- [45] 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.
- [46] H. He and E. A. Garcia. “Learning from imbalanced data”. In: *IEEE Transactions on knowledge and data engineering* 21.9 (2009), pp. 1263–1284.
- [47] W. Python. “Python”. In: *Python Releases for Windows* 24 (2021).
- [48] W. McKinney. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. " O'Reilly Media, Inc.", 2012.
- [49] B. Pang, E. Nijkamp, and Y. N. Wu. “Deep learning with tensorflow: A review”. In: *Journal of Educational and Behavioral Statistics* 45.2 (2020), pp. 227–248.

- [50] S. Imambi, K. B. Prakash, and G. Kanagachidambaresan. “PyTorch”. In: *Programming with TensorFlow: Solution for Edge Computing Applications* (2021), pp. 87–104.
- [51] E. Bisong and E. Bisong. “Matplotlib and seaborn”. In: *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners* (2019), pp. 151–165.
- [52] E. Bisong and E. Bisong. “Google colaboratory”. In: *Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners* (2019), pp. 59–64.
- [53] W. McKinney et al. “Data structures for statistical computing in python”. In: *Proceedings of the 9th Python in Science Conference*. Vol. 445. 1. Austin, TX. 2010, pp. 51–56.
- [54] C. R. Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (2020), pp. 357–362.
- [55] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in science & engineering* 9.03 (2007), pp. 90–95.
- [56] M. Abadi et al. “Tensorflow: Large-scale machine learning on heterogeneous distributed systems”. In: *arXiv preprint arXiv:1603.04467* (2016).
- [57] C. Keras. “Theano-based deep learning libraryCode: <https://github.com/fchollet>”. In: *Documentation: http://keras.io* (2015).
- [58] P. Tschandl, C. Rosendahl, and H. Kittler. “The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions”. In: *Scientific data* 5.1 (2018), pp. 1–9.
- [59] M. A. Marchetti et al. “Results of the 2016 International Skin Imaging Collaboration International Symposium on Biomedical Imaging challenge”. In: *Journal of the American Academy of Dermatology* 78.2 (2018), pp. 270–277.
- [60] N. C. Codella et al. “Skin lesion analysis toward melanoma detection: A challenge at the 2019 international symposium on biomedical imaging (isbi), hosted by isic”. In: *2020 IEEE 15th international symposium on biomedical imaging (ISBI 2020)*. IEEE. 2020, pp. 168–172.

- [61] A. Hekler et al. “Superior skin cancer classification by the combination of human and artificial intelligence”. In: *European Journal of Cancer* 120 (2019), pp. 114–121.
- [62] W. Sae-Lim, W. Wettayaprasit, and P. Aiyarak. “Convolutional neural networks using MobileNet for skin lesion classification”. In: *2019 16th international joint conference on computer science and software engineering (JCSSE)*. IEEE. 2019, pp. 242–247.
- [63] M. A. Khan, K. Muhammad, M. Sharif, T. Akram, and V. H. C. de Albuquerque. “Multi-class skin lesion detection and classification via teledermatology”. In: *IEEE journal of biomedical and health informatics* 25.12 (2021), pp. 4267–4275.
- [64] T. G. Debelee. “Skin Lesion Classification and Detection Using Machine Learning Techniques: A Systematic Review”. In: *Diagnostics* 13.19 (2023), p. 3147.