Bachelor of Science in Computer Science
April 2024

Cross-Dataset Generalization of Deep Learning Models for Melanoma Prediction

A Comparative Study of AlexNet, GoogLeNet, DenseNet, ResNet and Ensemble Approaches Beyond HAM10000

Venkata Sai Abhiram Akula
Vaishnavi Bojja

Faculty of Computer Science Engineering, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Computer Science Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science. The thesis is equivalent to 10 weeks of full-time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:
Author(s):
Venkata Sai Abhiram Akula
E-mail: veak23@student.bth.se

Vaishnavi Bojja
E-mail: vabo23@student.bth.se

University advisor:
Dr. Suejb Memeti, Ph.D
Department of Department of Computer Science

Faculty of Computer Science Engineering
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

Background:
Melanoma is a severe form of skin cancer that requires early diagnosis for effective treatment. Deep learning models, especially Convolutional Neural Networks (CNNs), have shown promise in medical image classification. This study aims to evaluate the generalization capabilities of individual deep learning models and an ensemble approach for melanoma prediction, trained on the HAM10000 dataset and validated on other test datasets, such as PH2 and DermNet.
Objectives:
The primary objective of this thesis is to evaluate the performance of AlexNet, DenseNet, ResNet, GoogLeNet, and ensemble model for classifying melanoma images by training the models on HAM10000 dataset and testing on PH2 and DermNet datasets. The secondary objective is to analyse the generalization ability of HAM10000 dataset to produce effective models.
Methods:
The datasets for the study are collected from Kaggle and official databases and are preprocessed using techniques like normalization and LabelEncoder. The models are compiled after applying data augmentation and reinforcement. Next, the models are trained with different train_test_split configurations (80%, 90% and 100%) of HAM10000 dataset and evaluated all the models (12 individual models, 4 per training group and 3 ensemble models, 1 per training group) on a custom dataset (merged PH2 and DermNet datasets).
Results:
The ensemble model (en_V100) achieved the best overall performance, with the highest accuracy of 83.56%, precision of 0.947, recall of 0.943, F1-score of 0.945, and the lowest Hamming Loss of 0.164. Amongst the individual models, ResNet demonstrated best performance due to its architecture with residual connections. The AlexNet models performed poorly in every training group due to its simple architecture failing to capture the complex patterns in the data. Additionally, HAM10000 dataset is proved to be effective as all the models performed with generalizable results.
Conclusions:
The ensemble approach outperformed the individual models in every training group, suggesting combining multiple architectures will results in an overall reliable model for melanoma prediction. The models trained on the HAM10000 dataset showed good generalization when evaluated on other diverse datasets (PH2 and DermNet), indicating the effectiveness of the dataset.

Keywords: Melanoma Prediction, Deep Learning, Medical Image Analysis, Convolutional Neural Networks, Ensemble Models.

Acknowledgments

We would like to express our deepest gratitude to our supervisor, Dr. Suejb Memeti, Ph.D for his constant support, expertise and insightful feedback throughout this research. We are grateful for the collaborative spirit and supportive environment fostered by Dr. Memeti.

We also extend our sincere thanks to Dr. Prashanth Goswami, our examiner, for providing constructive feedback and comments on the project plan. The feedback from Dr. Goswami has been crucial in strengthening the quality of our study.

Authors:
Venkata Sai Abhiram Akula
Vaishnavi Bojja

Contents

List of Figures

List of Tables

List of Acronyms

UV Ultra Violet

ML Machine Learning

ISIC International Skin Imaging Collaboration

HAM10000 Human Against Machine with 10000 training images

PH2 Dermatology Research Group at Pedro Hispano

CNN Convolutional Neural Network

PCA Principal Component Analysis

VAE Variational Autoencoder

GAN Generative Adversarial Network

TP True Positive

TN True Negative

FP False Positive

FN False Negative

PIL Python Imaging Library

IoU Intersection over Union

Chapter 1

Introduction

Cancer is a huge cluster of diseases that can affect any organ in the body. A cancerous tissue is characterized by uncontrollable and abnormal growth of cells. The prime factor contributing to fatality in cancer patients is metastasizing. It is a phenomenon when abnormal growth invades adjacent organs or other organs in the body [44]. Melanoma is an aggressive form of skin cancer that starts in cells called melanocytes. These are cells found in the epidermis and have melanin that gives colour to the skin. Melanoma occurs when skin damage from sunburns or other concentrated sources of UV radiation like tanning beds causes mutations in these cells resulting in uncontrolled aggressive growth beyond boundaries [37]. Melanoma claimed 150,000 lives in 2020, standing as the 17th most common cancer worldwide [43]. With a plethora of reasons like a compromised ozone layer, the cases have only been increasing, hence finding more resourceful ways to help detect these occurrences is a continuous effort. In this thesis, computer vision techniques are being experimented upon to test their efficiency.

Traditional methods to detect skin cancer involve visual analysis of lesions by naked eye of a professional and dermoscopy results. These readings may be subject to ambiguity. Only histological analysis, derived from surgical extraction of sample has enough validity to ensure accurate skin cancer diagnosis [18]. Hence finding novel automation methods for diagnostic augmentation is of utmost priority.

Machine learning (ML) is a sub field of artificial intelligence that gives machines the ability to imitate human behaviour. The performance of ML algorithms adapts with an increase in the number of available samples during the learning process [20]. Deep learning is a sub-domain of ML that trains computers to imitate natural human traits. It offers better performance parameters than conventional ML algorithms. Deep Learning is a neural network having three or more layers which try to simulate the human brain allowing it to learn from huge amounts of data [23]. It works similar to a neuron in the human brain where it transmits electrical impulses between components in the nervous system, here the perceptron is a similar unit that receives input signals and transforms them as output signals. This perceptron can be stacked to form layers, each with an assigned responsibility to understand different aspects of the input. The layers are meant to predict, refine and optimize the learning model [31]. Ensemble learning uses multiple learning algorithms to obtain better predictive performance, robustness and generalizability than could be obtained from any of the constituent learning algorithms alone. This approach allows the production of

better predictive performance compared to a single model [36]. The main causes of the difference in predicted and actual values are variance, noise, and bias. The ensemble model helps to reduce these factors. In this thesis, deep learning architectures such as AlexNet, GoogLeNet, ResNet, and DenseNet are proposed for evaluation as skin lesions caused by melanoma have varying sizes, shapes, and colours.

A comparative analysis of mentioned model performances along with their ensembled counterpart is executed. The widely used HAM10000 dataset is investigated through validation using data from other sources [24].The HAM10000 dataset comprises 10,015 dermatoscopic images of skin lesions, encompassing various diagnostic categories. The dataset includes images from different anatomical sites and lesions types, providing a well-annotated collection for the study of skin cancer. Validation of trained model is proposed to be implemented upon Dermnet and PH2 datasets. [19] [28]. One of the key challenges in the field of deep learning-based skin cancer detection is the ability of these models to generalize beyond the specific datasets used for training. The widely used HAM10000 dataset provides a well-annotated collection of dermoscopic images, but it may not fully represent the diversity of skin lesions encountered in real-world clinical settings. Factors such as differences in skin types, imaging conditions, and the prevalence of certain lesion types can impact the performance of deep learning models when applied to new datasets.

In order to accurately predict the image data, Convolutional neural network (CNN) architectures are most suited for medical image analysis. AlexNet has 8 layers namely, 5 convolution layers, 3 max-pooling layers out of which there are 2 Normalized layers, 2 fully connected layers and 1 SoftMax layer [17]. It has more filters per layer than other deep learning architectures. GoogLeNet is made up of multiple inception modules which consist of several convolutional layers of different kernel sizes [38]. ResNet or Residual Network consists of residual blocks where the input to a layer is attached to the output to that layer so that the residual or difference can be learnt to utilize identity mappings [30]. DenseNet is characterized by densely connected layers where the input from all the preceding layers is attached to the output layer. Every layer is interconnected in a dense block [22].

By experimenting with cross-dataset generalization with iterations, validation of trained model is proposed to be implemented upon Dermnet and PH2 datasets. This thesis aims to contribute to the advancement of deep learning-based skin cancer detection, ultimately leading to improved early diagnosis and better patient outcomes. The following Figure 1.1 is a simple representation showing the basic execution, the input and the possible output labels:

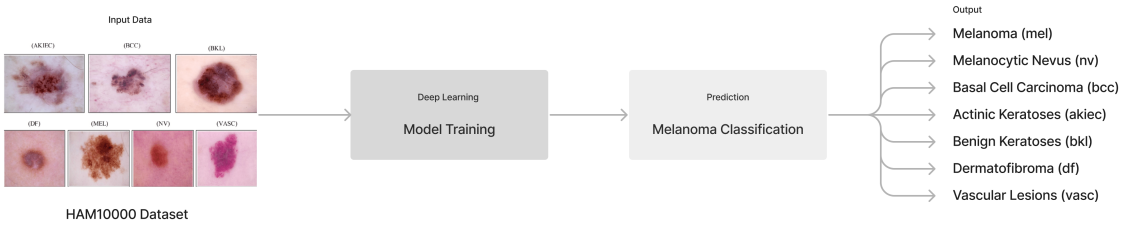


Figure 1.1: Multiclass Classification on Input images

1.1 Aim and objectives

The **aim** of this research is to assess the cross-dataset generalization capabilities of ensemble learning model, specifically those formed by merging AlexNet, GoogLeNet, DenseNet, and ResNet. The major focus is on training individual deep learning models and the ensemble model on the HAM10000 dataset for melanoma prediction and evaluating their performance across other datasets(Dermnet, PH2, ISIC) within the melanoma domain [19] [28].

The **objectives** of this research involves:

1. Using the HAM10000 dataset, train the individual deep learning models (AlexNet, GoogLeNet, DenseNet, ResNet) for the prediction of melanoma.

2. Combine predictions from AlexNet, GoogLeNet, DenseNet and ResNet using ensemble learning approaches to create a consolidated model. Using the HAM10000 dataset as a test dataset, assess and contrast the performances of individual models with the ensemble model.

3. Evaluate the HAM10000 dataset's effectiveness in improving the generalization performance of deep learning models. This entails a careful analysis of the dataset's potential to advance the development of more accurate generalized models.

1.2 Research Questions

RQ1: What advantages and challenges does the ensemble model, combining AlexNet, GoogLeNet, DenseNet, and ResNet, pose when trained on the HAM10000 dataset and evaluated on diverse data sources within the melanoma domain, compared to its individual counterparts?

Motivation: By assessing the ensemble model against individual counterparts, insights into the collective performance, strengths, and weaknesses of the ensemble approach will be gained. This exploration is crucial for optimizing model selection

and configuration in melanoma prediction systems.

RQ2: How effective is the HAM10000 dataset while generalizing performance in proposed models when investigated by testing the trained model on other datasets (Dermnet, PH2, ISIC)?

Motivation: The motivation behind this question is rooted in the significance of datasets in training robust and generalized deep learning models. Analyzing the HAM10000 dataset's impact on model generalization provides insights into its efficiency and suitability as a diverse melanoma dataset.

1.3 Ethical, societal and sustainability aspects

- **Bias:** If the models are trained predominantly on specific datasets, they might exhibit bias and lack generalization in populations not well-represented in the training data. Statistically people with African, Asian and Hispanic ancestries suffer from late diagnosis.
- **Privacy:** Medical data is sensitive to handle, ensuring that the data used is anonymized and that proper consent procedures are followed is crucial to protect individuals' privacy.
- **Education and Awareness:** Introducing deep learning models for melanoma detection requires educating both healthcare professionals and the general public about their capabilities, limitations, and the importance of regular medical check-ups.
- **Model Maintenance:** Ensuring the continued accuracy and relevance of melanoma prediction models beyond the initial research phase is important in order to keep up with evolving medical knowledge. Updating the datasets to cater to all groups is essential.
- **Explainability:** It is essential for healthcare professionals to completely understand and vouch for the tools that they are using to help their diagnostic decisions. This helps both the patients and professionals to better rely on said tools made for efficiency [41].
- **Transparency:** Most Deep Learning Networks use a 'black box' approach where the inner working and decision-making process are not visible. In high stake decision making process such as melanoma detection for diagnoses where this decision may affect treatment plans when model is augmented with real world health care professionals.

1.4 Scope

The scope of this thesis is to examine the performance of different deep learning models in the domain of melanoma detection. This research evaluates the four deep

learning models: AlexNet, GoogLeNet, DenseNet, and ResNet, and assesses their generalization when trained on the HAM10000 dataset and tested on other melanoma datasets like Dermnet, and PH2. It also explores the potential benefits of an ensemble model that combines these architectures.

This study also examines any inherent biases or limitations in the data and models that could affect their reliability in medical applications. The findings will help identify which model or ensemble of models provides the most reliable predictions for melanoma detection.

This thesis does not cover broader issues related to the development of deep learning algorithms or the complete range of potential datasets for melanoma prediction. Instead, the focus is on evaluating the selected models in the context of the specified datasets to understand their generalization capabilities and clinical relevance.

1.5 Outline

This section outlines the structure of this study. Chapter 1 introduces the focus of the study, objectives, research questions along with an overview of importance of the study. Chapter 2 provides background for the thesis by explaining the key concepts, terminology, and datasets used in the study. This chapter also covers topics such as machine learning, deep learning, and neural networks. Chapter 3 presents the literature review for the thesis, summarizing the previous work in melanoma prediction and identifying gaps this thesis aims to fill. Chapter 4 contains the methodology of the study, including data collection, preprocessing, model architectures and training. It also discussed the ensemble approach and model evaluation techniques. Chapter 5 presents the results obtained with focus on the chosen evaluation metrics and provides insights into the obtained results. Chapter 6 contains the discussion on the results obtained by answering the research questions and reflections. Chapter 7 concludes the thesis by summarizing the findings and suggesting future possible research directions. The references section lists all sources cited in this thesis.

Chapter 2

Background

2.1 Machine Learning

ML is a field of artificial intelligence that allows computer systems to learn from data, adapt, and improve their performance over time without being explicitly programmed. A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.- Tom Mitchell,1997 [3].
There are several types of ML, which vary based on the level of human intervention, the type of data they work with, the learning process, and the generalization strategy as show in the table 2.1 below [3]:

Learning Type	**Data Context**	**Learning Approach**
Human Intervention	Labeled Data	Supervised Learning
	Unlabeled Data	Unsupervised Learning
	Partially labeled data	Semi-supervised Learning
	Defined policies	Reinforcement Learning
Incremental Learning from live data	Fully adaptable to live data	Online Learning
	Only Offline data	Batch Learning
Generalising Ability	Measure of similarity	Instance-based Learning
	Learned Prediction model	Model-based Learning

Table 2.1: Types of ML based on data context and learning approach

- **Human Intervention**: Refers to the degree of labeled data or feedback required for the learning process.

  (a) **Supervised Learning** Simple supervised learning model involves learning from labeled data, where each input is associated with a specific output,i.e. classification tasks.

  (b) **Unsupervised Learning** Unsupervised learning model involves working with unlabeled data, focusing on discovering patterns and relationships.i.e. clustering tasks.

  (c) **Semi-Supervised Learning** Here, the model combines both labeled and unlabeled data to improve learning outcomes.

(d) **Reinforcement Learning** Model relies on a feedback loop with defined policies, where the agent learns by receiving rewards or penalties based on its actions.

- **Adaptability**: Indicates whether the learning process is continuous or batch-oriented.
  (a) **Online Learning** The model continuously adapts to new data as it becomes available.
  (b) **Batch Learning** Data is processed in fixed batches, with learning occurring in discrete stages.
- **Generalization**: Describes the approach to generalization used in ML.
  (a) **Instance-Based Learning** It relies on measuring similarity and *learning patterns by heart* to make predictions.
  (b) **Model-Based Learning** This approach involves creating a model or representation of the data to make predictions.

2.2 Deep Learning and Convolutional Neural Networks

Deep Learning is a type of unsupervised ML algorithm which processes the data in multiple layers hence extracting more features hidden deep. It works similar to a neuron in the human brain where it transmits electrical impulses between components in the nervous system. Here, the perceptron is a similar unit that receives input signals and transforms them as output signals. This perceptron can be stacked to form layers, each with an assigned responsibility to understand different aspects of the input. The layers are meant to predict, refine and optimize the learning model [31].

In tasks such as object detection and image classification,**CNNs** utilize three-dimensional data. Neural networks are composed of interconnected nodes, similar to the neurons in the human brain. These nodes are organized into layers, with an input layer, one or more hidden layers, and an output layer. Each node in the network has an associated weight and threshold value, when the sum of inputs at a layer exceeds threshold value, the output is sent to the next layer in the model. This ability of neural networks to learn complex patterns and relationships in data is what makes them so powerful [12]. CNNs are made up of three main concepts: Local fields, shared weights, and pooling [31].

- **Local field:** It is the localised portion of input image that the model is currently associated with.
- **Shared Weights:** Each local field has a weight and bias which can be commonly used for certain features hence reducing model training time and cost.
- **Pooling Layers:** Pooling layers are used to map the output from convolutional layer onto a feature map where the most important information from image is retained.

2.3 Neural Network Algorithms

This section briefly introduces some CNN architectures namely AlexNet, ResNet, DenseNet and GoogleNet.

2.3.1 Alexnet

AlexNet is one of the earliest CNNs to demonstrate effectiveness of deep learning models in image classification tasks. The architecture consists of 8 layers in total out of which 5 are convolutional layers and are followed by 3 fully connected layers. It uses max pooling and applies ReLU (Rectified Linear Unit) activation to introduce non-linearity [23].

2.3.2 ResNet

ResNet or Residual Network is a neural network which introduces residual connections which allows deep neural networks to be trained effectively. Most common ResNet models have 18, 34, 50, 101, or 152 layers. In the ResNet the residual blocks perform identity mapping and learn residual functions which helps the network to learn deeper representations [30].

2.3.3 DenseNet

DenseNet also called a densely connected convolutional network is an extension to the ResNet where it builds upon the concept of residual connections and adds dense connectivity to them in a feed-forward manner. This leads to more compact and efficient CNN models than traditional as it promotes feature reuse and reduction in number of parameters. Most common DenseNet models have 121, 169, 201, 264 layers [22].

2.3.4 GoogLeNet

GoogLeNet or Inception is a very popular neural network that uses a unique "Inception module" that applies filters of different sizes (1x1, 3x3, 5x5) in parallel, allowing the network to extract features parallelly. It also employs 1x1 convolutions for dimensionality reduction. There are 22 layers in a typical GoogleNet architecture along with 9 inception modules [38].

These CNN architectures have made significant contributions to skin cancer research due to their classification abilities.

2.4 Popular Melanoma Datasets

In the context of melanoma prediction and skin cancer classification, many datasets are popular for training the ML and deep learning models. These datasets are a comprehensive source for dermatoscopic images which are essential for developing ML

and deep learning algorithms. This section introduces three widely used dermoscopic datasets which are as follows:

2.4.1 HAM10000

Human Against Machine with 10000 dermatoscopic images (HAM10000) is the most used dataset in many deep learning research papers. HAM10000 has 10,015 images of various types of skin lesions including melanoma and benign conditions [24]. Each image is labeled with its diagnosis in the metadata file. This dataset has become a benchmark for skin cancer research.

2.4.2 PH2

Ph2 dataset contains dermatoscopic images especially pigmented skin lesions. The entire dataset contains 200 high quality images with their diagnosis along with metadata. The PH2 dataset contains ROI (region of interest) outline images also which corresponds to the dermatoscopic images. The dataset is annotated by dermatologists providing reliable data for ML models [28].

2.4.3 DermNet

DermNet is another widely used online resource with a large collection of dermatoscopic images and associated clinical information. The DermNet dataset is not well structured like HAM10000 or PH2 as it does not have separate metadata. The diagnosis of the particular dermatoscopic image is mentioned as the file name. DermNet is usually used as a supplementary dataset for augmenting datasets in skin cancer research [19].

These datasets provide a rich source of dermatoscopic images and metadata, offering a solid foundation for training and evaluating deep learning models in this field.

2.5 Data Augmentation

Data Augmentation technique is widely used in ML and deep learning to increase the size and diversity of the dataset by either creating new data from existing data or completely generating new data synthetically. The data augmentation is very essential when there is insufficient data which might lead to underfitting [27].

There are two commonly used data augmentation techniques. They are as follows:

2.5.1 Transformation-Based Augmentation

Transformation-Based Augmentation involves applying various transformation techniques on existing data to generate new data. The transformations are applied in such a way that the core characteristics of the data is maintained while introducing variations.

Some of the common transformation techniques for image data are as follows [27]:

(a) **Flipping**: Creating mirror images of the original image either vertically or horizontally.

(b) **Rotation**: Rotating the original image by a certain angle to simulate new variations.

(c) **Scaling and Zooming**: Changing the scale of the image by zooming in or out, adjusting the overall image size.

(d) **Cropping**: Selecting a subset of the original image.

(e) **Color Space**: Adjusting the brightness, contrast, and saturation of the original image.

(f) **Noise Injection**: Adding random noise or blur to the original image to simulate imperfections.

2.5.2 Synthetic Data Generation

Synthetic Data Augmentation takes data augmentation by a further step by creating new data using advanced techniques like Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs) and blending techniques. This technique is usually used when the dataset is very small and lacking diversity. Although this type of data augmentation is advanced and innovative, it is computationally difficult to achieve and is recommended only to be used when acquiring new data directly is not possible [34].

2.6 LabelEncoder

`LabelEncoder` is a python class present in scikit-learn (sklearn) library which converts categorical labels into numerical values. This involves assigning numerical values to the unique class labels [**?**]. ML models often can't process categorical labels directly, hence we use LaberEncoder to convert the categorical labels into numerical values preserving the mapping original categories and their corresponding numerical values [9].

2.7 Train_Test_Split() Function

The `train_test_split()` function is part of the Scikit-learn library and is used to divide the data into training and testing before feeding it to ML models. It takes a dataset, the desired proportion of testing data, and a random seed as inputs to produce the training and testing subsets [33].

The general structure of `train_test_split()` is as follows:

```
train_test_split(x, y, test_size=0.2, random_state=42)
```

where:

(a) **x** contains the input features of the dataset. This can be an array or matrix of data, with each row representing an individual data point and each column representing a different attribute or feature.

(b) **y** represents the target values or labels corresponding to the input features in **x**. These are the outputs the model is intended to predict.

(c) **test_size** specifies the proportion of the data to be used as a testing set. In this example, `0.2` means that 20% of the data is set aside for testing, while the remaining 80% is used for training. The `train_test_split()` function ensures that the unseen data is chosen as a test set.

(d) **random_state** controls how the data is split. It is a seed for the random number generator. When you use the same `random_state`, the data is split in the same way every time, allowing consistent results.

The `train_test_split()` function returns:

- **x_train**: Training subset of input features **x**.
- **x_test**: Testing subset of input features **x**.
- **y_train**: Training subset of target labels **y**.
- **y_test**: Testing subset of target labels **y**.

2.8 Evaluation Metrics

The performance and effectiveness of ML and deep learning models are measured using evaluation metrics [3]. Some of the most commonly used evaluation metrics for classification tasks are as follows:

2.8.1 Confusion Matrix

Confusion matrix is a widely used performance metric, especially in classification tasks involving multiple class labels. It summarizes the performance of a classification model by showing the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) [3].

A representation of the confusion matrix is as follows:

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

Confusion matrix can be used to calculate other performance metrics like accuracy, F1 score, precision, and recall. The components of the confusion matrix are:

(a) **TP (True Positive)**: The number of positive instances that are correctly predicted as positive by the model.

(b) **FP (False Positive)**: The number of negative instances that are incorrectly predicted as positive by the model.

(c) **FN (False Negative)**: The number of positive instances that are incorrectly predicted as negative by the model.

(d) **TN (True Negative)**: The number of negative instances that are correctly predicted as negative by the model.

2.8.2 Precision

Precision, also known as positive predictive value, quantifies the accuracy of positive predictions. It is defined as the ratio of true positives (TP) to the total number of predicted positives (TP + FP) [3].

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

2.8.3 Recall

Recall, also known as true positive rate, measures the proportion of true positives among all actual positives. It is defined as the ratio of true positives (TP) to the total number of actual positives (TP + FN) [3].

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

2.8.4 Accuracy

Accuracy is a common metric to evaluate the overall correctness of the model's predictions. It is calculated as the ratio of the sum of true positives and true negatives (TP + TN) to the total number of predictions (TP + TN + FP + FN) [3].

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

2.8.5 F1 Score

The F1 score is the harmonic mean of precision and recall. It is used to assess a model's performance when both false positives and false negatives are important [3].

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

2.8.6 Hamming Loss

Hamming loss is the proportion of incorrectly predicted labels with the true labels. The Hamming Loss for a multi-label classification task is calculated as follows [3]:

$$\text{Hamming Loss} = \frac{1}{N \times L} \sum_{i=1}^{N} \sum_{j=1}^{L} \mathbf{1}(y_{i,j} \neq \hat{y}_{i,j})$$

where:

- N is the number of samples,
- L is the number of labels for each sample,
- $y_{i,j}$ is the true label for the ith sample and the jth label,
- $\hat{y}_{i,j}$ is the predicted label for the same sample and label,
- $\mathbf{1}$ is an indicator function that returns 1 if the true and predicted labels don't match, and 0 if they do.

Hamming Loss value of 0 indicates perfect predictions, while a value of 1 signifies that all predictions are incorrect.

2.9 Libraries

2.9.1 Numpy

The Numpy python library provides powerful multidimensional arrays and matrices along with many useful mathematical functions to use on these arrays. Numpy arrays are more flexible to use than the default python lists, allowing efficient data manipulation [8]

2.9.2 Pandas

Pandas is a very powerful python library commonly used for data analysis and manipulation. Pandas provide various new data structures and data analysis tools. It is an extension of the Numpy library and is known to handle large amounts of data operations [40].

2.9.3 Matplotlib

Matplotlib is a powerful open source python library which provides tools to visualize with charts, graphs, and many more interactive visualizations. Matplotlib is also based on Numpy library and has a wide variety of plots to make visualization easier [4].

2.9.4 Scikit-Learn

Scikit-Learn is a python library for ML, providing a wide range of efficient tools for tasks such as classification, regression, clustering, and dimensionality reduction. It is built on SciPy, Numpy, Matplotlib libraries [7].

2.9.5 Keras

Keras is a very powerful python library for developing and evaluating deep learning models. It allows the programmers to experiment and create deep learning architectures with ease helping rapid prototyping. It runs on CPU as well as GPU providing distributed operation to reduce complexity on CPU [6].

2.9.6 Seaborn

Seaborn is a powerful data visualization library which is built on Matplotlib providing very informative and creative graphics like heat maps, scatter plots, etc. It is widely used in the data science community for its ability to create publication-quality figures with minimal code [42].

2.9.7 PIL

The Python Imaging Library (PIL) is a foundation python library for image processing and manipulation which is used to handle python image processing tasks like opening, saving, transforming images etc.

Chapter 3

Related Work

Previous research work in this field by Raza et al. explores the effects of acral lentiginious melanoma on people with African, Asian and Hispanic ancestries [35]. It delves into analysis of ensemble model performance of convolutional models (VGG16, Xception, InceptionResnetV2, DenseNet121, DenseNet169, and DenseNet210) based on transfer learning using a stacking approach. The performance of the proposed method was evaluated on a Figshare benchmark dataset and the impact of different augmentation techniques was analyzed. These ensemble results have shown higher performance than other models at 97.93 percent accuracy.

Alwakid et al. focused on image enhancement using ERSGAN(Enhanced Super-Resolution Generative Adversarial Networks) then analyzed prediction patterns on seven kinds of melanoma in the HAM10000 dataset [16]. These researchers have explored various deep learning techniques, such as feature fusion and selection frameworks, for multiclass skin lesion localization and classification. This approach extensively analyses the different aspects of the HAM10000 dataset where ROI (Regions of Interest) are localised from original image and used Resnet-50 to obtain 0.86 accuracy, 0.86 F1-score. 0.86 precision and 0.84 recall.

Kushimo et al. investigated a pre-trained DenseNet121 which served as the foundation for training [25]. Through transfer learning, the top layer was excluded, and fine-tuning was applied to all layers in the final Dense Block pre-trained on ImageNet. The model achieved 99 percentage accuracy in detecting melanoma in white skin and 98 percentage in dark skin. These results affirm the efficacy of the proposed model for melanoma detection in diverse skin tones.

Hossain et al. combined 10 deep learning models using max voting ensemble technique training on the ISIC dataset. This research is very close to this thesis topic as the ISIC dataset is used for training and generalization ablities are being validated using the HAM10000 dataset. Through transfer learning, This method involves utilizing the synergy between all these model architectures to effectively employ all of them together. The model achieved 93 percentage accuracy in detecting melanoma [21].

Wu et al. provided a review on the different deep learning frameworks employed in skin cancer detection. It explores several challenges such as the effects of data imbalance on model generalization ability. The review discusses the advantages and disadvantages of using the different deep-learning models. It focuses on the shortcomings of the algorithm towards becoming a complete diagnostic system. The experimental results revealed that the proposed deep learning methods achieved high

accuracy, with a dice similarity coefficient of 0.954 and an IoU index of 0.914 on the PH2 dataset [45].

3.1 Summary of previous research:

Authors	**Model used**	**Results obtained**
Raza et al	Xception	Accuracy is 95.17%
	Inception-ResNet-V2	Accuracy is 95.17%
	DenseNet121	Accuracy is 94.48%
	DenseNet201	Accuracy is 95.86%
	Stacked Ensemble Model	Accuracy is 97.93%
Alwakid et al	ResNet	Accuracy is 77%
	CNN	Accuracy is 78%
Kushimo et al	Custom DenseNet Model	Accuracy is 99%(White skin samples) & Accuracy is 98(Dark skin samples)%
Hossain et al	MobileNet	Accuracy is 77.20%
	AlexNet	Accuracy is 80.10%
	vgg16	Accuracy is 83.80%
	ResNet150	Accuracy is 86.05%
	DenseNet121	Accuracy is 88.30%
	DenseNet201	Accuracy is 88.80%
	InceptionV3	Accuracy is 89.50%
	ResNet50V2	Accuracy is 89.30%
	InceptionResNetV2	Accuracy is 90.20%
	Xception	Accuracy is 91.90%
Wu et al	MobileNet	Accuracy is 94.4%
	ResNet	Accuracy is 62.2%
	Weight Pruning Approach	Accuracy is 97.5%
	MergePrune Approach	Accuracy is 77.6%
	EfficientNet	Accuracy is 87.3%
	MobileNet and NasaNet	Accuracy is 82.0%
	MT-TransUNet	Accuracy is 91.2%
	SqueezeNet	Accuracy is 97.2%

As shown, there is extensive research work done in the field of skin cancer employing deep learning models, CNNs and ERSGAN in one instance. Few papers have exclusively dedicated work towards the detection of melanoma and its varieties in people of color. This gap is addressed by applying deep learning models on localised images exclusively on dark skin samples. Most of the papers have used the HAM10000 dataset as a benchmark dataset for its wide range and versatile base in several other skin conditions.

Research Gap

In this thesis, the reliability and analysis of the HAM10000 dataset is under focus. The contribution of this dataset when the models are modified for investigation un-

der several iterations in an experimental method. A cross dataset generalization approach is taken where the generalizing ability of HAM10000 dataset is tested against other popular datasets like PH2 and DermNet. This is a unique approach in the current field of research.

Chapter 4

Method

In this thesis, an experimental method has been undertaken where CNN models have been employed to train iteratively on 80%, 90% and 100% of the HAM10000 dataset. The trained models are then validated using two other diverse datasets in skin cancer domain, PH2 and Dermnet. An ensemble of the trained models is also analysed to evaluate its performance when compared to the individual neural models. Each model is evaluated using metrics like precision, recall, F1-score and Accuracy. The following Figure 4.1 is a representation of the experimental method undertaken in this thesis:

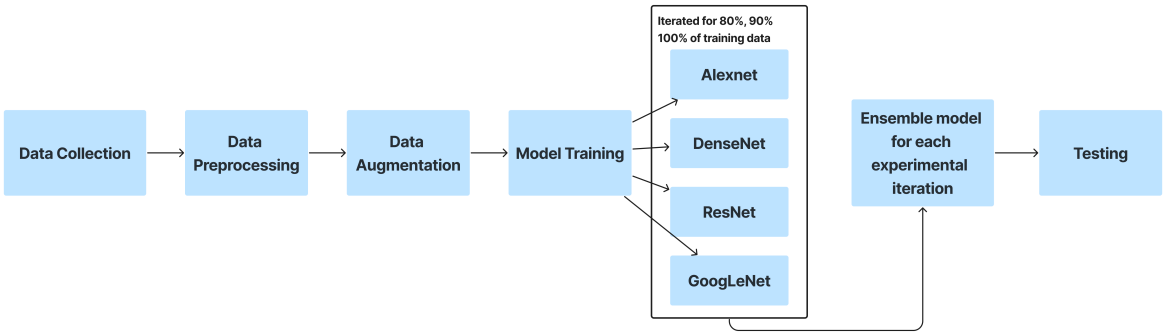


Figure 4.1: Chronological Flow of Experimental Methodology

4.1 Exploratory Data Analysis

4.1.1 Data Collection and Metadata Analysis

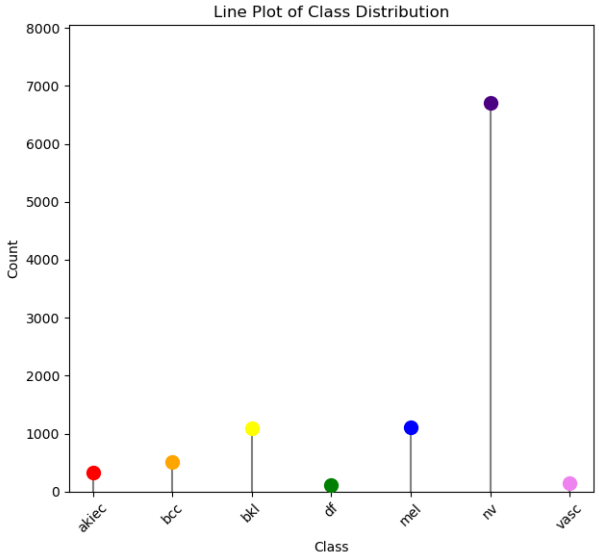
HAM10000 Dataset:

The HAM10000 dataset is publicly available dataset widely used for its diverse image base in various skin conditions including malignant melanoma. It consists of 10,015 dermatoscopic images of pigmented skin lesions collected from multiple regions, including melanoma, melanocytic nevi (moles), basal cell carcinoma, seborrheic keratosis, vascular lesions and other types of skin lesions. All labels are verified using hispathology results. The data is provided as two folders, namely training and testing. Since this thesis required custom split ratios, the whole dataset was merged and

trained [24]. The following is the list of attributes present within the HAM10000 dataset:

- Image ID: A unique identifier for each image.
- Lesion ID: A unique identifier for each lesion.
- Image: The actual image of the skin lesion captured through dermatoscopy.
- dx: Diagnosis of the lesion. It includes several categories such as melanoma, melanocytic nevus, basal cell carcinoma, etc.
- dx_type: The method of diagnosis, whether it's histopathology, clinical, follow-up, or consensus.
- Age: The age of the patient when the image was taken.
- Sex: The gender of the patient.
- Localization: The anatomical site of the lesion on the body.
- Moles: The number of moles in the image.
- Malignant: A binary attribute indicating whether the lesion is malignant or benign.
- Benign: A binary attribute indicating whether the lesion is benign or not.

  The frequency of images in each class is represented in the following Figure 4.2.



Class	Count
akiec	327
bcc	514
bkl	1099
df	115
mel	1113
nv	6705
vasc	142

Figure 4.2: Class Distribution in HAM10000 dataset

The following image 4.3 shows a sample of image data from the 7 classes under consideration in this thesis.

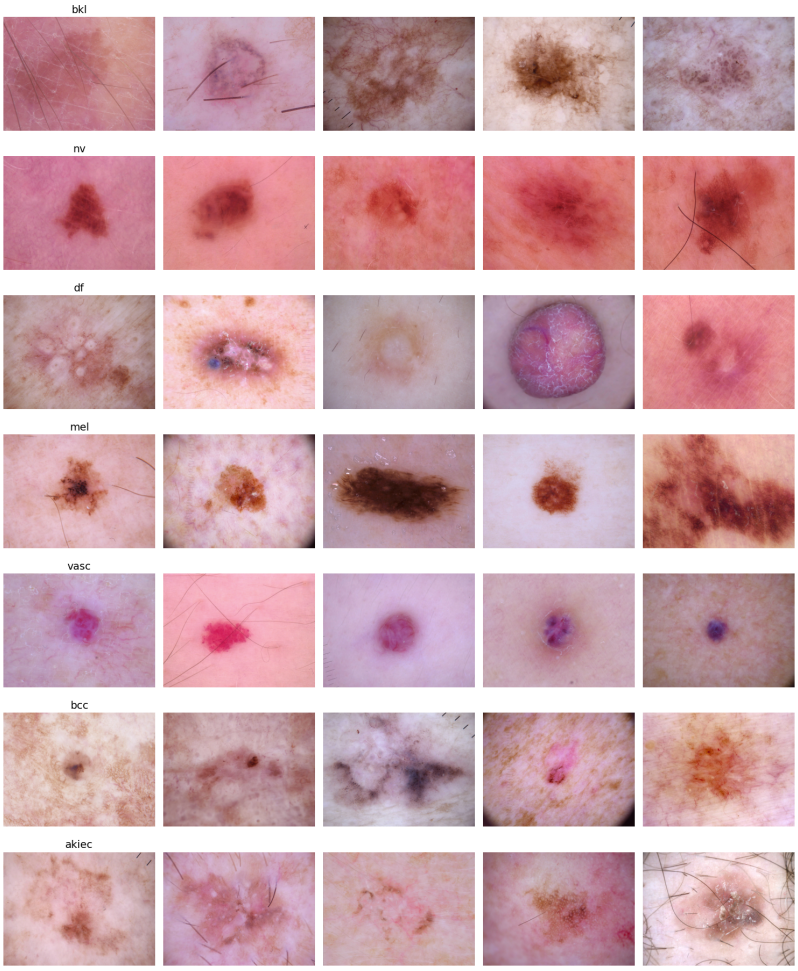


Figure 4.3: Data samples from each class in HAM10000 dataset

The following illustration 4.4 represents the frequency of melanoma cases according to age and gender. Some data gaps can be identified when data header is analysed. As dermatoscopic images alone do not show the whole clinical profile, additional data has been provided for each image sample. The patterns in probability of cancerous growth is important for the model to learn to make better informed predictions.

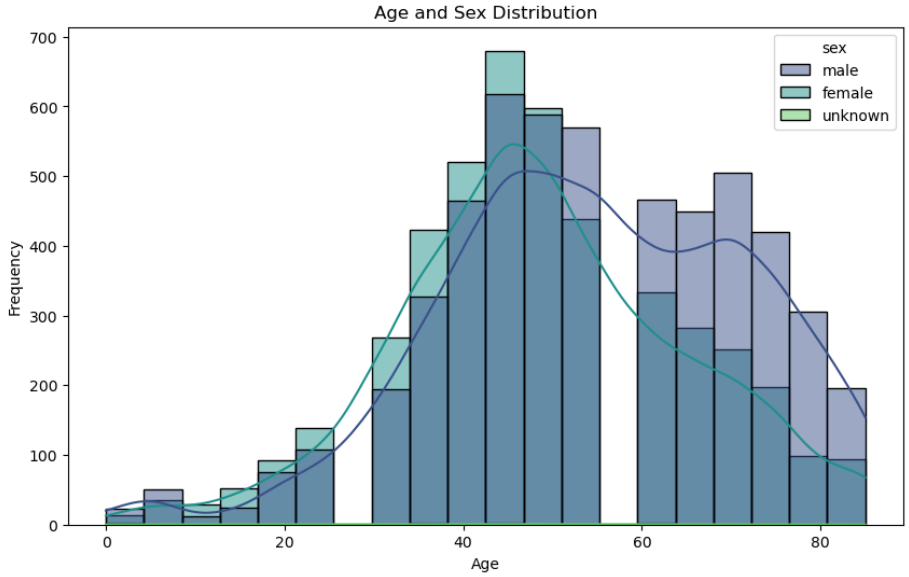


Figure 4.4: Age and Gender distribution in HAM10000 dataset

Each dermatoscopic image as shown in above illustration 4.3 differs according to the location of ailment, relevantly called the Region of Interest(ROI). Lesion location plays a major role in accurately predicting whether a lesion is malignant or not. As evident from representation below 4.5 most of the samples here are concentrated in the back region, so this will help the model to balance any bias while making predictions.

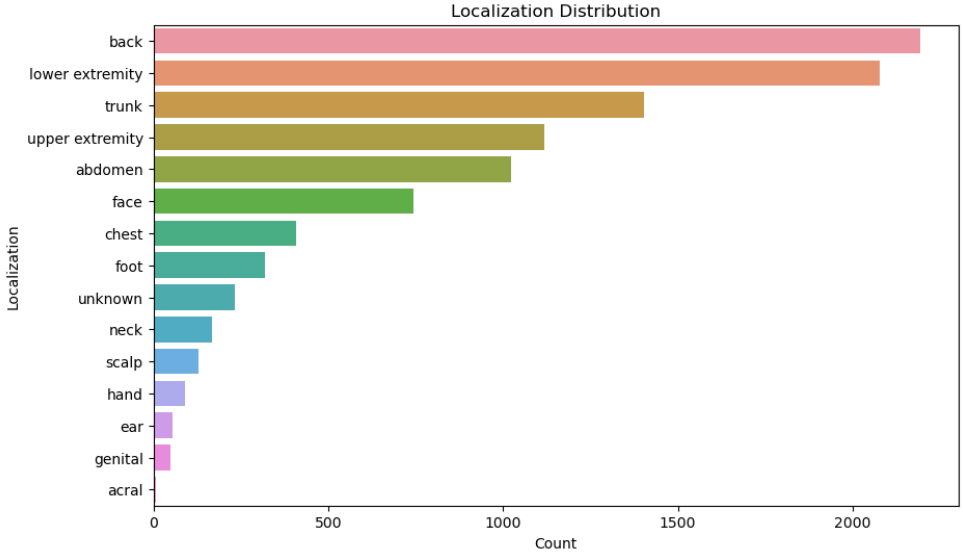


Figure 4.5: Lesion location distribution in HAM10000 dataset

Dermnet Dataset:

The original Dermnet dataset is a collection of more than 20,000 images dispersed among 23 broad labels out of which only 7 have been extracted for this thesis. Hence the modified dermnent dataset has 1903 images. It is a collection of dermatological images with both segmentation and classification labels, making it a valuable resource for evaluating deep learning models. The labels extracted are as follows [19]:

- akiec: Actinic Keratoses
- bcc: Basal Cell Carcinoma
- bkl:Benign keratoses like lesions
- df: Dermatofibroma
- nv: Melanocytic nevi
- vasc: Vascular lesions
- mel: Melanoma

As discussed earlier, the original Dermnet dataset distribution is shown in 4.6. It has a wide range of labels that can be utilized for several ML applications.

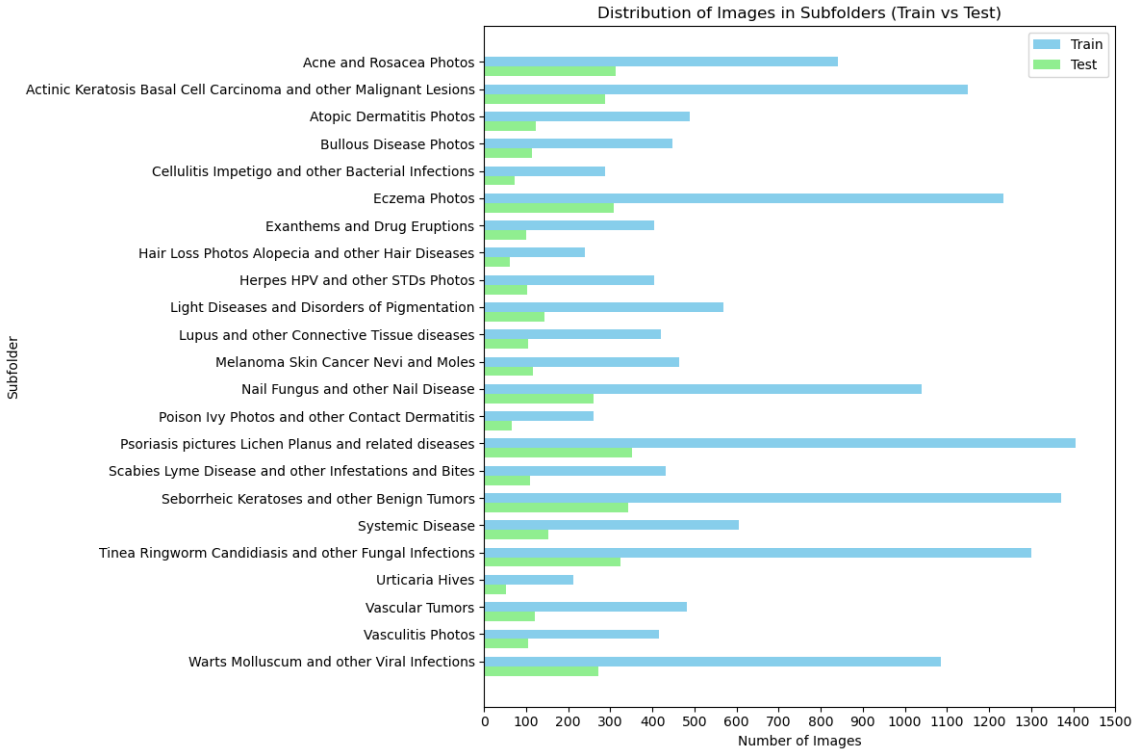


Figure 4.6: Original DermNet class distribution

The following data distribution illustration 4.7 shows the modified classes that were extracted from the original data pool. The common labels from HAM10000 and Dermnet are exclusively under consideration as test data along with dermoscopic data from PH2 dataset.

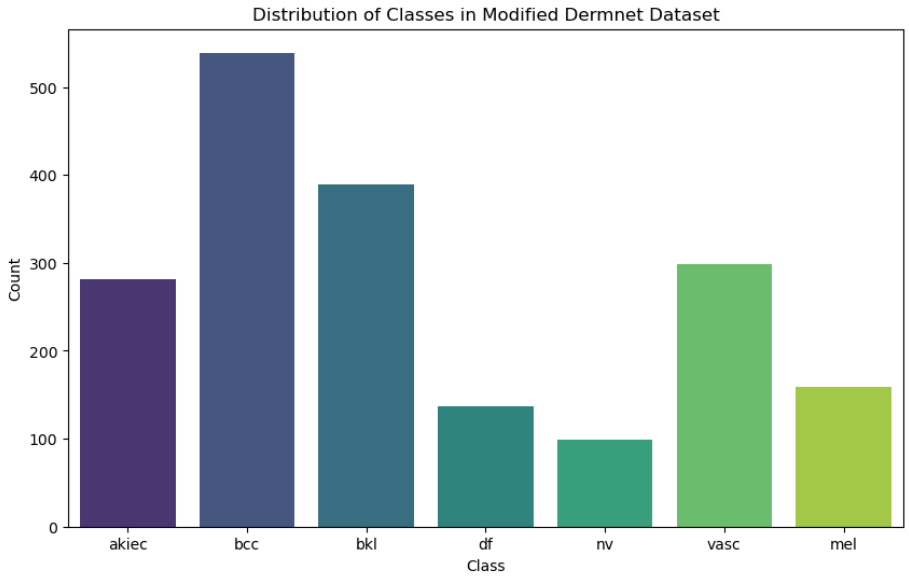


Figure 4.7: Modified DermNet class distribution

PH2 Dataset:

The PH2 dataset contains 200 dermoscopic images of melanocytic lesions. All images are labeled with pixel-level annotations. There are no pre-defined train-test splits in the dataset. It consists of two main labels, common nevi and malignant melanoma lesions [28].
The Figure 4.8 shows the three class labels present in PH2 Dataset, It has a higher number of sample for common nevus and atypical nevus class, hence it is required to normalize the labels according to the larger data sources being utilized in this thesis.

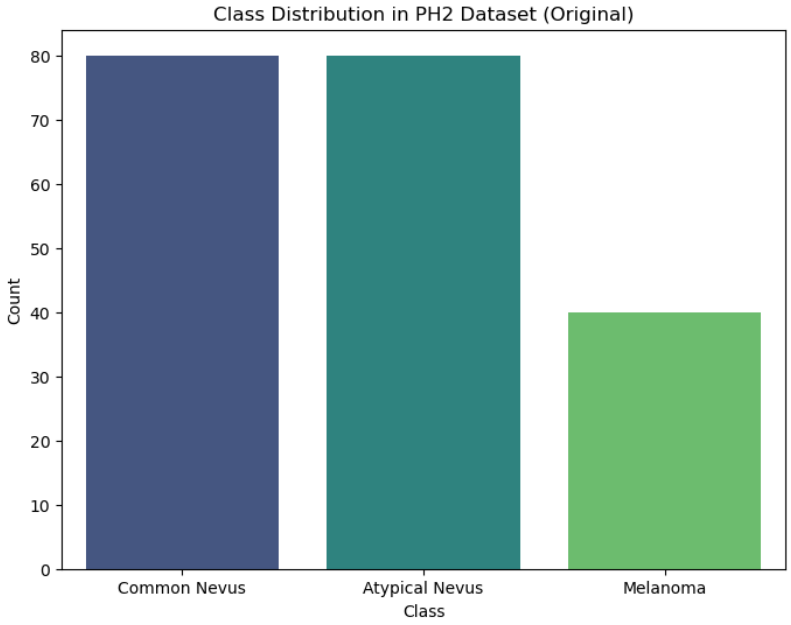


Figure 4.8: Original PH2 dataset class distribution

Now in Figure 4.9, only common nevus and melanoma image samples have been extracted from original dataset. These are to be merged with Dermnet samples as test data for Alexnet, GoogleNet, ResNet and DenseNet models already trained on the HAM10000 dataset as discussed previously.

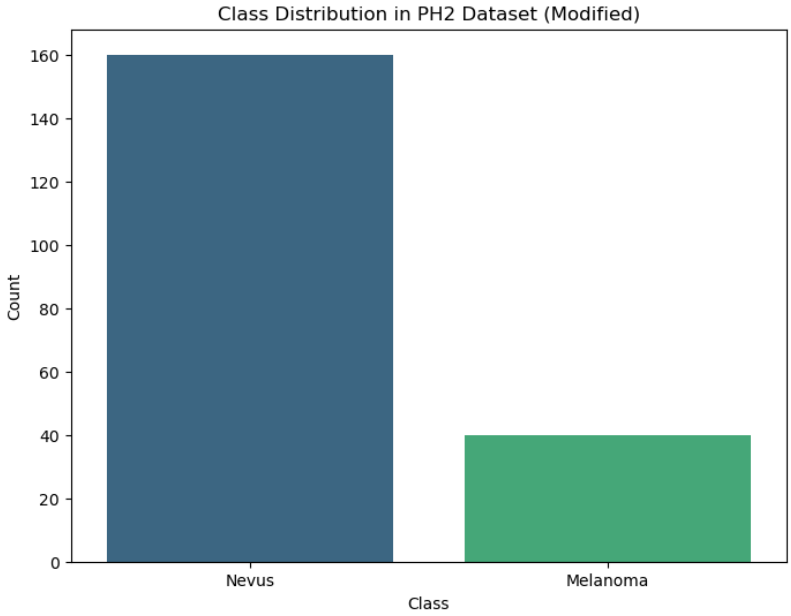


Figure 4.9: Modified PH2 dataset class distribution

4.2 Data Cleaning and Preprocessing

4.2.1 Handling Missing values

Dealing with missing values in the dataset is essential to prevent inefficient training process. Once the missing values were identified, appropriate imputation techniques were applied. Numerical columns were imputed with the median. This ensures consistency in data distribution and minimizes bias [27].

4.2.2 Resizing Images

It is essential to ensure that the data is uniform and homogeneous in all aspects so that the model can decipher any and all features properly without having to account for new complexities like correcting resolution bias in the image data. Here in the Figure 4.10 it is clear that all the data points are resized to size 75x75 while loading the data from CSV file itself [27].

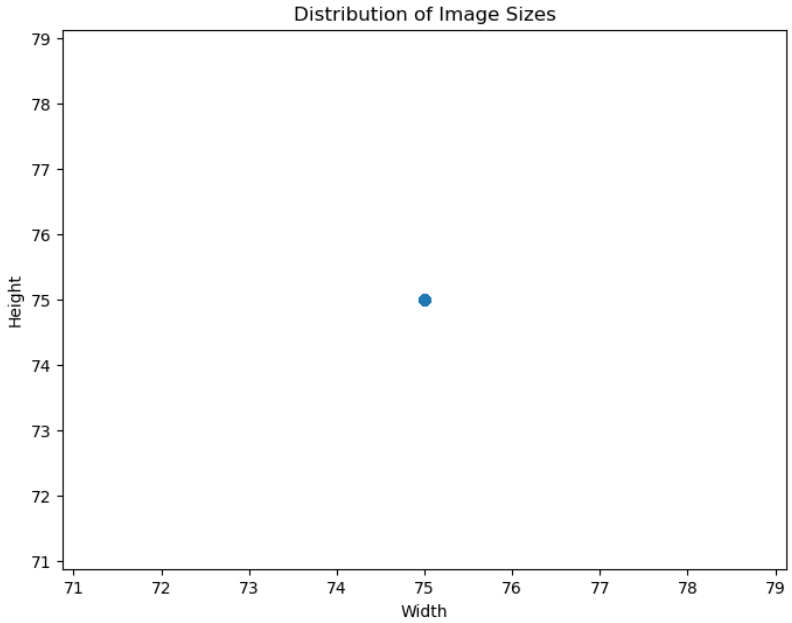


Figure 4.10: HAM10000 image size after resizing

4.2.3 Data Normalization

Data Normalization method is used to make sure that all the pixel values are on a consistent scale [14]. This consistency in training process allows the model to learn without being biased. There is a need to mitigate this bias, as in this thesis multiple dataset sources are being utilised. Here Min-Max Scaling method is used, where usually the data is normalized to a scale between 0 and 1. Pixel values are normalized by dividing the values by a constant. 255 is the maximum value for an 8-bit image, hence all data points were divided by 255 as shown below.

```
img = np.array(img) / 255.0
```

4.2.4 LabelEncoder

Label encoding is a technique used to convert categorical data into numerical format. This ensures that the labels are transformed into machine understandable format. Each category is mapped to a unique numerical value, hence ensuring consistent encoding across the dataset. This contributes to a homogeneous dataset ready for the model to train on efficiently [9].

```
from sklearn.preprocessing import LabelEncoder}
le = LabelEncoder()
le.fit(skin_df['dx'])
skin_df['label'] = le.transform(skin_df['dx'])
```

4.3 Data Augmentation

Data Augmentation is a technique used to artificially generate new varied data from existing data source. It is mainly used to prevent overfitting in data that is fed into deep learning models like AlexNet, DenseNet, ResNet and GoogleNet. It is a regularisation technique that can be used to solve for class imbalance concerns when large datasets have irregular data distribution for each class. Usually random swapping, deletion, insertion, and synonym replacement are examples of straightforward data augmentation [27]. Although to account for the medical dataset here, Instead of using trivial techniques, the classes that are lacking significantly have been reinforced with the ISIC-2018 4.15 image samples in that particular class label [5].

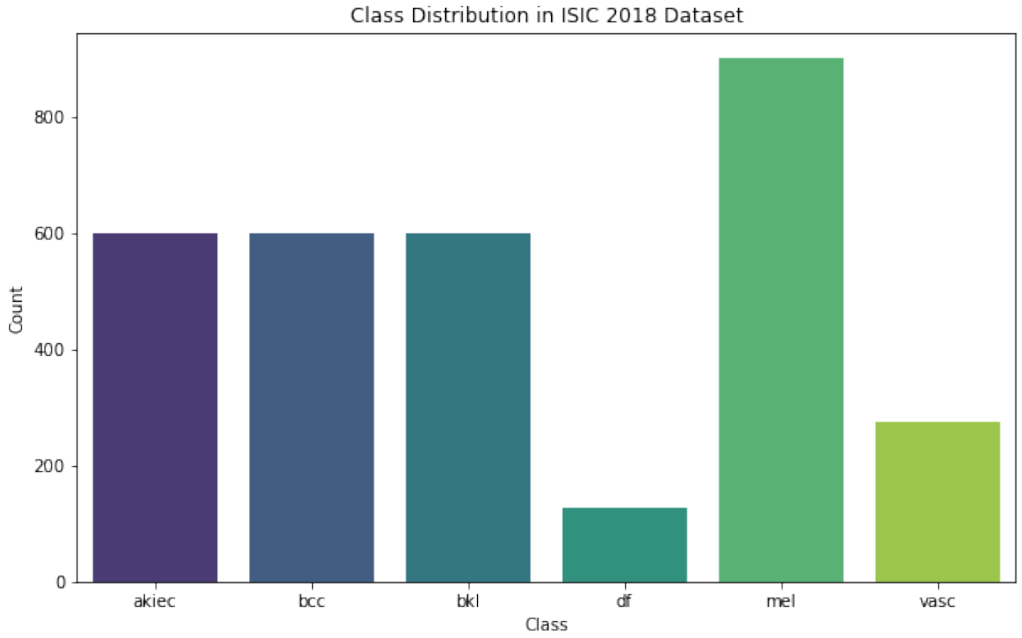


Figure 4.11: Class distribution of ISIC 2018 dataset

In this project, data augmentation was applied to ensure that each class had at least 600 images for robust model training. The following subsections explain the steps taken to achieve class balance and the methods used to determine optimal transformation parameters.

4.3.1 Reinforcement with ISIC 2018 Dataset

The initial class distribution in the HAM10000 dataset was as follows:
- bkl: 1,099 images
- nv: 6,705 images
- df: 115 images
- mel: 1,113 images
- vasc: 142 images
- bcc: 514 images
- akiec: 327 images

To ensure a balanced distribution, each class sample size was limited to 600 images.

This meant reducing the number of images for over represented classes and reinforcing underrepresented classes. In order to reinforce the under represented classes with relevant data , the ISIC 2018 Dataset samples were imported into the modified HAM10000 dataset with classes under 600 images.

Classes with more than 600 images (e.g., ‘nv‘, ‘bkl‘, ‘mel‘) were randomly reduced to 600 images. Classes with fewer than 600 images (e.g., ‘akiec‘, ‘bcc‘, ‘df‘, ‘vasc‘) were reinforced with images from ISIC 2018 to reach the target count of 600.
Two classes still had fewer than 600 images: ‘df‘ and ‘vasc‘. To address this, trivial data augmentation techniques were used to reach the target count.

4.3.2 Finding Optimal Parameters

To augment classes with fewer than 600 images, the best transformation parameters to maximize data pool without introducing significant noise or distortion are to be determined. Rather than applying transformations randomly, Bayesian optimization technique is used. It is a method to find optimal values for hyperparameters.The optimal transformation parameters identified through Bayesian optimization were [2]:
- Rotation range: 13 degrees
- Zoom range: 0.433870
- Width shift range: 0.467662
- Height shift range: 0.355253
- Shear range: 0.995326
- Rescale: 0.997017
With this approach the two under represented classes, ‘df‘ and ‘vasc‘ reached the required count of 600 images.

4.4 Model Architecture and Training

4.4.1 Model Selection

Four prominent CNN architectures—AlexNet, GoogLeNet, DenseNet, and ResNet were selected to evaluate their performance in cross-dataset generalization for melanoma prediction. These models have been commonly used in image classification tasks in various applications. Their ability to capture complex visual patterns makes them suitable for medical image analysis.
Rationale for Model Selection:

- **AlexNet:** Its architecture, comprising multiple convolutional layers, pooling layers, and fully connected layers, is designed to learn robust features from complex data, making it an ideal benchmark [17].

- **GoogleNet:** GoogleNet Architecture allows the extraction of features at multiple scales simultaneously. This approach is advantageous for capturing both local and global information from images, providing flexibility and depth [38].

- **DenseNet:** With dense connections between layers, DenseNet promotes feature reuse and efficient information flow. This leads to improved performance and reduced model complexity [22].

- **ResNet:** ResNet Architecture allows for residual connections that address the issue of vanishing gradients in deep networks. ResNet model carries out more stable training and better performance with deep architectures [30].

According to the systematic review "Skin Cancer Classification With Deep Learning: A Systematic Review" published in Frontiers in Oncology, these architectures are among the most commonly used and effective CNN models for skin cancer classification. So conducting a comparative study with these architectures is to understand their potential in cross-dataset generalization for melanoma prediction [45].

4.4.2 Defining Architecture

In this section, the architectures of the four CNN models used in our study are outlined: AlexNet, GoogLeNet (InceptionV3), DenseNet121, and ResNet50.

AlexNet:

It features an eight-layer deep structure comprising the following elements [26]:

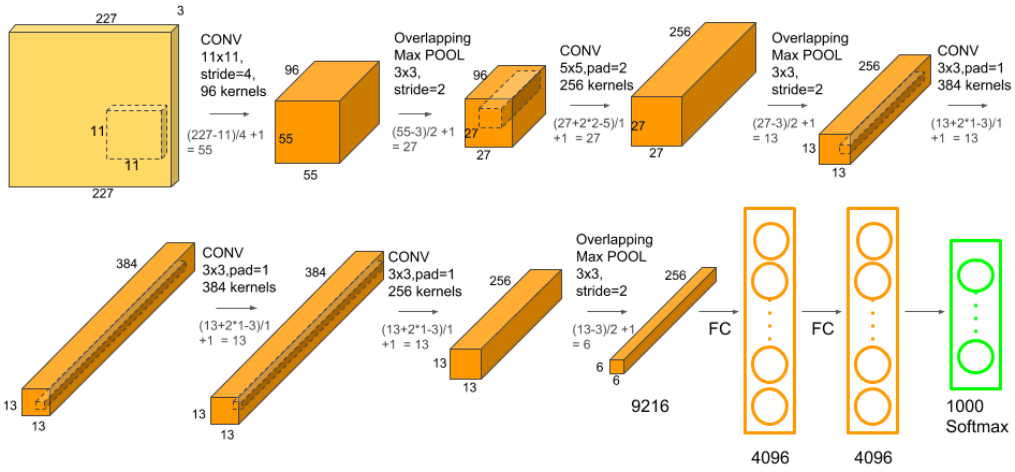


Figure 4.12: AlexNet Architecture [17]

- **Five Convolutional Layers:** The base of AlexNet is its five convolutional layers, designed to capture different features from the input images. The first convolutional layer uses an 11x11 kernel with a stride of 4, allowing rapid downsampling, while the subsequent layers use smaller kernels (3x3, 5x5), enabling more refined feature extraction.

- **Three Fully Connected Layers:** These layers, with the first two containing 4096 neurons each, transform spatial features into a one-dimensional representation suitable for classification. This high-capacity structure enables AlexNet to learn complex patterns in the data.

- **Dropout Layers:** To reduce overfitting, dropout layers with a dropout rate of 0.5 are inserted after the fully connected layers. Dropout randomly deactivates neurons during training, introducing regularization and improving model robustness.

- **Softmax Output Layer:** The final layer uses the Softmax function to generate probabilities for each class, facilitating multi-class classification tasks such as melanoma prediction.

GoogleNet(InceptionV3):

GoogleNet allows the network to extract features at multiple scales. This multi-scale approach contributes to the architecture's depth and flexibility [38].

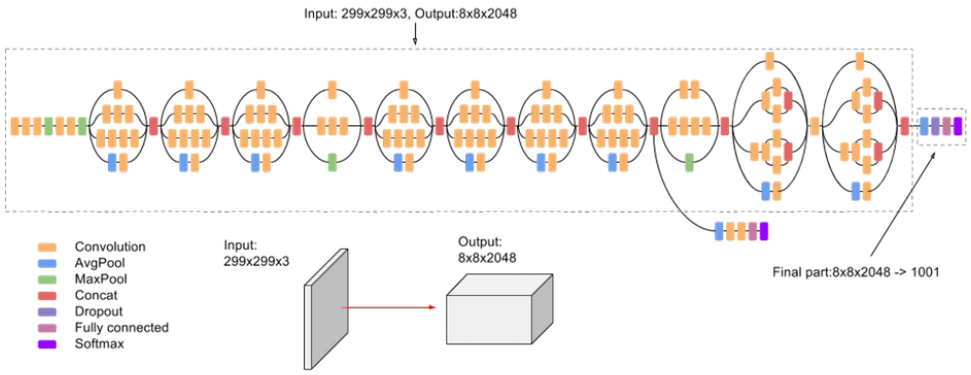


Figure 4.13: GoogLeNet Architecture [1]

- **Inception Modules:** Each module contains a combination of convolutional operations with various kernel sizes (1x1, 3x3, 5x5), along with pooling layers. This design helps capture both local and global information, promoting better generalization.

- **Global Average Pooling:** GoogLeNet employs global average pooling instead of fully connected layers, reducing the model's complexity while preserving essential features for classification.

- **Dense Layer:** After global average pooling, a dense layer with 1024 neurons further processes the features before passing them to the Softmax output layer for classification.

- **Softmax Output Layer:** The Softmax layer generates class probabilities, allowing the network to categorize input images into different classes.

DenseNet121:

DenseNet121 is characterized by dense connectivity between layers, encouraging feature reuse and efficient information flow. This design reduces model complexity while maintaining high performance [22].

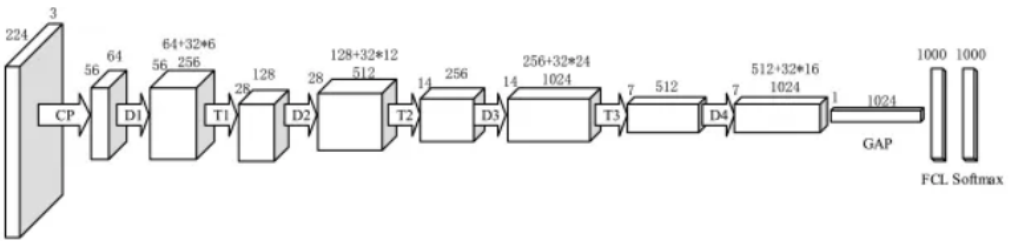


Figure 4.14: DenseNet Architecture [11]

- **Dense Blocks:** Each dense block consists of layers connected to all preceding layers, promoting feature reuse and enhancing learning.
- **Transition Layers:** These layers, positioned between dense blocks, reduce the feature map size through pooling, thereby maintaining efficiency and reducing overfitting.
- **Global Average Pooling:** Similar to GoogLeNet, DenseNet uses global average pooling to reduce spatial dimensions while retaining critical information.
- **Softmax Output Layer:** This final layer applies the Softmax function to classify the input into one of several categories, facilitating multi-class prediction.

ResNet50:

ResNet50 is distinguished by its use of residual connections, addressing the vanishing gradient problem in deep networks. This approach allows for deeper networks while maintaining training stability [30].

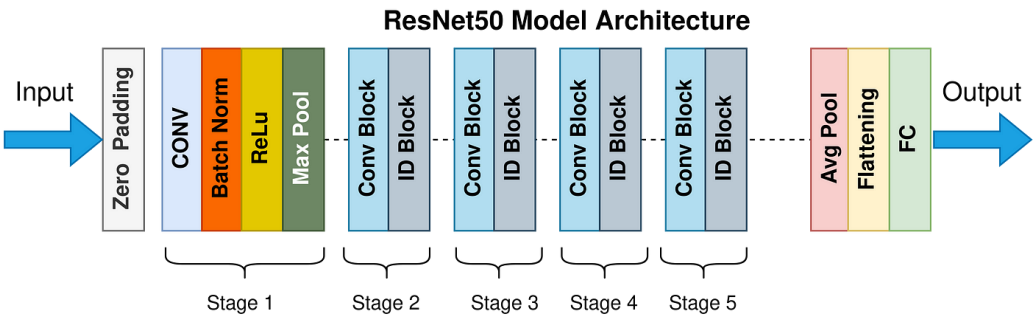


Figure 4.15: ResNet50 Architecture [29]

- **Residual Blocks:** Residual connections, also known as skip connections, bypass certain layers, allowing the network to learn residual functions. This structure supports deeper networks without compromising training stability.
- **Global Average Pooling:** This layer reduces model complexity, promoting computational efficiency without trading off accuracy.
- **Softmax Output Layer:** The Softmax output layer generates probabilities for each class, enabling effective multi-class prediction for melanoma diagnosis.

4.4.3 Model Compilation

Model compilation is the process of setting the key configurations for a deep learning model, including the optimizer, loss function, and metrics. This step defines how the model will be trained and evaluated. Proper compilation ensures efficient training, stability, and reliable performance [15]. The model compilation process undertaken in this project is as follows:

- **Optimizer:** The optimizer determines how the model's weights are updated during training. In this model, the Adam optimizer was used for its adaptive learning rate and robustness. It combines the benefits of both momentum and adaptive learning rate methods, providing effective convergence [39]. The key parameter for Adam is the learning rate, which was set at 0.0001. This value balances the speed of convergence with stability to avoid large updates that could lead to oscillations or instability.
- **Loss Function:** The loss function quantifies the difference between the model's predicted labels and actual labels. For this multi-class classification task, 'sparse_categorical_crossentropy' function was utilised. This loss function is ideal when the target labels are integers, as it computes the cross-entropy loss between the predicted probabilities and the actual class labels. Sparse categorical cross entropy is preferred for its simplicity and efficiency in multi-class settings [39].
- **Metrics:** Metrics are used to evaluate the model's performance during training and validation. Accuracy is set as the primary metric. Accuracy simply measures the proportion of correct predictions to the incorrect predictions. This metric provides a clear indication of the model's effectiveness in predicting the correct class [15].

```
from keras.optimizers import Adam
model.compile(
optimizer=Adam(learning_rate=0.0001),
loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
```

The above code snippet represents the compilation process for the four CNN models used in this study. This configuration defines how the models will be trained and how their performance will be evaluated. By standardizing the optimizer, loss function, and metrics, consistency is ensured across all models,

4.4.4 Model Training

Model training is the process of iteratively refining a deep learning model's parameters to optimize its performance. This stage involves feeding the model with training data, adjusting its weights, and monitoring performance metrics [10]. This section describes the training process for the four convolutional neural network: AlexNet, GoogLeNet (InceptionV3), DenseNet121, and ResNet50. It covers key considerations, including the train-test split, early stopping, and training parameters.

Core training parameters:

- **Batch Size:** This determines the number of samples processed in each iteration of training. Smaller batch sizes can improve generalization, while larger batch sizes can speed up training [32]. A batch size of 16 was selected to balance efficiency and stability.

- **Epochs:** An epoch represents a full pass through the training dataset [32]. We trained each model for 50 epochs, providing ample opportunity for learning while monitoring for overfitting.

- **Validation split:** A percentage of the training dataset used to validate the model's performance during training, helping detect overfitting. In this study, a 10% validation split was utilized. But, for the models of 100% training data group, there is no data left to validate, these models are simply saved ad then, validated and tested using the external testing dataset.

Train_Test_Split()

To evaluate model performance, the dataset is divided into training and testing sets using a specific ratio. In this study, different train-test splits ratios were used to assess their impact on training and generalization.

```
from sklearn.model_selection import train_test_split

x_train_80, x_test_20, y_train_80, y_test_20 = train_test_split(X, y,
test_size=0.2, random_state=42)
x_train_90, x_test_10, y_train_90, y_test_10 = train_test_split(X, y,
test_size=0.1, random_state=42)
```

The code snippet above illustrates how the 'train_test_split' function from Scikit-learn is used to divide the dataset into training and testing sets. The test size parameter determines the proportion of data reserved for testing.

Early Stopping

Early stopping is a technique to prevent overfitting by monitoring a specific metric during training and stopping if the metric does not improve for a certain number of epochs [13]. This approach helps maintain the model's generalization ability.

```
from keras.callbacks import EarlyStopping

# Early stopping callback
early_stopping = EarlyStopping(
    monitor='val_loss',
    patience=5,
    restore_best_weights=True
)
```

The code snippet above demonstrates how early stopping is configured. The 'patience' parameter controls how many epochs are allowed without improvement before stopping. The 'restore_best_weights' option ensures that the model reverts to its best state if early stopping is triggered.

To implement early stopping during model training, the callback functionality provided by Keras is used. The callback is added to the 'fit' method, allowing early stopping to be applied as part of the training process.

```
history_80 = model.fit(
    x_train_80, y_train_80,
    epochs=50,
    batch_size=16,
    validation_split=0.10,
    callbacks=[early_stopping],
    verbose=2)
```

This code snippet shows the early stopping technique applied during model training, with a validation split of 10%. The validation split reserves 10% of the training data for validation during training. This approach monitors of the model's progress and early detection of overfitting.

4.4.5 Model Saving

The primary reasons for saving models are ensemble learning, and metric extraction. Saving models after training allows for rapid reloading, enabling quick validation on test data. This approach improves efficiency when evaluating model performance, as it eliminates the need to retrain models for testing purposes. Saving individual model files facilitates ensemble learning by allowing multiple models to be combined for improved accuracy. This technique is useful when integrating predictions from different architectures, leading to enhanced model performance. Lastly, by saving model predictions, it is possible to analyze them to extract key performance metrics such as accuracy, precision, recall, and F1-score. This analysis provides insights into the model's effectiveness and guides further adjustments to optimize performance.

4.5 Ensemble Approach

Ensemble learning involves combining multiple models to improve prediction accuracy. AlexNet, GoogLeNet (InceptionV3), DenseNet121, and ResNet50 are trained using three different train-test split ratios: 80%, 90%, and 100%. This resulted in 12 trained models.

To create ensemble models, the trained models were grouped by their train-test split ratio, forming three ensembles:
- **Ensemble 80**: Combined the four models trained with 80% of the data.
- **Ensemble 90**: Combined the four models trained with 90% of the data.
- **Ensemble 100**: Combined the four models trained with 100% of the data.

Voting Classifier ensembling technique was used to merge the corresponding models. Soft voting aggregates the probabilities from the individual models, allowing for a more simple ensemble prediction [3]. By combining predictions from multiple models, the ensemble approach helps to reduce variance, improve overall accuracy, and increase the stability of predictions.

Ensemble learning is particularly useful in contexts where individual models may have unique strengths and weaknesses. By integrating multiple models, the ensemble approach provides a more reliable and accurate prediction framework for melanoma classification. This method also allows for easier scalability and flexibility in model deployment, providing a robust solution for complex classification tasks.

4.6 Model Testing and Evaluation

The trained models were evaluated using standard classification metrics, including precision, recall, F1-score and accuracy. These metrics provide insights into the models' performance in predicting melanoma classes that have multiple features [3].

Evaluation Metrics

- **Precision**: Measures the proportion of true positive predictions out of all positive predictions.
- **Recall**: Quantifies the model's ability to identify all relevant instances among the actual positives.
- **F1-score**: Harmonic mean of precision and recall, providing a balanced assessment of the model's performance.
- **Accuracy**: Indicates the overall correctness of the model's predictions.
- **Hamming Loss**: Measures the fraction of labels that are incorrectly predicted.

Evaluation Process

After training, each model was tested using a separate test dataset. This is a custom dataset merging required classes from PH2 and DermNet that are common in the HAM10000 dataset. The corresponding results signify the generalizing ability of the models as well as the quality of HAM10000 dataset. It will show if the dataset accounts for bias in medical data.

Chapter 5

Results and Analysis

After the models are tested on the custom dataset, each model's performance is assessed thoroughly. The confusion matrix, precision, recall, F1 score, accuracy and Hamming Loss are some standard metrics chosen for assessing the models. The models were trained with three different train-test split configurations which are 80%, 90%, and 100% training data, resulting in a total of 12 trained models. Additionally, these models were ensembled into three groups based on their training configurations, resulting in 3 ensemble models.

To maintain consistency, we adopted a simple model nomenclature:

- The first two letters represent the type of model i.e, architecture.
- An underscore(_) seperates the model name from versioning.
- The versioning is represented by a 'V' followed by the percentage of data use for training.

For example, "dn_V80" means the model is DenseNet with 80 percent training data.

5.1 Accuracy

Accuracy is the percentage of correct predictions out of the total predictions made by the model. Accuracy is one of the commonly used metric to evaluate and compare the performance of the models The Table 5.1 displays the obtained accuracy results for individual and ensemble models with respect to the training groups.

Table 5.1: Accuracy of individual and ensemble models

Model Group	AlexNet	GoogLeNet	DenseNet	ResNet	Ensemble
80% Training	77.12%	79.75%	80.25%	81.10%	82.50%
90% Training	80.32%	81.65%	82.10%	83.25%	82.30%
100% Training	81.45%	82.78%	83.35%	84.12%	83.56%

The accuracy results shows that en_V80 model achieved the highest accuracy in 80% training group, rn_V90 model achieved the best accuracy in 90% training group and rn_V100 achieved highest accuracy in 100% training group. AlexNet had the lowest accuracy in every group. Among the ensemble models, en_V100 performed with highest accuracy.

5.2 Precision

Another performance metric chosen is precision, it is the proportion of true positive predictions out of all positive predictions made by the model. Table 5.2 shows the precision scores for individual and ensemble models with respect to the training groups.

Table 5.2: Precision of individual and ensemble models

Model Group	AlexNet	GoogLeNet	DenseNet	ResNet	Ensemble
80% Training	0.862	0.871	0.890	0.901	0.905
90% Training	0.870	0.887	0.910	0.911	0.930
100% Training	0.876	0.893	0.921	0.921	0.947

Table 5.2 indicates that the Ensemble model achieved the best precision in every training group, whereas AlexNet had performed the lowest in each training group. These findings suggest that the ensemble models can accurately identify positive cases.

5.3 Recall

Another performance metric used to assess the models is recall. Recall measures the ratio of true positive predictions to the total actual positive instances. The Table 5.3 depicts the recall scores for individual and ensemble models with respect to the training groups.

Table 5.3: Recall of individual and ensemble models

Model Group	AlexNet	GoogLeNet	DenseNet	ResNet	Ensemble
80% Training	0.839	0.858	0.856	0.883	0.909
90% Training	0.843	0.866	0.872	0.895	0.920
100% Training	0.851	0.873	0.885	0.900	0.943

From the Table 5.3, ResNet performed consistently in each training group. AlexNet scored the lowest in each group. Overall the ensemble models performed well according to the recall score.

5.4 F1-Score

The F1 score is a good overall evaluation metric as it combines the precision and recall into a single value. F1-score is the harmonic mean of precision and recall, providing a balanced metric. The Table 5.4 shows the F1-scores for individual and ensemble models with respect to the training groups.

Table 5.4 indicates that the Ensemble model achieved the best F1 score in every training group, whereas AlexNet had performed the lowest in each training group.

Table 5.4: F1-Score of individual and ensemble models

Model Group	AlexNet	GoogLeNet	DenseNet	ResNet	Ensemble
80% Training	0.850	0.864	0.873	0.892	0.907
90% Training	0.856	0.876	0.891	0.903	0.925
100% Training	0.863	0.883	0.903	0.911	0.945

en_V100 has the highest F1-score, indicating a balanced performance in terms of precision and recall.

5.5 Hamming Loss

Hamming Loss is the measure of ratio of incorrect labels to the total number of labels in a multi-label classification task. Lower Hamming Loss means better performance. Zero is the ideal value of Hamming Loss. Table 5.5 shows the Hamming Loss values of individual and ensemble models with respect to each training group. The Hamming

Table 5.5: Hamming Loss of individual and ensemble Models

Model Group	AlexNet	GoogLeNet	DenseNet	ResNet	Ensemble
80% Training	0.228	0.203	0.197	0.190	0.175
90% Training	0.197	0.184	0.174	0.163	0.177
100% Training	0.185	0.172	0.158	0.148	0.164

Loss values shows that the en_V80 performed well in 80% training group, rn_V90 performed well in 90% training group and rn_V100 is performing well in 100% training group meaning it is predicting with less number of incorrect predictions.

Table 5.6: Performance Metrics of Individual and Ensemble Models

Model Group	Accuracy	Precision	Recall	F1-Score	Hamming Loss
AlexNet					
an_V80	77.12%	0.862	0.839	0.850	0.228
an_V90	80.32%	0.870	0.843	0.856	0.197
an_V100	81.45%	0.876	0.851	0.863	0.185
GoogLeNet					
gn_V80	79.75%	0.871	0.858	0.864	0.203
gn_V90	81.65%	0.887	0.866	0.876	0.184
gn_V100	82.78%	0.893	0.873	0.883	0.172
DenseNet					
dn_V80	80.25%	0.890	0.856	0.873	0.197
dn_V90	82.10%	0.910	0.872	0.891	0.174
dn_V100	83.35%	0.921	0.885	0.903	0.158
ResNet					
rn_V80	81.10%	0.901	0.883	0.892	0.190
rn_V90	83.25%	0.911	0.895	0.903	0.163
rn_V100	84.12%	0.921	0.900	0.911	0.148
Ensemble					
en_V80	82.50%	0.905	0.909	0.907	0.175
en_V90	82.30%	0.930	0.920	0.925	0.177
en_V100	83.56%	0.947	0.943	0.945	0.164

Chapter 6

Discussion

The AlexNet, DenseNet, ResNet, and GoogLeNet models were trained in groups using three different percentages of augmented HAM10000 data: 80%, 90%, and 100%. The models were then ensembled by training group, resulting in ensembles for the 80%, 90%, and 100% trained models. A total of 15 models: 12 individual and 3 ensemble models were tested on a custom dataset created by extracting relevant image folders from the PH2 and DermNet datasets.

To assess the performance of each model, five metrics were used: precision, recall, accuracy, F1 score, and Hamming Loss. The accuracy results, shown in Table 5.1, indicate that the ResNet models proved their effectiveness among individual models by consistently scoring well—81.10%, 83.25%, and 84.12%—for the 80%, 90%, and 100% training groups, respectively. The ensemble models also demonstrated consistency with an increasing accuracy with increasing training data: 82.50%, 82.30%, and 83.56%. In contrast, AlexNet performed poorly this is likely due to its simpler architecture and less layers, which might not capture the complexity of the HAM10000 dataset. Among the individual models, the rn_V100 performed the best, while among the ensemble models, the en_V100 showed the highest accuracy. However, accuracy is not the only metric to consider. Precision, recall, F1 score, and Hamming Loss provide new perspective on model performance.

The precision results, shown in Table 5.2, suggests that the ensemble models performed well in each training group, indicating that they captured the complexity of the HAM10000 dataset with high precision. Table 5.3 for recall, Table 5.4 for F1 score, and Table 5.5 for Hamming Loss also show that ensemble models consistently outperformed other individual models in every training group in these metrics, proving the argument that ensemble models are the most effective. Even with the lower accuracy, the ensemble models achieved higher precision, recall, and F1 scores, along with lower Hamming Loss, indicating better overall performance. ResNet models demonstrated significant dominance over other individual models due to their architecture, which includes residual connections. This feature provides the model with training stability and the ability to adapt to new data, leading to better prediction and consistent results.

Overall, these findings suggest that ResNet models perform well individually, but the ensemble models provide the best overall performance across various metrics. This supports the idea that combining multiple models architectures will definitely capture the complexity of image data leading to improved results.

RQ1: What advantages and challenges does the ensemble model, combining AlexNet, GoogLeNet, DenseNet, and ResNet, pose when trained on the HAM10000 dataset and evaluated on diverse data sources within the melanoma domain, compared to its individual counterparts?

Answer: The results show that the ensemble model outperformed the individual models across various performance metrics, including accuracy, precision, recall, F1-score, and Hamming loss as shown in the Figures 6.1, 6.2, 6.3, 6.4 and 6.5. The ensemble model achieved the highest accuracy of 83.56% in the 100% training group, surpassing the individual ResNet, DenseNet, GoogLeNet, and AlexNet models. Additionally, the ensemble model demonstrated the best precision (0.947), recall (0.943), and F1-score (0.945) in the 100% training group, indicating its superior ability to balance true positives, false positives, and false negatives. The ensemble model also exhibited the lowest Hamming loss of 0.164 in the 100% training group, suggesting it made the fewest incorrect label predictions.

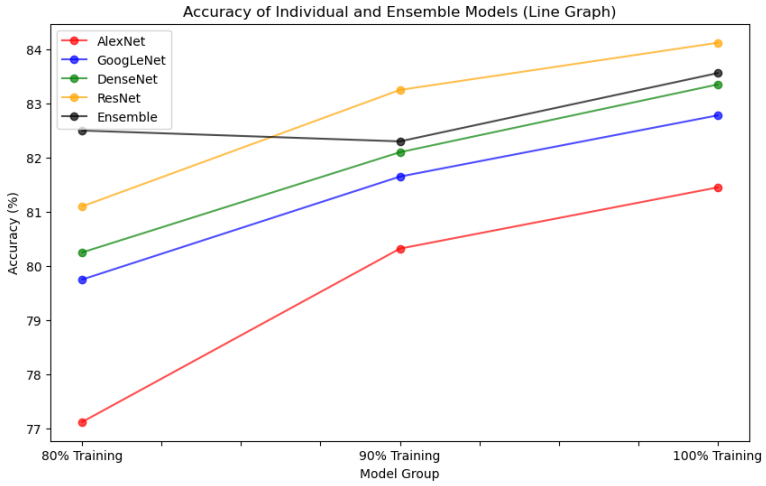


Figure 6.1: Accuracy Line Plot of all models

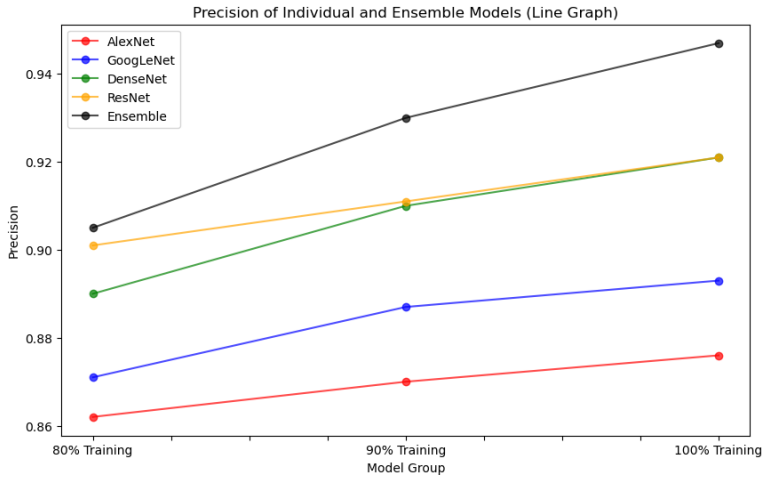


Figure 6.2: Precision Line Plot of all models

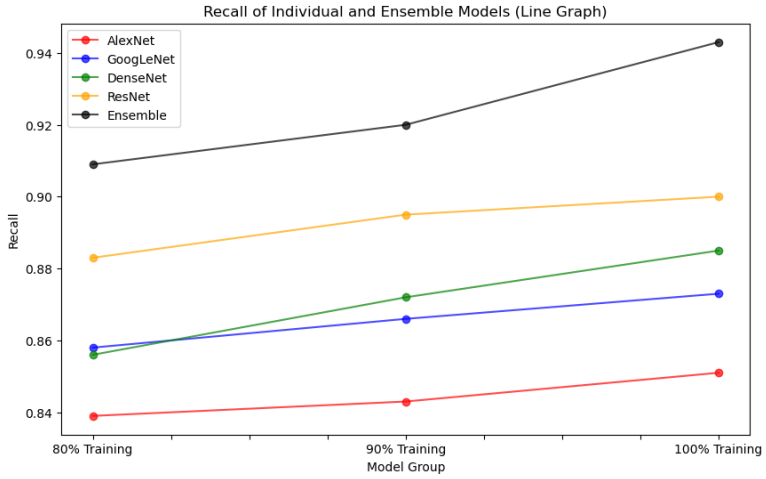


Figure 6.3: Recall Line Plot of all models

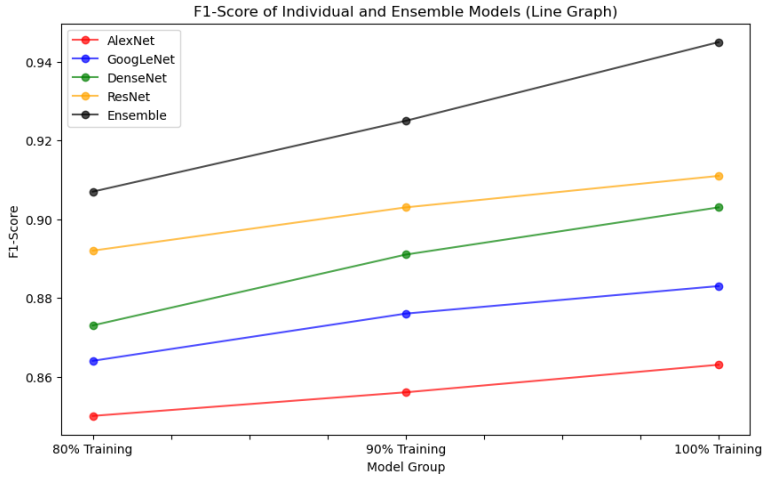


Figure 6.4: F1-score Line Plot of all models

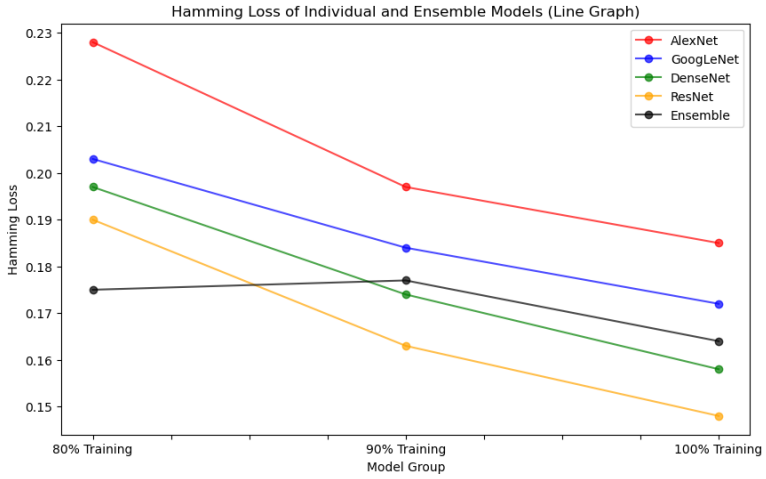


Figure 6.5: Hamming Loss Line Plot of all models

These findings suggest that the ensemble approach was able to leverage the complementary strengths of the individual models, resulting in enhanced overall performance. However, the ensemble model's performance did not always improve as the training data percentage increased, as seen in the 90% and 100% training groups. This indicates that there may be challenges or limitations in further enhancing the ensemble's performance by simply increasing the training data, and alternative strategies, such as model architecture refinements or including additional diverse models, may be necessary.

RQ2: How effective is the HAM10000 dataset while generalizing performance in proposed models when investigated by testing the trained model on other datasets (Dermnet, PH2, ISIC)?

Answer: The results demonstrate that the models trained on the HAM10000 dataset were able to generalize their performance to the other test datasets, Dermnet, PH2, and ISIC, as evidenced by the Hamming loss values (Table 6). The ensemble model, particularly the ResNet-based variants, exhibited the lowest Hamming loss across the different training groups, indicating their ability to capture more generalizable features and patterns from the HAM10000 dataset.

In contrast, the individual AlexNet, GoogLeNet, and DenseNet models showed more variability in their performance across the test datasets, as indicated by the higher Hamming loss values. This suggests that the ensemble approach and the ResNet-based models were better able to maintain their predictive capabilities on the diverse data sources.

6.1 Reflection

These results suggest that the HAM10000 dataset is a suitable and effective resource for training deep learning models for melanoma prediction, as the models were able to generalize their performance to external datasets. However, the results also highlight the importance of evaluating model performance on multiple, diverse datasets to ensure the robustness and generalizability of the developed melanoma prediction system.

The superiority of the ensemble model can be seen, it consistently outperformed individual models across various performance metrics, highlighting the strength of combining diverse architectures like AlexNet, GoogLeNet, DenseNet, and ResNet. ResNet models, in particular, demonstrated robustness and adaptability due to their sophisticated and residual architecture, making them the best performers among individual models. Simpler models like AlexNet lacked in capturing the dataset complexity, highlighting the need for advanced architectures in medical image analysis. However, the ensemble model's performance did not improve with more training data, it stayed consistent. Data augmentation techniques proved to be particularly useful with individual model accuracies. This research had better results than previous similar work due to data augmentation with external data and experimentation with different ratios of training and test ratios. The HAM10000 dataset alone has biased

data because of very different image numbers under each label. Data augmentation helped to balance this issue.

6.2 Limitations

While the ensemble model performed better overall, it did not always improve significantly with additional training data. The complex HAM10000 dataset proved to be difficult for simpler models like AlexNet to handle, demonstrating that these models can have difficulty processing detailed data. The HAM10000 dataset, which was primarily employed in the study for training, may not fully represent the range of real-world melanoma variants, which could have an impact on the models' practical performance. Additionally, while merging different models resulted in better performance, doing so increased system complexity and computational resource requirements, which may not be feasible in actual clinical situations. Following research efforts should be focused on refining the construction of these hybrid models, utilising a wider range of datasets, and properly balancing model complexity and efficacy.

Chapter 7

Conclusions and Future Work

This thesis evaluated the performance of various deep learning models, namely AlexNet, DenseNet, ResNet, and GoogLeNet. These models were trained on different percentages of augmented HAM10000 data for classifying skin lesion images. The goal was to analyze the generalizing ability of these models while assessing the quality of HAM10000 medical data. This study also assessed whether the ensemble model could outperform individual models.

The results showed that ResNet achieved the highest accuracy among the individual models, likely due to its architecture with residual connections, which provides training stability. However, the ensemble models surpassed individual models across several metrics, including precision, recall, F1 score, and Hamming Loss. The combination of various model architectures allows for a more complete representation of the data, leading to more reliable classification outcomes. The ensemble models' superior performance, particularly in terms of precision and recall, has important implications for clinical applications, where the accuracy and reliability of image classification are critical.

The HAM10000 dataset proved to be suitable and effective resource for training deep learning models for melanoma prediction, as the models were able to generalize their performance to external datasets.

7.1 Future Work

Although this study achieved significant results, there are numerous potential directions for future work. A few of these are outlined below:

- **Advanced Data Augmentation:** Given the limited size of the medical datasets, there is huge scope for data augmentation optimization such as using more advanced techniques like Generative Adversarial Networks (GANs) and Variational Auto Encoders (VAE).
- **Transfer Learning:** Using more complex pretrained models like Vision Transformers (ViT) as they use attention based mechanisms to capture complex patterns in the data.
- **More Diverse Datasets:** Although this study used HAM10000, PH2 and DermNet datasets, future work can incorporate much diverse datasets to reduce model bias and increase diversity.

- **Explainable AI (XAI):** By applying SHAP or LIME techniques to deep learning models, one can visualize which features or areas of the image have the most influence on the model's decisions, allowing clinical validation.
- **Quantum Approach:** Given the importance of research in healthcare, incorporating deep learning with quantum computing can drastically enhance model accuracy and reliability, leading to more trustworthy outcomes.

References

[1] Advanced guide to inception v3 | cloud TPU | google cloud. [Online]. Available: https://cloud.google.com/tpu/docs/inception-v3-advanced

[2] HAM10000: Data augmentation optimisation. [Online]. Available: https://kaggle.com/code/jnegrini/ham10000-data-augmentation-optimisation

[3] Hands-on machine learning with scikit-learn, keras, and TensorFlow, 2nd edition [book]. ISBN: 9781492032649. [Online]. Available: https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/

[4] Introduction to matplotlib. Section: Python. [Online]. Available: https://www.geeksforgeeks.org/python-introduction-matplotlib/

[5] ISIC2018 challenge task1 data (segmentation). [Online]. Available: https://www.kaggle.com/datasets/tschandl/isic2018-challenge-task1-data-segmentation

[6] Keras: The high-level API for TensorFlow | TensorFlow core. [Online]. Available: https://www.tensorflow.org/guide/keras

[7] Learning model building in scikit-learn. Section: AI-ML-DS. [Online]. Available: https://www.geeksforgeeks.org/learning-model-building-scikit-learn-python-machine-learning-library/

[8] NumPy -. [Online]. Available: https://numpy.org/

[9] sklearn.preprocessing.LabelEncoder. [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

[10] Training parameters - amazon machine learning. [Online]. Available: https://docs.aws.amazon.com/machine-learning/latest/dg/training-parameters.html

[11] Transfer learning with keras using DenseNet121 | by bouzouitina hamdi | medium. [Online]. Available: https://bouzouitina-hamdi.medium.com/transfer-learning-with-keras-using-densenet121-fffc6bb0c233

[12] What are convolutional neural networks? | IBM. [Online]. Available: https://www.ibm.com/topics/convolutional-neural-networks

[13] What is early stopping? [Online]. Available: https://www.educative.io/answers/what-is-early-stopping

[14] What is normalization in machine learning. [Online]. Available: https://deepchecks.com/glossary/normalization-in-machine-learning/

[15] Why you need to compile your keras model before using model.evaluate() | saturn cloud blog. Section: blog. [Online]. Available: https://saturncloud.io/blog/why-you-need-to-compile-your-keras-model-before-using-modelevaluate/

[16] G. Alwakid, W. Gouda, M. Humayun, and N. Us Sama, "Melanoma detection using deep learning-based classifications," *International Journal of Environmental Research and Public Health*, vol. 10, no. 1, p. 26, 2023.

[17] S. Bhosale. (2022, 6) Alexnet architecture explained. Online. [Online]. Available: https://medium.com/@siddheshb008/alexnet-architecture-explained-b6240c528bd5

[18] E.-G. Dobre, M. Surcel, C. Constantin, M. A. Ilie, A. Caruntu, C. Caruntu, and M. Neagu, "Skin cancer pathobiology at a glance: A focus on imaging techniques and their potential for improved diagnosis and surveillance in clinical cohorts," vol. 24, no. 2, p. 1079. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9866322/

[19] S. Goel. (2024) Dermnet - skin disease image data. Kaggle Dataset. [Online]. Available: https://www.kaggle.com/datasets/shubhamgoel27/dermnet

[20] H. Habehh and S. Gohel, "Machine learning in healthcare," vol. 22, no. 4, p. 291, publisher: Bentham Science Publishers. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8822225/

[21] M. M. Hossain, M. M. Hossain, M. B. Arefin, F. Akhtar, and J. Blake, "Combining state-of-the-art pre-trained deep learning models: A noble approach for skin cancer detection using max voting ensemble," vol. 14, no. 1, p. 89, number: 1 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2075-4418/14/1/89

[22] G. Huang and et al. (2016) Densenet. Online. [Online]. Available: https://paperswithcode.com/method/densenet#:~:text=A%20DenseNet%20is%20a%20type,sizes)%20directly%20with%20each%20other

[23] IBM. (2024) Deep learning. Accessed: Feb. 2, 2024. [Online]. Available: https://www.ibm.com/topics/deep-learning

[24] Kaggle. (2018, 9) Skin cancer mnist: Ham10000. Online. [Online]. Available: https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000

[25] O. O. Kushimo, A. O. Salau, O. J. Adeleke, and D. S. Olaoye, "Deep learning model to improve melanoma detection in people of color," *Arab Journal of Basic and Applied Sciences*, vol. 30, no. 1, pp. 92–102, Dec 2023.

[26] K. Le. Alexnet and image classification. [Online]. Available: https://lekhuyen.medium.com/alexnet-and-image-classification-8cd8511548b4

[27] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," vol. 3, no. 1, pp. 91–99. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666285X22000565

[28] T. Mendonça, P. M. Ferreira, J. Marques, A. R. S. Marcal, and J. Rozeira, "Ph^2: A dermoscopic image database for research and benchmarking," 2013. [Online]. Available: https://www.fc.up.pt/addi/ph2%20database.html

[29] S. Mukherjee. The annotated ResNet-50. [Online]. Available: https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758

[30] MyGreatLearning. (2020, 9) Introduction to resnet or residual network. Online. [Online]. Available: https://www.mygreatlearning.com/blog/resnet/

[31] M. Nielsen, "Neural networks and deep learning."

[32] K. Nyuytiymbiy. Parameters and hyperparameters in machine learning and deep learning. [Online]. Available: https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac

[33] R. Python. Split your dataset with scikit-learn's train_test_split() – real python. [Online]. Available: https://realpython.com/train-test-split-python-data/

[34] H. Rashid, M. A. Tanveer, and H. Aqeel Khan, "Skin lesion classification using GAN based data augmentation," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 916–919, ISSN: 1558-4615. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8857905?casa_token=WdK7cusr8k4AAAAA:39IinZlbrJlDtNZB-bYLdqAnQypgg5CtqODMWToN1v09qFa5lPJvp2J_bDCq8VMonqARnb_s3Q

[35] R. Raza, F. Zulfiqar, S. Tariq, G. Anwar, A. B. Sargana, and Z. Habib, "Melanoma classification from dermoscopy images using ensemble of convolutional neural networks," *Mathematics*, vol. 10, p. 26, Dec 2021.

[36] T. N. Rincy and R. Gupta, "Ensemble learning techniques and its efficiency in machine learning: A survey," in *2nd International Conference on Data, Engineering and Applications (IDEA)*, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9170675?casa_token=3ooHJsi3V0YAAAAA:3CC0ITWEPI8FYnUirqvU_GbrBU5H20tJVzaPjsFDgG92mYoTxs3wXMwnSxYu-qZeI78UFhepeg

[37] Skin Cancer Foundation. (2024) Melanoma. Accessed: Feb. 2, 2024. [Online]. Available: https://www.skincancer.org/skin-cancer-information/melanoma/

[38] C. Szegedy and et al. (2014) What is googlenet? Online. [Online]. Available: https://www.educative.io/answers/what-is-googlenet

[39] K. Team. Keras documentation: Optimizers. [Online]. Available: https://keras.io/api/optimizers/

[40] T. P. D. Team, "pandas: Powerful data structures for data analysis, time series, and statistics." [Online]. Available: https://pandas.pydata.org

[41] G. Trova. The pros and cons of explainable AI: Gaining trust in AI models. [Online]. Available: https://www.softwareimprovementgroup.com/unraveling-the-incomprehensible-the-pros-and-cons-of-explainable-ai/

[42] M. Waskom, "seaborn: statistical data visualization," vol. 6, no. 60, p. 3021. [Online]. Available: https://joss.theoj.org/papers/10.21105/joss.03021

[43] World Cancer Research Fund International. (2024) Skin cancer statistics. Accessed: Jan. 31, 2024. [Online]. Available: https://www.wcrf.org/cancer-trends/skin-cancer-statistics/

[44] World Health Organization. (2024) Cancer. Accessed: Feb. 2, 2024. [Online]. Available: https://www.who.int/health-topics/cancer

[45] Y. Wu, B. Chen, A. Zeng, D. Pan, R. Wang, and S. Zhao, "Skin cancer classification with deep learning: A systematic review," vol. 12, p. 893972.

Faculty of Computer Science Engineering, Blekinge Institute of Technology, 371 79
Karlskrona, Sweden