SKIN CANCER DETECTION AND CLASSIFICATION FROM DERMATOSCOPIC IMAGES USING DEEP LEARNING METHODS

A THESIS
SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By

Serdar KALAYCI

JUNE 2023

Serdar KALAYCI

A Master's Thesis

AGU 2023

SKIN CANCER DETECTION AND CLASSIFICATION FROM DERMATOSCOPIC IMAGES USING DEEP LEARNING METHODS

A THESIS
SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE OF
ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By

Serdar KALAYCI

JUNE 2023

SCIENTIFIC ETHICS COMPLIANCE

I hereby declare that all information in this document has been obtained in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Serdar KALAYCI

REGULATORY COMPLIANCE

M.Sc. thesis titled **“SKIN CANCER DETECTION AND CLASSIFICATION FROM DERMATOSCOPIC IMAGES USING DEEP LEARNING METHODS”** has been prepared in accordance with the Thesis Writing Guidelines of the Abdullah Gül University, Graduate School of Engineering & Science.

Prepared By	Advisor
Serdar KALAYCI	Prof. Bülent YILMAZ

Head of the Electrical and Computer Enginering Graduate Program
Assoc. Prof. Zafer AYDIN

ACCEPTANCE AND APPROVAL

M.Sc. thesis titled **“SKIN CANCER DETECTION AND CLASSIFICATION FROM DERMATOSCOPIC IMAGES USING DEEP LEARNING METHODS”** and prepared by Serdar Kalaycı has been accepted by the jury in the Electrical and Computer Engineering Graduate Program at Abdullah Gül University, Graduate School of Engineering & Science.

08 / 06 / 2023

JURY:

Advisor	: Prof. Bülent YILMAZ
Member	: Assoc. Prof. Zafer AYDIN
Member	: Assoc. Prof. İsa YILDIRIM

APPROVAL:

The acceptance of this M.Sc. thesis has been approved by the decision of the Abdullah Gül University, Graduate School of Engineering & Science, Executive Board dated ….. /….. / ……….. and numbered …………..……. .

……….. /……….. / ………..

Graduate School Dean
Prof. İrfan ALAN

ABSTRACT

SKIN CANCER DETECTION AND CLASSIFICATION FROM DERMATOSCOPIC IMAGES USING DEEP LEARNING METHODS

Serdar KALAYCI
MSc. in Electrical and Computer Engineering
Supervisor: Prof. Bülent YILMAZ

June 2023

Early detection of skin cancer is crucial for successful treatment and improved patient outcomes. The most prevalent form of cancer is skin cancer and if left undetected, it can spread and become more difficult to treat. A dangerous and frequently fatal type of skin cancer is melanoma. Regular skin examinations and self-examinations can help identify suspicious moles or lesions, which can then be evaluated by a dermatologist. In addition, advances in technology and artificial intelligence have enabled the development of tools for automated skin cancer screening, providing a convenient and efficient means of early detection. This can lead to more efficient diagnosis, reduced healthcare costs and improved patient care. By evaluating skin lesions from images, deep learning techniques have shown considerable potential in increasing the precision of melanoma detection. By using large datasets and complex neural networks, deep learning algorithms can effectively distinguish between benign and malignant skin lesions with high accuracy. Ensemble of CNN models helps improve the performance and reliability of the classification task. By combining the predictions of multiple CNN models lead to more accurate and robust predictions. In this thesis, for melanoma classification problem, many different data augmentations techniques applied and different convolutional neural networks architectures evaluated, applied vignetting effect filter and hair noise in accordance with the dataset and results of ensemble of the best CNN models are promising. This thesis attempts to produce a reliable model for the classification of melanoma by conducting experiments on two combined publically accessible data sets, ISIC 2019 and ISIC 2020. On the testing sets in our studies, the proposed solution attained 95.75% AUC.

Keywords: Deep Learning, Convolutional Neural Networks, Vignetting Effect, Hair Noise, Skin Cancer

ÖZET

DERİN ÖĞRENME YÖNTEMLERİ KULLANARAK DERMATOSKOPİK GÖRÜNTÜLERDEN OTOMATİK CİLT KANSERİ TESPİTİ VE SINIFLANDIRILMASI

Serdar KALAYCI
Elektrik ve Bilgisayar Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Prof.Dr. Bülent YILMAZ
Haziran-2023

Cilt kanserinin erken teşhisi, başarılı tedavi ve daha iyi hasta sonuçları için çok önemlidir. Cilt kanseri en yaygın kanser türüdür ve tespit edilmezse yayılabilir ve tedavisi daha zor hale gelebilir. Melanom, cilt kanserinin ciddi ve genellikle ölümcül bir şeklidir. Düzenli cilt muayeneleri daha sonra bir dermatolog tarafından değerlendirilebilecek olan şüpheli benleri veya lezyonları belirlemeye yardımcı olabilir. Buna ek olarak, teknolojideki ve yapay zekadaki gelişmeler, otomatik cilt kanseri taraması için araçların geliştirilmesini mümkün kıldı ve erken teşhis için uygun ve etkili bir araç sağladı. Bu, daha verimli tanıya, daha düşük sağlık maliyetlerine ve daha iyi hasta bakımına yol açabilir. Derin öğrenme yöntemleri, görüntülerden cilt lezyonlarını analiz ederek melanom tespitinin doğruluğunu artırmada büyük umut vaat ediyor. Derin öğrenme algoritmaları, büyük veri kümelerini ve karmaşık sinir ağlarını kullanarak iyi huylu ve kötü huylu cilt lezyonlarını yüksek doğrulukla etkili bir şekilde ayırt edebilir. CNN modelleri topluluğu, sınıflandırma performansını ve güvenilirliğini artırmaya yardımcı olur. Birden fazla CNN modelinin tahminlerini birleştirilmesi daha doğru ve sağlam tahminlere yol açar. Bu tezde, melanom sınıflandırma problemi için birçok farklı veri artırma tekniği uygulanmış ve farklı evrişimli sinir ağları mimarileri değerlendirilmiş, veri setine uygun olarak uygulanan vinyet etkisi ve kıl gürültüsü ve en iyi CNN modellerinden oluşan topluluk sonuçları umut vericidir. Bu tez, halka açık iki veri seti olan ISIC 2019 ve ISIC 2020 üzerinde deneyler yaparak melanom sınıflandırması için sağlam bir model oluşturmayı amaçlamaktadır. Çalışmalarımızda, önerdiğimiz çözüm test setlerinde %95,75 doğruluk elde etti.

Anahtar kelimeler: Derin Öğrenme, Evrişimli Sinir Ağları, Vinyetting Etki, Kıl Gürültüsü, Cilt Kanseri

Acknowledgements

I would like to extend my thanks to my supervisor Prof. Bülent Yılmaz for his supervision and support. I also want to thank my family and especially my dear wife, for their patience and tolerance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DNA	Deoxyribonucleic Acid
UV	Ultraviolet
SVM	Support Vector Machine
RF	Random Forest
CNN	Convolutional Neural Network
KNN	K-Nearest Neighbors
NB	Naïve Bayes
DL	Deep Learning
BCE	Binary Cross Entropy
AUC	Area Under Curve
MBConv	Mobile Inverted Bottleneck Convolution
CLAHE	Contrast Limited Adaptive Histogram Equalization
ISIC	International Skin Imaging Collaboration
AR	Augmented Reality
VR	Virtual Reality
GPU	Graphics Processing Unit
CPU	Central Process Unit

To my sons, whom I wish to be great scientists in the future

Chapter 1

Introduction

1.1 Motivation

Skin cancer that is on the rise and is severe and potentially lethal is melanoma. It ranks in the top five most prevalent malignancies in males and the top seven in women. With 2% people getting it, it is relatively common in the general population. Additionally, melanoma is to blame for 75% of skin cancer-related fatalities [1].

A diagnostic technique called dermatoscopy enables a closer study of the skin without the need for invasive procedures. At first, it was mainly utilized to evaluate pigmented melanocytic lesions, however, it has now been applied in multiple areas of dermatology. The dermatoscope is increasingly being used as a diagnostic tool, similar to how the stethoscope is used by general practitioners and pathologists [2].

Dermatoscopy images have been used to classify melanoma and deep learning has emerged as a promising tool that could be more accurate and effective than conventional techniques. Researchers have looked into using deep learning models on dermatoscopy images, which offer precise views of skin lesions, in order to make more accurate and automated melanoma diagnoses. Deep learning has been proven successful in melanoma classification using dermatoscopy images in numerous studies. When using dermatoscopy images to distinguish between malignant and benign skin lesions, Lopez et al. [3] developed a deep neural network that obtained good accuracy. A deep ensemble architecture was presented by Codella et al. [4] and outperformed human experts in melanoma classification tests using dermatoscopy images.

Deep learning and ensemble methods for melanoma classification have produced encouraging outcomes. The ensemble can gain from the diversity and complementary qualities of distinct models by merging the predictions of various models. This can lessen the effects of overfitting, cut down on generalization errors and increase the stability of the classification system. Rather than using a single model, in order to accurately detect melanoma, Kim et al. [5] suggested an ensemble model built on multiple deep residual networks.

1.2 Organization of the Thesis

The remainder of the thesis is structured as follows: A thorough overview of the steps in medical diagnosis and dermatoscopy in this field is given in Chapter 2, which starts with medical facts about skin and skin cancer, the next section provides general information on machine learning as well as technical information on the CNN models utilized in this thesis to discuss deep learning and transfer learning and finally a general overview of ensemble learning methodologies is offered. Chapter 3 provides a thorough description of the research using medical and dermatoscopy images that were carried out using machine learning and intricate deep learning techniques. Chapter 4 begins by giving comprehensive details regarding the data set used in this thesis and then the flow of the proposed system and the key procedures were presented followed by metrics for evaluating performance requirements. Numerous approaches that are described in the training flow are explained and compared in Chapter 5. In Chapter 6, the results of the suggested techniques are covered and Chapter 7 concludes by summarizing the argument, outlining its important contributions, explaining where the findings take us and what possible future research might entail.

Chapter 2

Background

2.1 Skin Cancer

The biggest organ in the human body, the integumentary system, which includes the skin and its appendages (hair, nails, perspiration and oil glands), has an average surface area of 2.0 square meters. The skin's primary function is to shield the body from various external factors, such as germs, chemicals and temperature. Additionally, the skin has bacterial-killing secretions and the pigment melanin serves as a chemical barrier against UV radiation, which can injure skin cells [6].

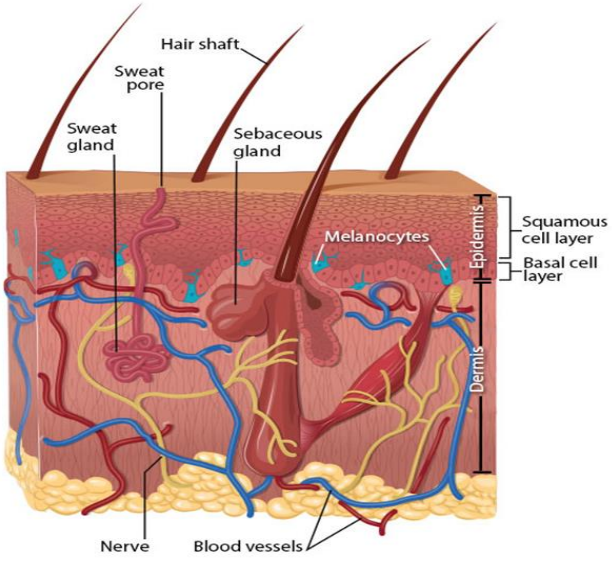


Figure 2.1 Skin layers [7]

Although the skin is made up of several layers, the epidermis, which is the top layer and the dermis, which is the layer below it, are the two main layers, as shown in figure 2.1.

Uncontrolled cell development on the skin due to alterations in DNA structure causes what leads to skin cancer. In the world, it is the 17th most prevalent cancer, with 1.8 million cases anticipated in 2020 [8]. For patients whose melanoma is found early, the predicted five-year survival rate is over 99 percent. The survival rate falls to 71% when the disease progresses to the lymph nodes and to 32% when it spreads to distant organs [9]. The majority of skin cancer cases occur on sun-exposed body parts such the hands, face, neck, arms, ears, chest and scalp [10].

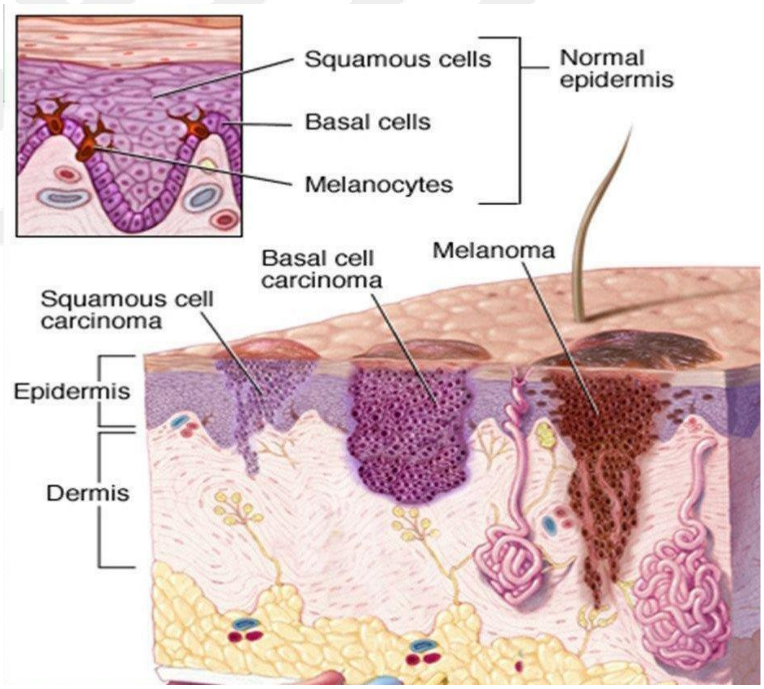


Figure 2.2 Skin layers and the most common skin cancer types [11]

The three most prevalent types of skin cancer are melanoma, squamous cell carcinomas and basal carcinomas. The two most typical kinds of cancer are basal and squamous cell carcinomas. They both start in the skin's basal and squamous layers and while both are typically curable, they can both be disfiguring and expensive to treat. Contrarily, melanomas, the third most common form of skin cancer, begin in the melanocytes. It is the most fatal because to its propensity to spread to other bodily parts, especially vital organs

[10]. Excessive exposure to ultraviolet (UV) rays from the sun, tanning beds or sunlamps is the main cause of skin cancer. Sunburns can occur as a result of UV radiation damaging skin cells. But over time, UV harm accumulates and causes skin texture changes, early aging and occasionally skin cancer. In contrast to malignancies that develop within, skin cancers form on the outside and are typically noticeable. By keeping an eye out for odd changes in the skin, skin cancer can be diagnosed early. Early skin cancer detection offers the best chance for successful medical care [12].

2.2 Medical Diagnosis

The medical diagnosis is influenced by the patient's past, social interactions , ethnicity and exposure to sun. In the office, suspicious lesions are biopsied and sent to the laboratory for permanent paraffin section processing and pathologist evaluation on representative glass slides [13].

2.2.1 Medical Practices in Dermatology

The study of illnesses of the skin, hair and nails is known as dermatology which focuses on their diagnosis and treatment. Dermatology encompasses a wide range of research and clinical activities aimed at understanding and diagnosing various skin conditions and abnormalities. It involves the examination and assessment of both normal and abnormal skin conditions, as well as the diagnosis and management of skin diseases, cosmetic concerns, cancers and aging-related issues.

Dermatology comprises the study of the skin, subcutaneous hair, fat, oral mucaso, nails and genital membranes. It involves the use of different investigative techniques and therapeutic approaches, including dermatohistopathology (the microscopic examination of skin tissue), topical and systemic medications, cosmetic procedures, dermatologic surgery, phototherapy, radiotherapy immunotherapy and laser therapy [14].

2.2.2 Imaging in Dermatology

Technology has been successfully adapted in the field of dermatology to improve visual skin examination. The dermatologist now has new instruments to noninvasively examine skin features both macroscopically and microscopically because of advancements in optics and light technology. Although there is a learning curve involved, dermatoscopy has established itself as the gold standard of care in the majority of dermatologic practices. There have been various studies establishing its use and demonstrating its benefits due to its widespread availability. Similar to dermatoscopy, dermatologists can now see the skin more clearly thanks to the use of digital imaging, 3D imaging, ultrasound and optical coherence tomography. Whether used to track nevus progression or noninvasively diagnose, detect or characterize cancer margins, these techniques are transforming the sector [15].

2.2.3 Dermatoscopy and Its Advantages with Limitations

The use of a dermatoscope to examine skin lesions is known as dermatoscopy. This technique, sometimes referred to as dermatoscopy or epiluminescence microscopy, enables examination of skin lesions without being impeded by skin surface reflections. It is an in-vivo method that has long been effective for assessing suspected skin lesions [16].

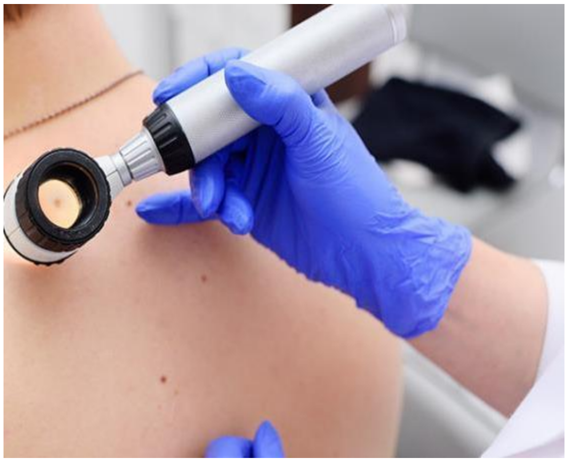


Figure 2.3 Dermatoscopy [17]

To identify lesions and distinguish non-melanoma skin malignancies such basal cell carcinoma or squamous cell carcinoma or melanocytic lesions from dysplastic lesions melanomas dermatoscopy can be used. A growing number of dermatological disorders, including those affecting the scalp, hair and nails, as well as pigmentary dermatoses, inflammatory dermatoses, infectious dermatoses, have recently been identified as having dermatoscopy. As the utility of dermatoscopy increases, practitioners in virtually all specialties should be knowledgeable about this simple, non-invasive and high-yield diagnostic technique [18]. When done by specialists, dermatoscopy has been shown in numerous studies to be helpful in the identification of melanoma. It could improve clinical diagnosis accuracy by up to 35% and lessen the need to remove benign lesions. It can cause primary care to refer to more suspicious lesions and less common ones [19].

Dermatoscopy has many benefits. It magnifies skin 10 times to make it easier to diagnose a variety of skin lesions. Detects both pigmented and non-pigmented skin cancer more sensitively, precisely and accurately than unaided eyes. Compared to a visual examination using just the eyes alone, dermatoscopy increases the accuracy of skin lesion diagnosis. It facilitates the differentiation between benign and malignant lesions by allowing the observation of fine structures and patterns that are invisible to the human eye. Due to its ability to identify specific dermatoscopic characteristics linked to malignancy, dermoscopy aids in the early detection of melanoma. A non-invasive approach, dermatoscopy does not include any invasive treatments. It is a recommended option for standard skin inspections because it is painless and well-tolerated by patients. On the other hand, it has some drawbacks as well. It requires proper training and outcomes interpretation is arbitrary. Applications are limited by the low magnification. Dermatoscopic image interpretation can be arbitrary, which causes inter-observer variability among various dermatologists. Standardized criteria and training programs can increase consistency in interpretation. Dermoscopy is mostly used for pigmented lesions like melanoma and melanocytic nevi. Due to the less well-defined dermatoscopic features of non-melanocytic and non-pigmented lesions, it may only be partially useful in treating these conditions. Dermatoscopy requires specialist tools which could be expensive. In some circumstances, especially those with little resources, access to high-quality dermatoscopes may be restricted [20].

2.3 Machine Learning

Machine learning is a branch of artificial intelligence that focuses on developing mathematical models and algorithms that can recognize patterns and insights in data without explicit programming [21]. This implies that a computer can be trained on a sizable dataset and then utilize that training to make decisions or predictions about new data.

2.3.1 Categories of Machine Learning Algorithms

Four major categories can be used to classify machine learning algorithms, which are based on the types of learning problems they are designed to address. A list of these categories is:

- **Supervised Learning:** Using labeled data, an algorithm is trained in supervised learning. This indicates that a related output or target variable is linked to the input data. The algorithm gains knowledge from this labeled data and can subsequently be used on new, unlabeled data to generate predictions or choices. By adjusting the algorithm's weights or parameters during training, reduce the gap between the output that is anticipated and the output that is actually produced by using supervised learning [21].
- **Unsupervised Learning:** Using algorithms to assess unclassified or unlabeled data is the subject of the machine learning subfield known as unsupervised learning. The input data does not have a corresponding output or target variable and the algorithm learns to identify patterns, relationships or clusters without any prior information about the data. Unsupervised learning's ultimate goal is to extract meaningful insights and structure from the data, such as hidden patterns or collections of related data points [22].
- **Semi-supervised Learning:** Semi-supervised learning, a kind of machine learning, is used to create a model from a mix of unlabeled and labeled data. The labeled data is used to learn patterns and make predictions, while the unlabeled data helps to improve the accuracy of predictions. This method is advantageous since it is often more difficult and costly to obtain labeled data than unlabeled data. Semi-supervised learning algorithms frequently achieve higher accuracy than supervised learning

algorithms because they make use of the wealth of unlabeled data rather than only labeled data [23].

- **Reinforcement Learning:** A type of machine learning called reinforcement learning (RL) involves an agent interacting with its surroundings and learning from the input it receives. The agent is rewarded or punished based on its actions in the environment and its objective is to discover the best behavior that will maximize the total reward in the long run. By experimenting and adapting its behavior based on feedback, the agent learns through trial and error. In a variety of industries, including robots, gaming and recommendation engines, reinforcement learning has been successfully applied [24].

2.3.2 Neural Networks

Mathematical models called neural networks replicate the composition and functionality of actual neurons. These models consist of numerous interconnected processing units termed nodes or artificial neurons. Each neuron gets input from other neurons, adds up those inputs, performs a nonlinear transformation and then sends the outcome to other neurons. Usually, the connections between neurons are weighted, signifying that some inputs have more impact than others on the output of the neuron [25].

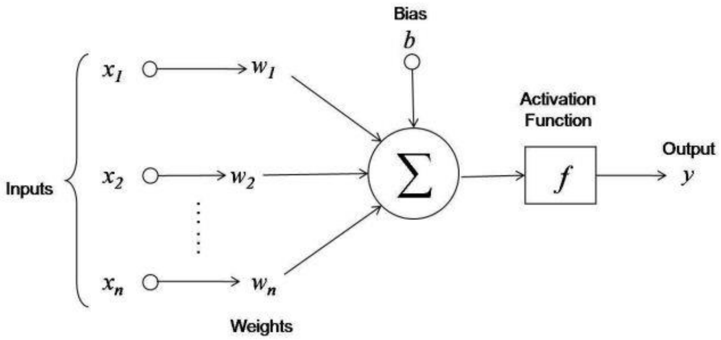


Figure 2.4 A basic artificial neuron [26]

Neural networks come in a wide range of varieties. There are many more neural network types, as well as modifications and combinations of the more popular types, which are listed below:

- **Feed forward neural networks:** The most typical kind of neural network, where data travels in a straight line from input nodes through hidden layers to output nodes.

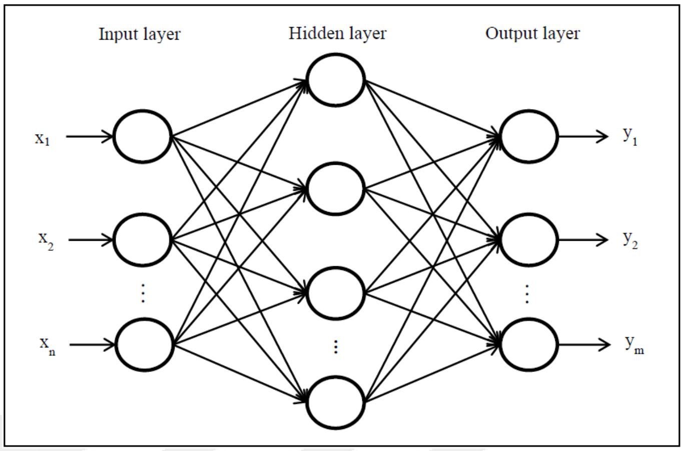


Figure 2.5 The two layered feed forward neural network's structure [27]

- **Recurrent neural networks (RNNs):** Networks having feedback connections, which enable data to loop back into the network, are able to process input sequences like time series data or text written in natural language [28].
- **Convolutional neural networks (CNNs):** Convolutional layers apply filters to find local patterns in the input in image processing networks. Through the use of convolutional layers, CNNs are created to automatically recognize and extract hierarchical patterns and characteristics from the input data [29].
- **Autoencoders:** Networks created for unsupervised learning that reduce the dimensions of input data before reconstructing the original input from the reduced representation [30].
- **Generative adversarial networks (GANs):** Networks created to produce artificial data that is similar to the training data. They consist of two networks: a discriminator network that attempts to distinguish fake data from actual data and a generator network that creates new data [31].
- **Reinforcement learning networks (RLNs):** Networks designed for learning from trial-and-error interactions with an environment, where the network receives rewards or punishments for actions taken in the environment [32].

2.3.3 Convolutional Neural Networks

A subclass of deep neural networks called convolutional neural networks has showed promise in a number of computer vision applications, including object segmentation, image classification and object detection. The visual cortex in the brain has been the inspiration for CNNs, a kind of feedforward neural network that aims to automatically learn meaningful regions from images and extract features from images without the need for manual feature engineering [29]. CNNs typically consist of many layers, including the following, which are described in detail in the subsections.

2.3.3.1 Input Layer

A convolutional neural network's input layer stores input images as an array of numbers, where each member represents a pixel in the image. Based on their dimensions and pixel count, the images are represented as a matrix array.

2.3.3.2 Convolution Layer

In order to extract features from the input data, the convolutional layer, which is the main component, is accountable. The convolution procedure is carried out by a series of learnable filters sliding over the input image or feature map [33].

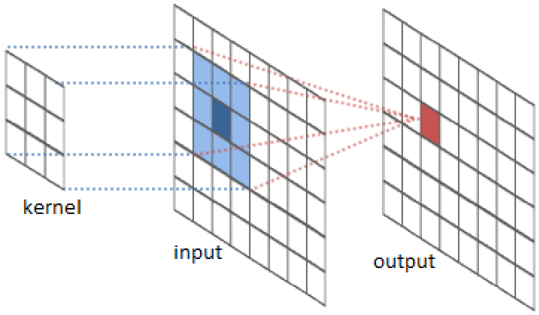


Figure 2.6 Convolution operation [34]

By conducting a dot product between each filter's weights and a specific area of the input data, the output feature map is produced by each filter as a single value. The convolutional layer may learn to recognize patterns and features at various scales and locations by swiping the filters over the input data. It can then learn to recognize intricate features and patterns in the input data.

2.3.3.3 Pooling Layer

Following convolutional layers, pooling layers are frequently applied to cut down on the spatial dimensions of the feature maps and add some translation invariance to the network. The most prevalent pooling technique is max pooling, which only retains the greatest value obtained in a specific local region of the feature map and discards the remaining values.

2.3.3.4 Activation Function

The activation function layer of a convolutional neural network adds nonlinearity to the output of the layer before it. Each neuron's output from the layer before is subjected to the element-by-element application of the activation function to create a new output [28]. This layer's goal is to give the model nonlinearity so that it can pick up on intricate, nonlinear interactions between the input and output. The sigmoid function, ReLU (Rectified Linear Unit) [35] and its subtypes Leaky ReLU [36] and ELU (Exponential Linear Unit) [37] are common activation functions.

2.3.3.5 Dropout

Dropout regularization technique is used by convolutional neural networks to improve generalization performance and decrease overfitting [38]. During the training phase, a certain percentage of neurons in a layer are randomly deactivated. By "dropping out" or "zeroing out" a portion of the neuron activations during each training iteration, the dropout strategy works by adding unpredictability into the network. Different combinations of neurons are triggered or silenced during training, forcing the network to learn increasingly robust and generalizable characteristics. Specifically defining a dropout probability or dropout rate, which establishes the likelihood of deactivating a neuron, is how dropout is really utilized in practice. A random binary mask is applied to the activations of the neurons in the dropout layer during training, setting a portion of them to zero for each input example. The entire network is utilized during inference or testing without dropout, but the weights of the neurons are changed to reflect the decreased activations during training.

2.3.3.6 Batch Normalization

Convolutional neural networks can be trained more steadily and quickly by using the batch normalization technique [39]. It entails dividing by the standard deviation and subtracting the mean from the activations of a layer over a small sample of training samples. By minimizing internal covariate shift or the alteration in the distribution of layer activations during training, batch normalization benefits in the stabilization of the training process. As a result, there is a quicker network convergence and less need for rigorous initialization or learning rate tweaking.

2.3.3.7 Fully Connected Layer

Every fully coupled neuron in a convolutional neural network layer, also known as a dense layer, is coupled to every neuron in the layer behind it. This layer seeks to learn higher-level features by combining the lower-level features that the preceding layers have acquired. Each neuron in the fully connected layer performs a weighted sum of the inputs after the input is typically flattened into a one-dimensional array of values, followed by an activation function. The output of the completely linked layer is a vector of probabilities showing the likelihood of each class designation. The fully connected layer is often the last layer in a CNN and its output is used for making predictions.

2.3.3.8 Output Layer

The output layer in a convolutional neural network is responsible for producing the final output of the network based on the extracted features from the previous layers. It typically consists of one or more neurons that compute a numerical score or probability for each possible output class. The output layer applies a suitable activation function to the computed score to convert it into a meaningful prediction or decision. Depending on the nature of the task, regression, binary classification or multi-class classification may be employed as the activation function [40].

2.4 Deep Learning and Transfer Learning

Deep learning is a branch of machine learning that models and resolves complicated issues requiring vast amounts of data by using artificial neural networks. These neural networks, which consist of many layers of connected nodes, can learn to recognize patterns and other properties in the data using a procedure known as backpropagation. In fields like audio and image identification, natural language processing and others where conventional machine learning techniques have found it difficult to make significant progress, deep learning algorithms have been extremely effective [29].

Transfer learning is a deep learning method that involves training a new model on a related task utilizing an existing neural network that has already been learnt. Transfer learning is the concept that a neural network can apply the knowledge it gains from addressing one problem to another problem that is closely related [41]. The pre-trained network, also known as the source network, is often trained on a large dataset and has mastered the recognition of a variety of patterns and characteristics.

2.4.1 Approaches to Transfer Learning

The transfer learning process involves adapting the learned representations of the source network to the new target task by fine-tuning the weights of some or all of the layers in the network. This method frequently produces better results than training a model from scratch and can significantly minimize the quantity of data and compute needed to do so [42]. There are several common ways to apply transfer learning in deep learning:

- **Feature extraction:** A pre-trained model is used in this technique to extract features from the input data. The newly created model, which is trained for a particular job, such as classification or regression, is then fed the extracted features.
- **Fine-tuning:** With this approach, a pre-trained model is used and the weights of some or all of its layers are changed in response to a new task. When the new task's structure resembles that of the original task for which the pre-trained model was created, fine-tuning is very beneficial.

- **Pre-training:** This method involves using an existing model that has already been trained as a base to train a new model on a separate but related job. For instance, a new model can be trained on a smaller dataset of images from the medical field using a model that has already been trained on a big dataset of images from the natural world.

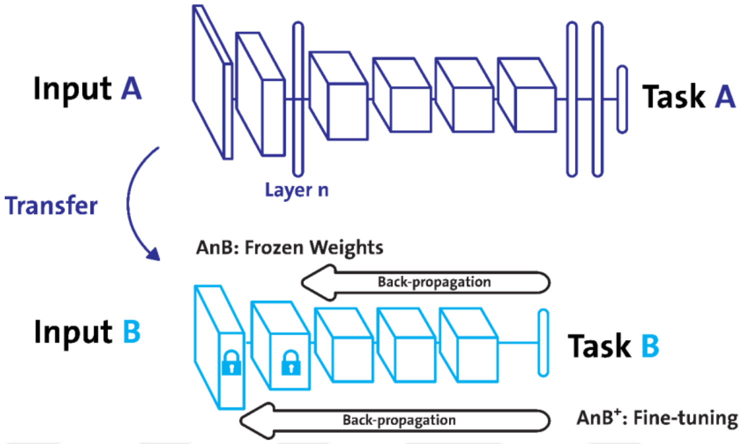
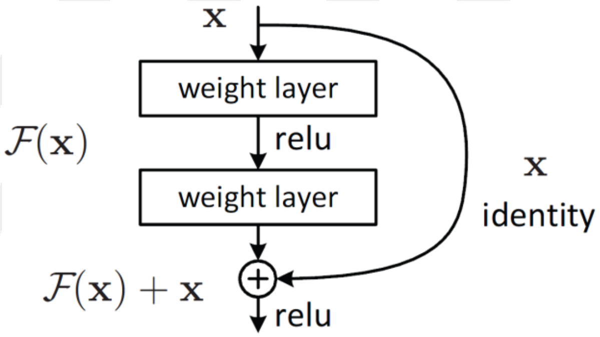


Figure 2.7 Illustration of transfer learning [43]

These approaches can be combined and customized to suit different applications and datasets. Particularly in situations when the amount of accessible data is constrained, transfer learning can be a potent technique for lowering the amount of data and training time necessary to attain good performance on a new task. The popular CNN models used in the transfer learning method and used in this thesis are presented in the subtitles with their detailed explanations and prominent architectural features.

- **ResNet**: ResNet, also known as Residual Network, is a well-known convolutional neural network architecture that has drawn significant attention for a number of computer vision problems. He et al. [44] introduced it, as a result of their image recognition research in 2015. Skip connections, also known as residual connections, were introduced by the ResNet design. By addressing the vanishing gradient problem, these connections enable the training of extremely deep neural networks, which can enhance model performance. ResNet's primary building blocks are residual networks, in which intermediate layers of a block are taught a residual function using the input from the block. Other variations of ResNet exist with varying amounts of layers but the same fundamental concept. The usage of residual blocks, which enables the network to learn identity mappings. As shown in figure 2.8, the residual block also has a shortcut connection that skips one or more levels and has two convolutional layers as part of its construction. Instead of learning the full mapping from scratch, the network can instead learn a residual mapping, which is the difference between the block's input and output. Because of this, the vanishing gradient problem, which appears in deep neural networks when the gradients get too small to spread throughout the network, is mitigated.



x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
x
weight layer
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2.8 Residual block [44]

- **DenseNet:** Short for Dense Convolutional Network, DenseNet is a deep learning architecture that has become well-known for its effective parameter management and enhanced gradient flow. The term was first used by Huang et al. [45] in their 2017 research titled "Densely Connected Convolutional Networks." The well-known DenseNet design decreases the number of parameters, encourages feature reuse, enhances feature propagation and lessens the vanishing gradient issue. Each layer in a thick convolutional neural network has a feed-forward link to every other layer. Every layer in DenseNet receives as additional input the feature maps of all layers that came before it and delivers its own feature maps to all levels that follow, as seen in figure 2.9. As a result, each level below layer n receives n inputs. By down sampling layers, CNN routinely tries to alter the size of the feature map. DenseNet, on the other hand, separates the network into numerous, densely connected sections, enabling feature concatenation and down-sampling. Inside the blocks, the size of the feature map is unaltered. Convolution and pooling are two down-sampling techniques used outside of dense blocks, however concatenation is employed inside of dense blocks since the feature maps inside are all the same size.

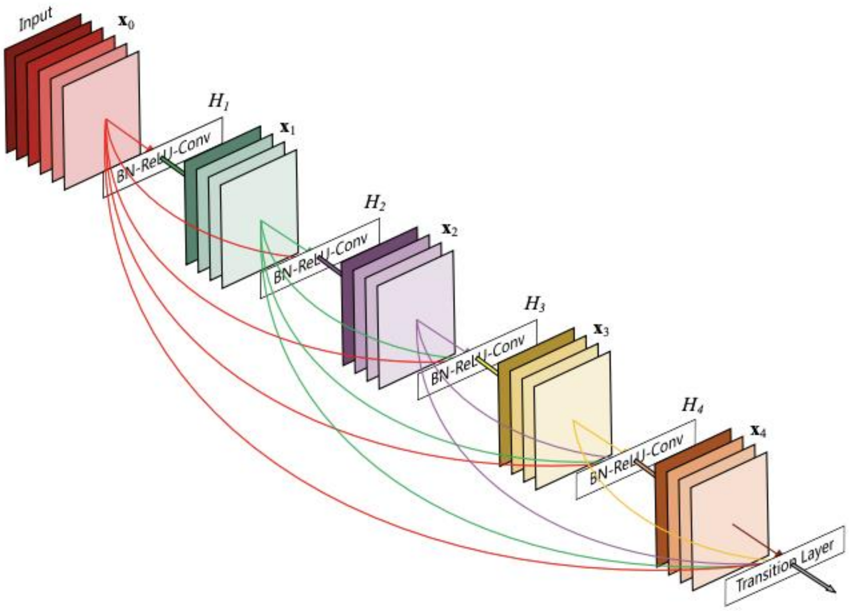


Figure 2.9 5-Layer dense block [45]

- **SE-ResNeXt:** The advantages of ResNeXt and Squeeze-and-Excitation (SE) blocks are combined in SE-ResNeXt, a deep learning architecture. Hu et al. [46] first mentioned it in their study titled "Squeeze-and-Excitation Networks”. ResNeXt is an architecture for convolutional neural networks that extends the ResNet architecture by introducing the concept of "cardinality." The cardinality parameter controls the number of parallel paths that process the input data within each residual block. This increases the model's capacity for representation and enables the network to record a wider range of feature interactions. Each residual block in ResNeXt is made up of a number of convolutional layers, batch normalization and ReLU activation function. The input to the block is processed by a series of simultaneous "transform" layers that perform various convolutions on the data after it has been reduced in dimension by a "squeeze" layer. In order to create the output of the block, the output of the transform layers is lastly "unsqueezed" and added to the initial input. The Squeeze-and-Excitation (SE) ResNeXt architecture is an extension of the ResNeXt architecture, which in turn is an extension of the ResNet architecture. SE-ResNeXt employs a block structure with the addition of a "squeeze-and-excitation" module, as shown in figure 2.10. The weighting of each feature channel is adaptively adjusted by the squeeze-and-excitation module using a channel-by-channel feature recalibration process. It entails of two steps: a "squeeze" step that lowers the feature map's spatial dimension to a single value and a "excitation" stage that models channel dependencies and teaches how to assign relevance scores to each of them. A modular block structure with multiple parallel paths is used to process the input data. However, it also includes the SE module in each block, which enables it to learn more informative feature representations by selectively emphasizing important channels.

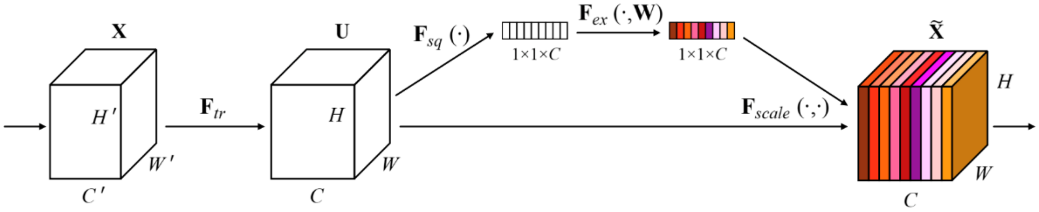


Figure 2.10 A Squeeze-and-Excitation(SE) block [46]

- **ResNeSt:** ResNeSt is a deep learning architecture that was first described by Zhang et al. [47] in their paper titled "ResNeSt: Split-Attention Networks" from 2020. The ResNet architecture, on which Resnest is built, leverages residual blocks to enable the training of extremely deep networks. However, Resnest introduces a new concept called "split attention" to enhance the feature representation. The idea behind split attention is to split the input feature maps into groups, then apply attention mechanism on each group independently and finally concatenate the outputs. The split procedure separates the input feature maps into a number of branches. To capture various types of information, each branch employs a unique set of transformations. Different kernel sizes or dilation rates could be part of these modifications. By executing a weighted total, the merge process mixes the data from the several branches. A channel-wise attention method is used to adaptively learn the weights for the summation. The network can concentrate on key features thanks to this attention mechanism, which gives informative channels higher weights while suppressing irrelevant ones. In addition, In order to better capture objects at various scales, Resnest employs a "repeated multi-scale feature aggregation" technique to capture multi-scale features. This is accomplished by combining features from various network levels while keeping the network's depth and width in balance.

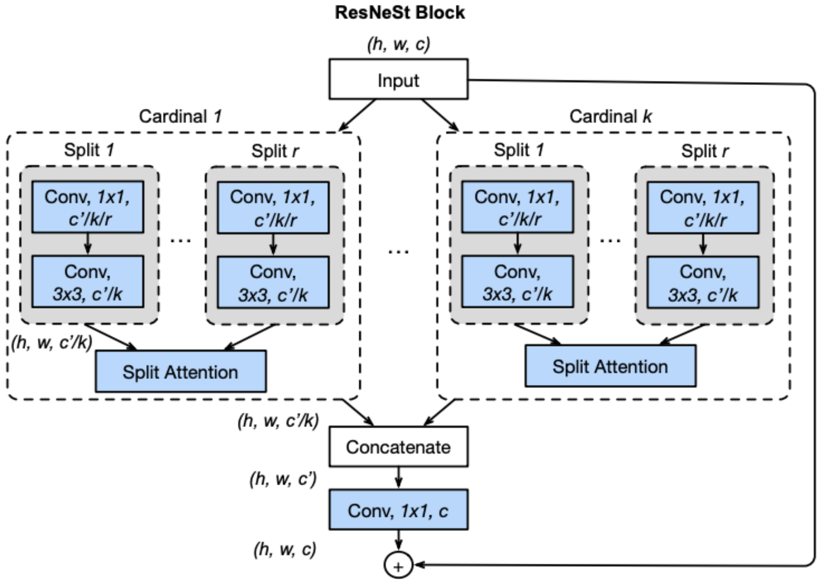


Figure 2.11 ReNeSt block [47]

- **EfficientNet:** EfficientNet is a group of deep learning models that Tan et al. [48] first described in their 2019 publication, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." With regard to various computer vision tasks, a neural network architecture called EfficientNet aims to achieve cutting-edge accuracy while maximizing model efficiency. Model size and computational cost may be balanced with performance because of the architecture's use of a scaling approach that uniformly adjusts the depth, resolution and width of the network. EfficientNet uses a compound scaling technique to scale the network's depth, resolution and width, as shown in figure 2.12. The model's depth is enhanced by including additional layers and its width is increased by including more filters in each layer. By employing larger input images, the resolution is raised. The compound scaling approach is calibrated to strike a compromise between the accuracy and computational cost trade-offs. EfficientNet also introduces a novel compound scaling method for the convolutional layers, called the "mobile inverted bottleneck convolution" (MBConv). The MBConv is made up of an inverted residual block that includes a shortcut connection, a depthwise convolution and a pointwise convolution. The accuracy of the model is maintained as the number of parameters is decreased using the MBConv.

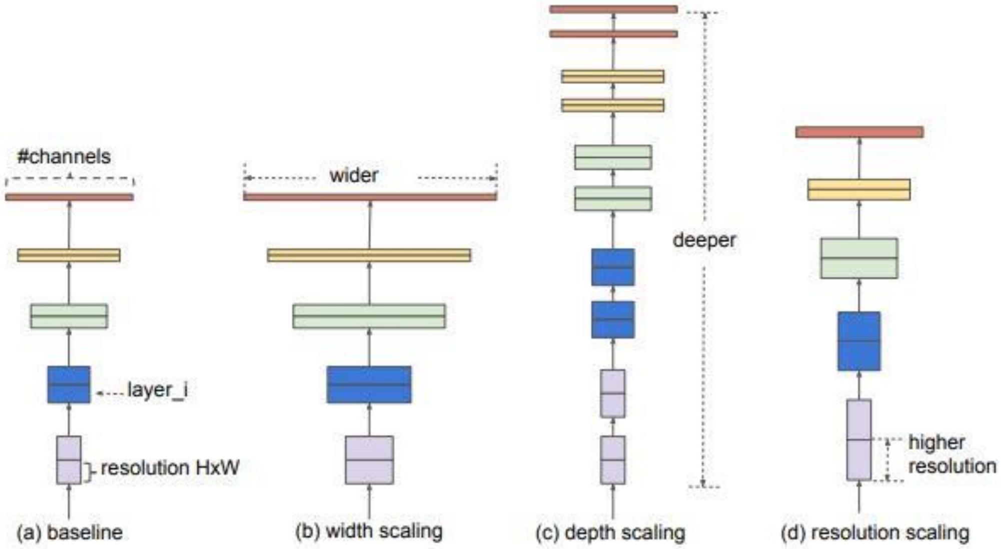


Figure 2.12 Model scaling in EfficientNet [48]

- **TResNet:** A novel family of convolutional neural networks that are optimized for GPU performance is introduced in "TResNet: High Performance GPU-Dedicated Architecture" by Ridnik et al. [49]. Although prior CNNs have had FLOPs (floating-point operations per second) optimized, the authors contend that this does not always result in the greatest performance on GPUs. They suggest several architectural modifications that boost CNNs' GPU performance while preserving or even enhancing accuracy. TResNets have introduced many major architectural changes. A SpaceToDepth stem layer, which is more effective for convolutions on GPUs, transforms the input image into a deeper representation. There contain s many different layers, such as a downsampling layer with anti-aliasing technology that lowers the spatial resolution of the image without aliasing effects, a layer called In-Place Activated BatchNorm that combines the activation and batch normalization procedures. To minimize processes and enhance performance, there offers a novel technique for selecting block types that selects the best possible block type for every tier. It also provides squeeze-and-excitation layers that have been improved to increase network performance without compromising accuracy.

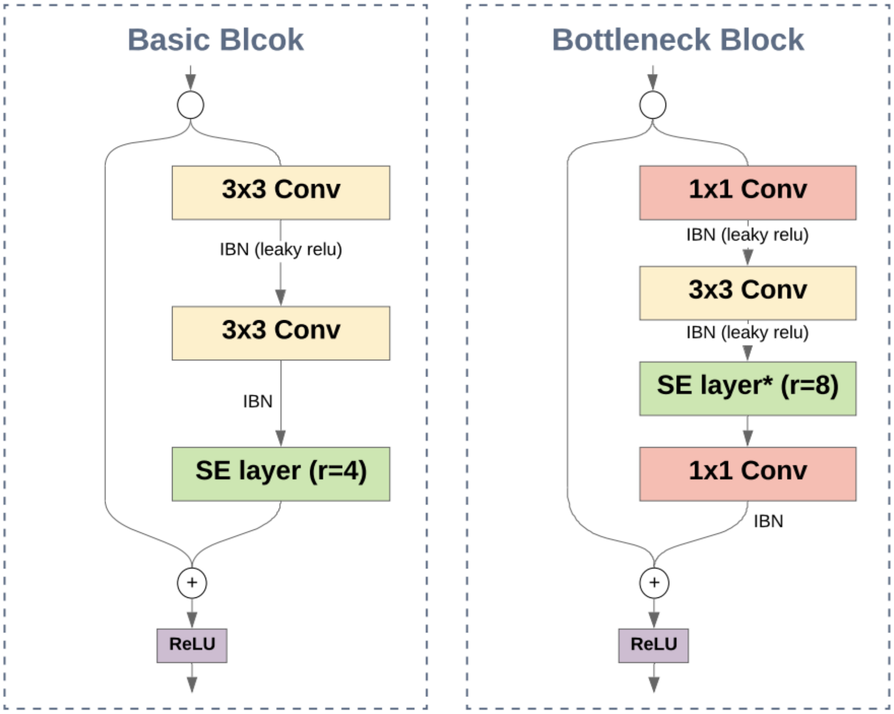


Figure 2.13 TResNet basic block and bottleneck design [49]

- **ConvNeXt:** In the study "A ConvNet for the 2020s" by Liu et al. [50], ConvNeXt is a pure convolutional neural network architecture. They contend that whereas earlier CNN designs were accuracy-focused, this resulted in a number of architectural decisions that made them ineffective and challenging to expand. ConvNeXt uses a hierarchical design, a unique attention mechanism and a progressive training process to address these problems. ConvNeXt's hierarchical structure was inspired by the visual cortex's hierarchy. Each stage in the network learns at a different level of abstraction. In the first stage, simple elements like corners and edges are learned. The following stage involves learning intermediate features, like forms and textures. High-level features, such as objects and features, are learned in the final step. ConvNeXt's innovative attention mechanism enables the network to concentrate on an image's key details. This is accomplished by assigning each characteristic a weight that represents its relative importance for the classification task. The feature maps' spatial and channel dimensions are both subject to the attention mechanism. ConvNeXt's progressive training method enables the network to pick up increasingly complicated features as it is trained.

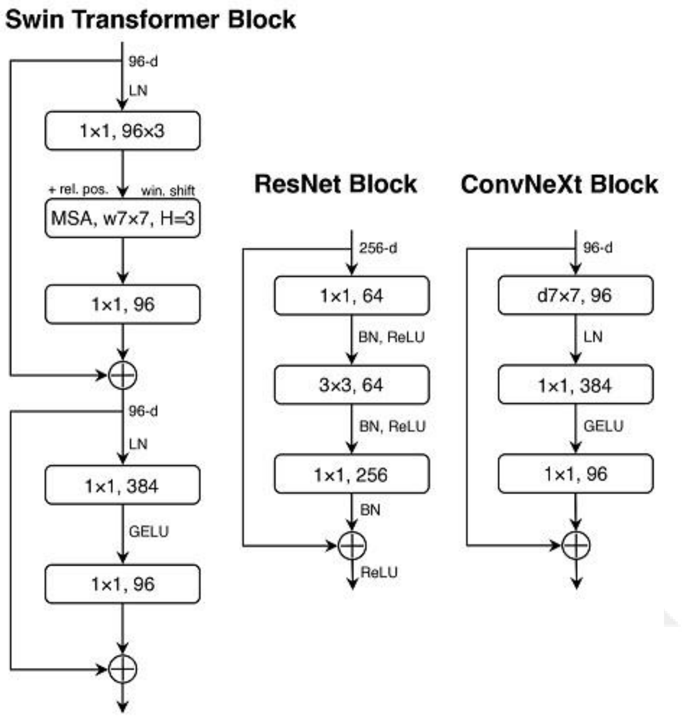


Figure 2.14 Block designs for ResNet, Swin Transformer and ConvNeXt [50]

2.5 Ensemble Learning

In order to create predictions or judgments, ensemble learning, a machine learning technique, combines numerous unique models. The foundation of ensemble learning is the idea that by pooling the predictions of different models, the ensemble can perform better than any one model working alone [51]. A base model or weak learner is the name given to each individual model in the ensemble. Classification, regression and clustering are just a few of the machine learning issues that can be solved with ensemble learning. There are a number of commonly used collective learning techniques, some of which are presented with detailed explanations in the subsections.

2.5.1 Boosting Ensemble

Multiple weak models are trained progressively using the boosting ensemble technique [52], with each succeeding model aiming to fix the errors of the prior models. Boosting combines the predictions of these weak models in an effort to enhance the ensemble's overall performance.The following steps are involved in the process:

- **Model Training**: On the whole training dataset, a weak model is initially trained.
- **Weighted Data:** A weight is given to each instance in the training dataset; the weight is initially set to equal values. The weights show how significant each instance is in the following model training.
- **Iterative Training:** Model training is carried out across a number of iterations. The weights of the incorrectly categorized examples from the prior model are increased with each iteration, while the weights of the instances that were correctly identified are dropped. This enables the succeeding models to concentrate on the challenging instances.
- **Model Combination:** The predictions of all the weak models are integrated using a weighted voting or averaging process. Usually, the weights are chosen based on how well each model performed throughout training.

The main concept behind boosting ensemble is that the ensemble can learn to fix its errors and enhance its predictive performance by training models consecutively and placing greater focus on the instances that were incorrectly classified. In deep learning, boosting has found utility, particularly in the form of algorithms like AdaBoost [53], Gradient Boosting [54] and XGBoost [55]. The use of boosting techniques in deep learning may, however, necessitate careful consideration of computer resources, as training numerous models consecutively can be computationally costly.

2.5.2 Bagging Ensemble

Bagging ensemble, also known as bootstrap aggregating [56], entails building the final prediction by integrating the predictions of many models that were trained independently using different subsets of the training data. The goal of bagging ensemble is to lower variance and boost the models' ability to generalize. The following steps are involved in the bagging ensemble process:

- **Bootstrap Sampling**: The training data are divided into numerous subgroups using random sampling with replacement. The size of each subset, also known as a bootstrap sample, is the same as the size of the initial training data, but it may also contain duplicate instances.
- **Base Model Training:** For each bootstrap sample, a different base model is learned. Convolutional neural networks can be used as base model.
- **Base Model Predictions:** On the validation or test dataset, predictions are made using each base model after training.
- **Ensemble Prediction:** The base model predictions are combined via a process called aggregation. Popular aggregation methods for classification tasks include majority voting, which selects the class with the most support from the basis models.

The main concept behind bagging ensemble is that it can capture distinct patterns and lessen the influence of outliers or noisy samples by training multiple models on various subsets of the data. Improved generalization performance and robustness result from this.

2.5.3 Stacking Ensemble

Stacking ensemble, also known as stacked generalization [57], is a deep learning technique where numerous models, referred to as base models, are trained and their predictions are integrated using a different model, referred to as a meta-model. By learning to successfully integrate their predictions, the stacking ensemble aims to take advantage of the individual models' strengths and enhance overall performance. The following steps are involved in the stacking ensemble process:

- **Training Base Models**: Using the training dataset, several base models are trained. Convolutional neural networks might serve as one of these basic models. Each basic model gains the ability to extract various features and record various facets of the data.
- **Base Model Predictions:** Following training, predictions are made on the validation dataset using the base models. These predictions act as inputs for the subsequent action.
- **Training of a meta-model:** With the input features of the predictions from the base models and the targets of the corresponding ground truth labels, a meta-model is trained. The meta-model gains the capacity to combine predictions from the underlying models to provide final results. The meta-model can be created using any machine learning algorithm, such as logistic regression, support vector machines or even another deep learning model.
- **Ensemble Prediction:** Once the meta-model has been trained, it can be utilized to make predictions on new, unseen data. The fundamental model predictions are generated first and then the meta-model is used to get the final prediction.

The idea behind stacking ensemble is that the meta-model learns to weight the predictions of the base models based on their individual strengths and weaknesses. By doing so, the total prediction performance is improved and the limits of individual models are overcome.

Chapter 3

Literature Review

In the discipline of dermatology, the classification of melanoma is a crucial issue because it entails the identification of skin cancer. In this domain, reliable and accurate melanoma classification models have been developed using traditional methods of machine learning. These techniques often start with the extraction of characteristics from skin lesion images, then use machine learning algorithms to classify the lesions as melanoma or non-melanoma.

The ABCD rule is one of the earliest and most well-known systems for classifying melanoma [58], which uses visual inspection of the skin lesion to identify asymmetry, border irregularity, color variation and diameter. However, this method relies on subjective visual assessment and may not always be reliable. The accuracy of melanoma classification has been enhanced by the development of additional conventional machine learning techniques. For instance, using feature-based approaches, a number of important characteristics, including texture, color and shape have been retrieved from images of skin lesions. Then, these attributes are used to train machine learning systems.

One of the commonly used traditional machine learning techniques for melanoma classification is the Support Vector Machines (SVMs) [59]. SVMs are binary classifiers that can classify data points into two classes by finding the best hyperplane that separates them. Asymmetry, color variegation, border irregularity and diameter are features taken from dermatoscopic images that have been used to diagnose melanoma. In a study by Yuan et al. [60] was used SVM-based texture classification in order to early detect melanoma. They experimented the algorithm using 22 pairs of real skin lesion images and they achived 70% accuracy when 200-feature vectors are chosen from each sample. In another study by

Gilmore et al. [61] was built a dermatologist diagnostic system based on SVM models to detect melanoma. 14 geometrical and color aspects were examined. They experimented using four different kernels: sigmoid, polynomial, RBF and decreasing k-MOD. Using a set of 199 dermatoscopic images (98 dysplastic ,101 melanomas), the best SVM model using the k-MOD decreasing kernel function achieved 89% sensitivity and an AUC of 76%. Another commonly used traditional machine learning algorithm for melanoma classification is Random Forest (RF) [62], which is an ensemble learning algorithm that combines multiple decision trees. The RF algorithm is a classical machine learning method used for melanoma classification. Multiple decision trees are used in the RF ensemble learning technique to increase the model's robustness and accuracy. In a study Janney et al. [63] proposed a machine learning-based approach for the classification of melanoma from dermatoscopic data. The approach used a set of intensity and texture features extracted from 900 dermatoscopic images and then trained a classifier to distinguish between melanoma and benign lesions. Unsharp masking and an anisotropic diffusion filter were used to improve the images. 5 different classifiers were compared, including the Random Forest algorithm. The area under the receiver operating characteristic curve (ROC) was used to evaluate the performance of the classifiers and RF technique classified melanoma substantially better with 93%. Besides these traditional machine learning methods for melanoma classification, other methods have also been used, such as K-Nearest Neighbors (KNN) [64] and Naive Bayes (NB) [65]. A non-parametric classification approach called KNN classifies an instance based on the class of its k-nearest neighbors in the feature space. In a study Kavitha et al. [66] proposed an efficient system that involves classifying data based on textual characteristics and total of 250 dermatoscopic images were experimented. The training set images were used to train Gray level co-occurrence matrix (GLCM) and Speeded Up Robust Features (SURF), which provided accuracy for KNN classifiers of 78.2% and 85.2% for global texture feature extraction and local texture feature extraction, respectively. In another study Linsangan et al. [67] proposed a system in which the lesion was classified into melanoma, non-melanoma and unknown classes following data preparation using Raspberry Pi device that featured segmentation and feature extraction. They collected images from International Skin Imaging Collaboration (ISIC) [68]. Testing was done on 15 images using a KNN classifier and a precision of 86.67% was achieved. Based on Bayes' theorem, Naive Bayes (NB) is a probabilistic classification algorithm. The essential premise is that the features demonstrate conditional independence given the class label. In other words, the value of each feature is considered to be independent of the values of other features, given the known class label. In a study by Balaji et al. [69] performed a novel dynamic graph cut algorithm. They performed skin lesion segmentation, extract texture, color and asymmetry features from a segmented skin region and used a Nave Bayes classifier for skin disease classification. They used ISIC 2017 dataset for testing and achieved 91.2% accuracy for melanoma cases.

However, these traditional ML methods have some limitations in melanoma diagnosis, such as the need for manual feature extraction, sensitivity to feature selection and limited ability to handle large amounts of data. These limitations have led to the development of deep learning (DL) methods, which have shown promising results in melanoma classification based on raw images.

Deep learning methods have shown significant success in melanoma classification. Esteva et al. [70] carried out one of the first research in this field. In order to distinguish skin lesions as malignant or benign using dermatoscopic images. The developed deep learning model performed as well as board-certified dermatologists. In this study, over 130,000 images of skin lesions were used to train a convolutional neural network to classify the lesions as benign or malignant. Subsequently, a number of studies have explored the potential of deep learning for melanoma classification. Adegun et al. [71] developed a system for melanoma classification that utilizes a multi-stage and multi-scale approach. Lesion-classifier, a new technique they introduced, divides skin lesions into non-melanoma and melanoma based on the outcomes of pixel-wise classification. The effectiveness of their approach was evaluated using two widely recognized benchmark skin lesion datasets: ISIC 2017 and Hospital Pedro Hispano (PH2) [72]. The experimental results demonstrated that their method outperformed several state-of-the-art methods. On the ISIC 2017 dataset as well as the PH2 dataset, they attained an accuracy of 95%. Another study by Tschandl et al. [73] evaluated the classification of pigmented skin lesions between human readers and cutting-edge machine learning systems. They introduced ISIC 2018 dataset of 10015 training images and 1511 test images. According to the study, artificial intelligence algorithms were more precise than human specialists. With sets of 30 randomly chosen lesions, the most effective machine learning algorithms averaged 7.94 more accurate diagnoses than the typical human reader and 6.65 more accurate diagnoses than expert readers. Another study by Le et al. [74] using ISIC 2018 proposed a modified ResNet-50 deep learning model. The model's average accuracy was 93% thanks to the tuning and modification of pre-trained model architecture and training methods including focus loss and class-weighting.

In addition to the 2018 dataset, ISIC has recently introduced 2 different datasets. These were the ISIC 2019 [75, 76, 77] and the ISIC 2020 [78] datasets. In a study by Kassem et al. [79] suggested a model that made use of pre-trained GoogleNet [80] models and transfer learning. They put the suggested model's capacity to classify various kinds of skin lesions to the test using the ISIC 2019 dataset. The eight distinct classes of skin lesions were correctly classified using the suggested approach. They had a classification accuracy rate of 94.92%. Another study by Gessert et al. [81] using the ISIC 2019 suggested using multi-resolution EfficientNets in conjunction with significant data augmentation, loss balancing and ensembling approaches. They demonstrated improved performance of models with large input sizes. They also suggested that models that do not make use of all the information included in the images alone can benefit from metadata. Their ensemble optimal method with metadata achieved a sensitivity of 74.2%.

On the other hand, the ISIC 2020 dataset have also been used in many studies for melanoma classification puposes. In a study Karki et al. [82] suggested an ensemble-based method. To enhance the classification performance, a number of augmentation approaches, like hair addition, have been utilized as preprocessing. Test-time augmentation has been found to help the model get the optimal decision by averaging out the errors. The proposed training strategy for the identification of melanoma appears to work better with the depth and width model. The effectiveness of the ensemble models was evaluated using the area under the ROC curve. Using an ensemble of all EfficientNet-B5 models and one EfficientNet-B6 model, they achieved a 0.9411 area under the ROC curve on hold out test data. Another study by Kaur et al. [83] using ISIC 2020 suggested a model called LCNET that proposed a few preprocessing techniques, including image scaling, oversampling, augmentation and created an accurate model for classifying melanoma lesions. They achieved an average accuracy of 90.48% on ISIC 2020 dataset.

Except for the studies that work on the ISIC 2019 and ISIC 2020 dataset separately, there are not many studies in the literature using these two datasets together. In a study Tziomaka et al. [84] suggested an ensemble approach using deep neural networks of various dimensions and activation functions. To handle various image resolutions, multi-resolution EfficientNets were utilized. To further diversify the ensemble technique, the models were used once again in an architecture with a new activation function that considers the metadata. The model with the highest ROC-AUC score, which is 94.04%. Another study that used both datasets together was suggested by Jaisakthi et al. [85]. They suggested an automated technique for classifying skin lesions that makes use of dermatoscopic images and patient metadata. They carried out a variety of investigations using two different transfer learning techniques, such as feature extraction and fine tuning. In the feature extractor technique, the features from the subsequent layers and the contextual data were combined and the LGBM classifiers were trained on them. They were able to acquire an EfficientNet-B6 AUC score of 0.9174 with this model. To further improve the results, they applied a fine-tuning approach that combines the last layers of the pretrained architecture with a simple neural network that accepts contextual data as input. Through the use of this method, they were able to lessen the problem of hyper-parameter tweaking and achieve a higher AUC score for Efficient B6 with Ranger Optimizer of 0.9681.

All things considered, deep learning techniques have demonstrated significant promise for melanoma classification and have produced results with excellent accuracy on sizable datasets of skin lesion images. In addition, it is understood that as the amount of data set trained and more effective models are used, more consistent and stable classification performances emerge. In this regard, especially the datasets made available by ISIC trigger promising developments for the solution of this difficult melanoma classification problem.

Chapter 4

Materials and Methods

4.1 Datasets

4.1.1 The ISIC Archive

ISIC (International Skin Imaging Collaboration) is a global consortium that aims to improve the early detection of melanoma through the use of digital imaging. The project began in 2010 as a collaboration between skin imaging experts, dermatologists and computer scientists from around the world [68].

The ISIC Archive is a publicly available database of skin images collected by the consortium for research purposes. It contains over 50,000 images of skin lesions, including melanoma, acquired using various imaging modalities such as dermatoscopy, clinical photography and confocal microscopy. The images are accompanied by metadata such as age, gender and lesion location, as well as diagnostic labels provided by dermatologists. The ISIC Archive has been used extensively for the development and evaluation of automated melanoma detection algorithms based on machine learning and deep learning techniques. Several studies have reported high performance of these algorithms in detecting melanoma, with some achieving sensitivity and specificity exceeding those of dermatologists [86].

The International Skin Imaging Collaboration (ISIC) organizes an annual challenge to advance the field of dermatology using computer vision and machine learning techniques. The ISIC 2019 dataset [75, 76, 77, 87] focuses on the automated classification of skin lesion images into nine categories: melanoma, melanocytic nevus, benign keratosis, actinic keratosis / Bowen's disease (intraepithelial carcinoma), basal cell carcinoma, dermatofibroma, squamous cell carcinoma, vascular lesion and unknown. The ISIC 2019 dataset is a collection of skin lesion images, which is the largest dataset for skin lesion analysis to date. It consists of 25,332 images, which were collected from a variety of sources, including hospitals, clinics and research centers. The images were captured using a variety of devices, including professional cameras and smartphones and under different lighting conditions. The images were annotated by dermatologists with ground truth labels. The ISIC 2019 dataset consists of 25,331 dermatoscopy images with labels and related metadata, such as the location of the skin lesion and the patient's age and gender. There are eight distinct diagnostic groups in which the labels of the ISIC 2019 dermatoscopy images fall. Melanoma, basal cell carcinoma, benign keratosis, melanocytic nevus, vascular lesion, actinic keratosis, dermatofibroma and squamous cell carcinoma are the specific diagnoses that are contained in the dataset.

The ISIC 2020 dataset [78] include dermatoscopic images and is publicly available dataset that can be used to classify and diagnose skin cancer. ISIC 2020 is the latest iteration of this dataset and was introduced in 2020. The ISIC 2020 dataset consists of 32,542 benign and 584 malignant skin lesions from more than 2,000 patients. Each image in the dataset is accompanied by a set of clinical metadata, including the patient age, gender and the anatomic location of the lesion and an anonymous patient identification number, which enables the mapping of lesions from the same patient. The ROC-AUC score is used as the ranking's evaluation tool and the ISIC 2020 Challenge's objective is to classify benign and malignant tumors. The dataset's benign images fall into one of eight categories (nevus, solar lentigo, cafe-au-lait macule, seborrheic keratosis, lichenoid keratosis, atypical melanocytic proliferation, lentigo NOS and unknown), while all of the dataset's malignant images are diagnosed as melanoma. Notably, there are no examples of basal cell or squamous cell carcinoma in the sample, which limits the issue to melanoma classification.

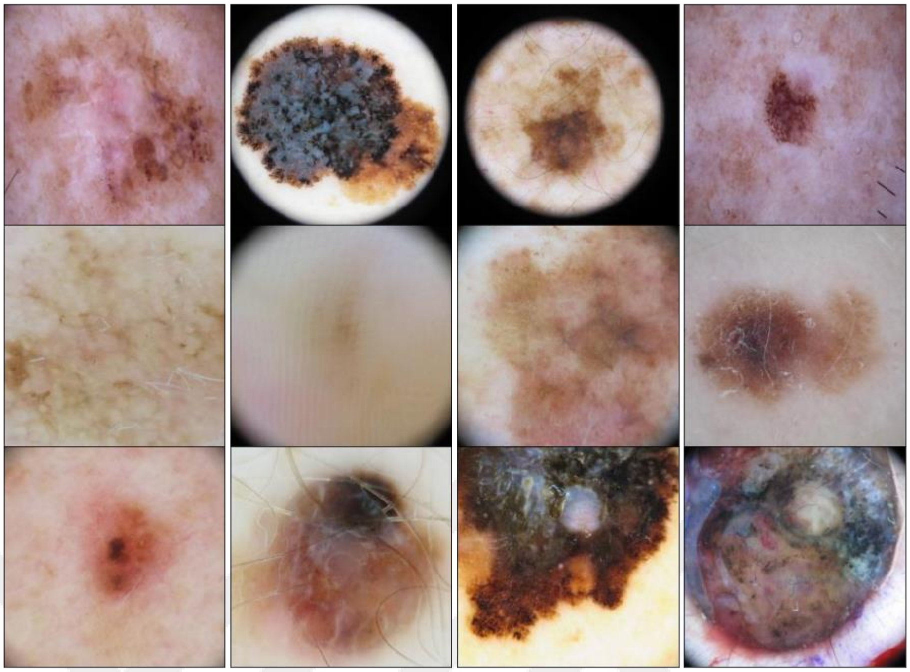


Figure 4.1 Melanoma samples

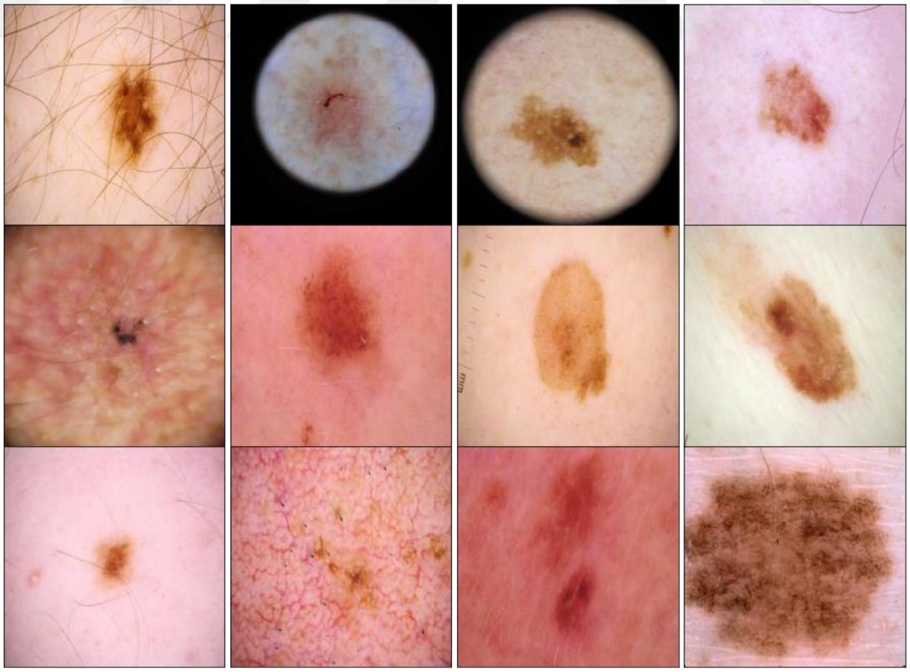


Figure 4.2 Non-melanoma samples

4.1.2 Class Distributions

The ISIC 2019 dataset consists of 9 classes which are Melanoma (MEL), Basal Cell Carcinoma (BCC), Melanocytic Nevus (NV), Benign Keratosis (BKL), Actinic Keratosis (AK), Vascular Lesion (VASC), Squamous Cell Carcinoma (SCC), Dermatofibroma (DF) and none of the others (UNK). Class distributions of the classes given in figure 4.3 with counts.

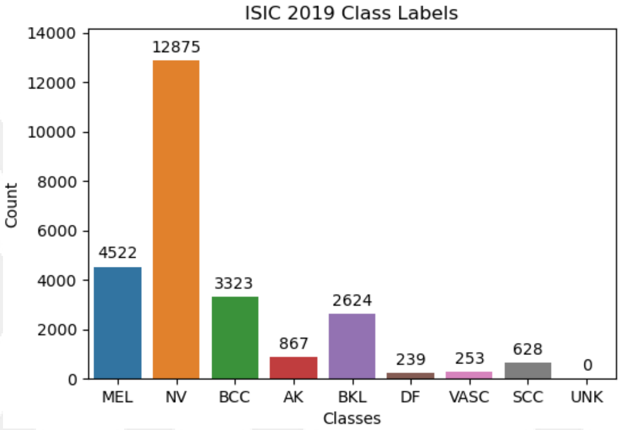


Figure 4.3 Class labels in ISIC 2019 dataset

The ISIC 2019 dataset contains a total of 25331 images, 20809 samples belong to the non-melanoma class, while 4522 samples, which corresponds to 17.85% of the samples, are in the melanoma class.

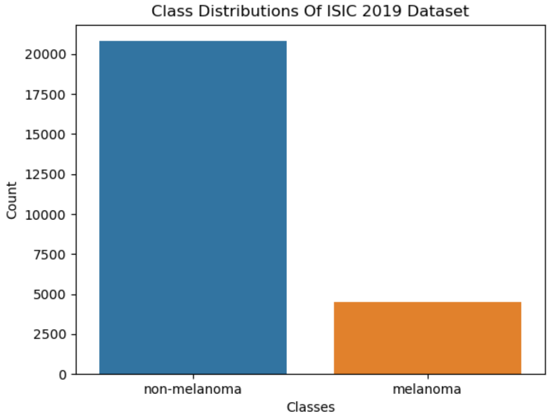


Figure 4.4 Class distributions of ISIC 2019 dataset

The ISIC 2020 dataset contains a total of 33126 images, 32542 samples belong to the non-melanoma class, while 584 samples, which corresponds to 1.76% of the samples, are in the melanoma class.

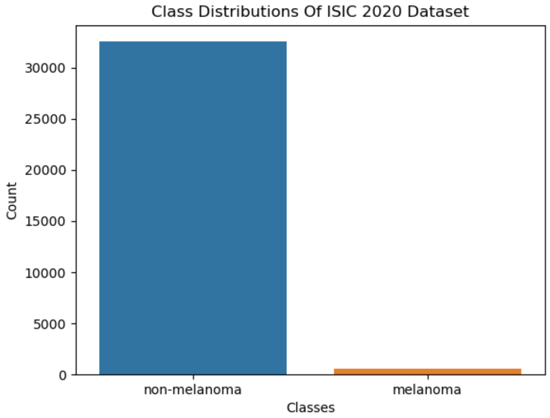


Figure 4.5 Class distributions of ISIC 2020 dataset

After merging two datasets, the final dataset contains a total of 58,457 images. 53,351 samples belong to the non-melanoma class, while 5,106 samples, which corresponds to 8.73% of the samples, are in the melanoma class.

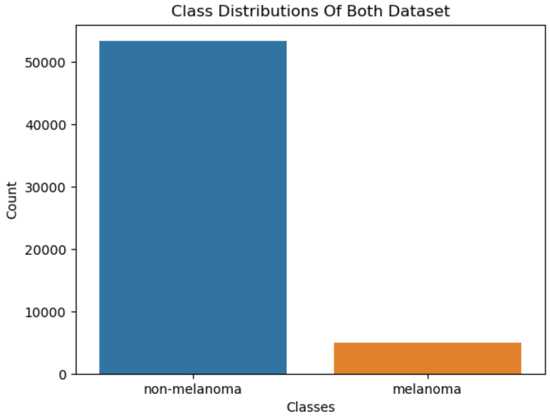


Figure 4.6 Class distributions of both dataset after merging

As shown from the figure 4.6, the final dataset appears to be highly imbalanced.This extremely affects the melanoma bias. In order to get rid of this problem, it is very important to choose the performance metric correctly.

4.2 Proposed Model

There are various stages in the proposed system. Started with the collection of the data given as the first stage. In the next stage, the process continued with the preparation of the data, followed by the data preprocessing stage. The process was terminated with model building and model evaluation. These processes are shown in the figure 4.7 below.

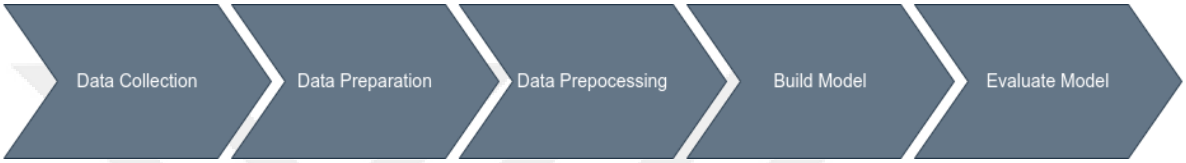


Figure 4.7 Training flow

In terms of flow, the first stage is to specify the issue that the model is meant to address and the kind of data that is needed. The information was gathered from a variety of sources, including open databases, once the data type has been determined. Data was first pre-processed to eliminate noise, deal with missing values and standardize the data. For deep learning models, this phase is crucial since it assures that the data is reliable and consistent. The training, test and validation sets were ultimately constructed using the pre-processed data. The training set was utilized to create the deep learning model, the validation set to modify the hyperparameters and prevent overfitting and the test set to assess the model's performance.

4.2.1 Data Preparation

It was necessary to merge the metadata from the ISIC 2019 dataset with those from the ISIC 2020 dataset. As the "anatomy site general" feature was in both ISIC 2019 and ISIC 2020 datasets, they were categories that could match. For the patient ID feature that is in ISIC 2020 but not in ISIC 2019, all samples in ISIC 2019 were filled with null. Age values given at 5 intervals were also converted into features. With the use of one hot encoding technique, category variables such as age, gender and anatomy site general were turned to binary vectors after the metadata had been combined. To ensure there were no idle features, these transformations were performed to the full dataset, which meant they included the test set that had previously been allocated but excluded it. Following these transformations, the resulting metadata features are:

- **Age :** From 0 to 90 at intervals of 5
- **Gender :** Male,female,unknown gender
- **Site:** Torso, head/neck, oral/genital, lower extremity, upper extremity, palms/soles and none.

4.2.2 Data Preprocessing

Data Preprocessing steps are given in table 4.1 below. The parameters given in Table 4.1 are explained in the subsections.

Table 4.1 Data preprocessing steps

Parameter	**Value**
Image Resizing	224 * 224 * 3
Train-Test Split	85% Train, 15% Test
Stratified K-Fold Cross Validation	5
Standardization and Normalization	[0,1]

4.2.2.1 Image Resizing

Image resizing refers to the process of changing the dimensions of the input image before feeding it into the network. This is typically done to ensure that all images are of the same size and aspect ratio, which is necessary for the network to learn and generalize effectively. All images were resized using the bilinear interpolation method, which computes new pixel values as a weighted average of the surrounding pixels. In particular, the value is determined for each new pixel location by interpolating between the four closest nearby pixels using a weighted average. The weights are based on the separation between the new pixel location and its surrounding pixels [88]. Bilinear interpolation is a simple and computationally efficient technique for resizing images and it is commonly used in convolutional neural networks to prepare input images for processing. By resizing images, the computational requirements of the network can be reduced without significantly sacrificing the accuracy of the model. In this thesis, all images from both dataset resized to 224*224*3. Thus, all images have been resized to a single size with 3 channels.

4.2.2.2 Stratified Train-Test Split

It is essential to have a reliable method of evaluating model performance. One such technique is the train-test split, which divides a dataset into a training set and a test set. The test set is used to assess the model's performance after the training set has been used to evaluate the model. A straightforward and widely used technique is the train-test split, in which a portion of the dataset is randomly chosen for training and the remaining data is utilized for validating [89]. Split process involved the use of stratified technology. A method or procedure known as stratification guarantees that various classes or categories are proportionately represented in the data. The test set assesses the model's performance on new data while the training set is used to optimize the model's parameters. In this thesis, 85-15 split was used where 85% of the dataset is used for training and the 15% is used for testing.

4.2.2.3 Stratified K-Fold Cross Validation

A method for assessing a model's performance on a specific dataset is cross-validation . A form of k-fold cross-validation that is frequently applied to classification issues is stratified k-fold cross-validation [89]. The dataset is divided into k folds of equal size and using stratified k-fold cross-validation, the model is trained and tested k times. Each time, one of the folds is used as the validation set while the remaining k-1 folds serve as the training set. The stratified aspect of the data is produced by ensuring that the proportion of samples from each class in the training and validation sets is nearly equal. This approach is quite useful when working with datasets that are unbalanced and have an unequal distribution of samples by class.

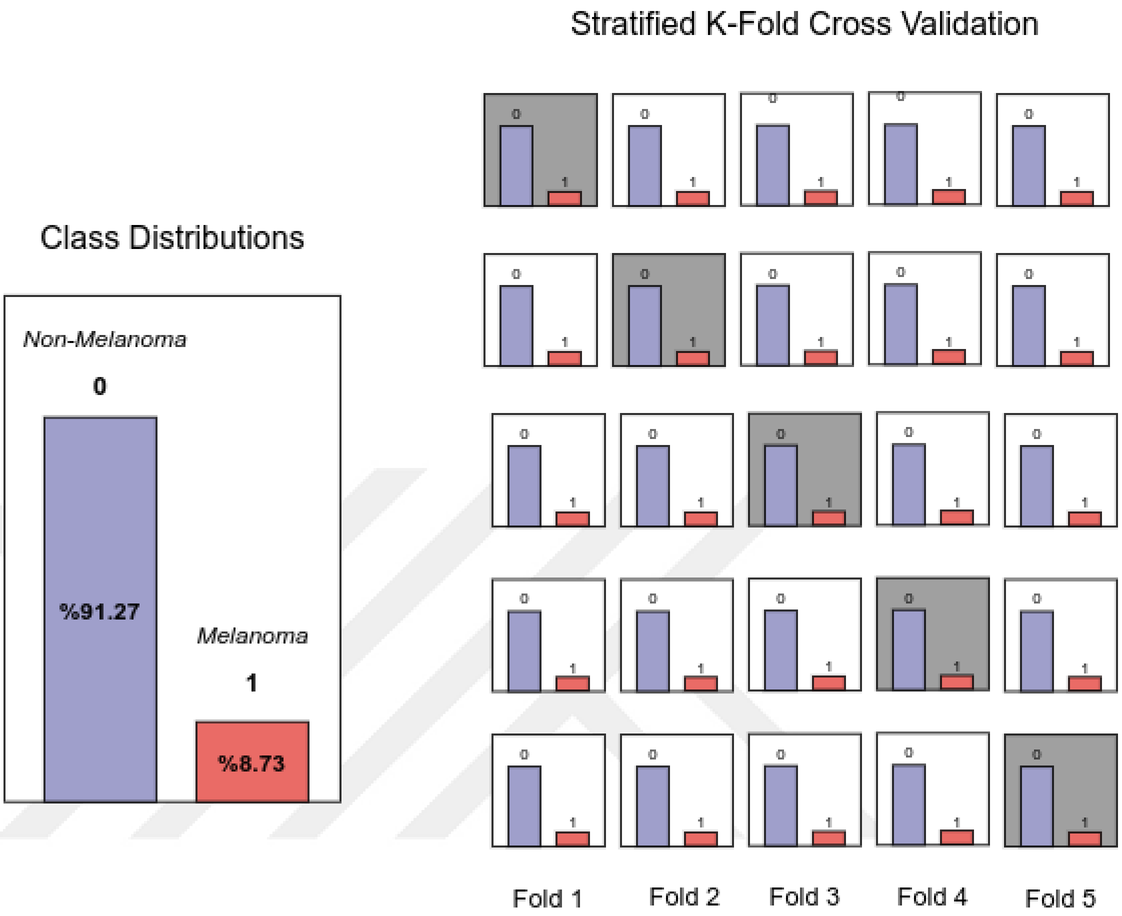


Figure 4.8 Stratified K-Fold Cross Validation

Without stratification, there is a chance that some classes will not be adequately represented in the training set or the validation set, which will produce biased results. Because each sample is used at least once for both training and validation, stratified k-fold cross-validation allows us to obtain a more accurate assessment of the model's performance on unobserved data. This gives a better idea of how effectively the model generalizes to new data and can help to see potential problems like overfitting that may not be obvious when using simply a single train/test split.In this study we took the value of k as 5. That is, 17% of the entire data set was used as a test set at each training stage.

4.2.2.4 Standardization and Normalization

Standardization and normalization are preprocessing techniques used to transform the input data in a CNN before feeding it into the model. Standardization and normalization are applied to make the data features more meaningful, comparable and suitable for model training [89]. The process of standardization entails changing the input data so that the standard deviation is equal to one and the mean is equal to zero. This transformation is carried out independently for each feature in the data, so that each feature has a mean of zero and a standard deviation of one. Standardization helps to center the data and make it more suitable for training neural networks by making the data distribution more symmetrical and reducing the effect of outliers.

Normalization, on the other hand, involves scaling the input data to a fixed range, usually between 0 and 1. This transformation is also carried out independently for each feature in the data, so that each feature is scaled to the same range. Normalization makes the data more consistent and can enhance the model's convergence during training. Standardization and normalizing are frequently combined in practice to preprocess the input data for a CNN. Standardization and normalization work together to improve the input data's uniformity, comparability and suitability for model training.

4.3 Evaluation Progress

In order to achieve the most accurate and stable results, a process consisting of many stages was followed. First of all, it was tried to choose a CNN model and model parameters that are mostly used in imbalanced datasets in general. We compared whether the Shades Of Gray method [90], which we thought would be helpful in data preprocessing, contributed to the Color Constancy stage and generally proceeded with the pros and cons of using different hyperparameter sets specific to the subject. In the solution of such problems, problem-specific parameters other than the generally selected parameters were preferred and the hyperparameter optimization phase was continued. In the last case, considering the pros and cons of problem-specific data augmentation, in order to solve the overfitting problem, two different image processing-based data augmentation techniques were applied separately and together, giving comparative scores in order to get better results. After these stages, performance comparison results were obtained by considering the image features in other CNN models or the features of the model in which the image and metadata were combined. At this stage, 6 models were selected among the 8 models with the highest scores. Finally, the process was terminated with the ensemble method, in which different methods were also used and the best method was chosen by using the predictions of 6 different models selected in order to obtain more stable and better results by using the models together.

As seen in figure 4.9, the process has been tried to be explained with flow decision charts. All models were compared with the AUC metric by taking the average of validation socres in the validation data set prepared with stratified a 5-fold cross validation method. In comparative tables, recall, sensitivity, weighted F1-Score and AUC metrics were presented. Moreover, high scores received in the tables are marked in bold. The following sections was first started with the base model and then, as seen in the figure 4.9, while continuing with each stage, the detailed explanation of the techniques used and the results, as well as the achieved scores were presented together with the previous achieved.

4.3.1 Base Model

Training started with the building of the base model on which performance comparisons would be made in the later time during the experiments. For this, first of all, some hyper parameters had to be kept constant for the building of the model. Some elements are often needed to construct a CNN model for image classification in deep learning, as previously discussed. Using transfer learning with weights pretrained on ImageNet [91] instead of building a new CNN model can be retrained to perform the new task with higher accuracy and less training time.

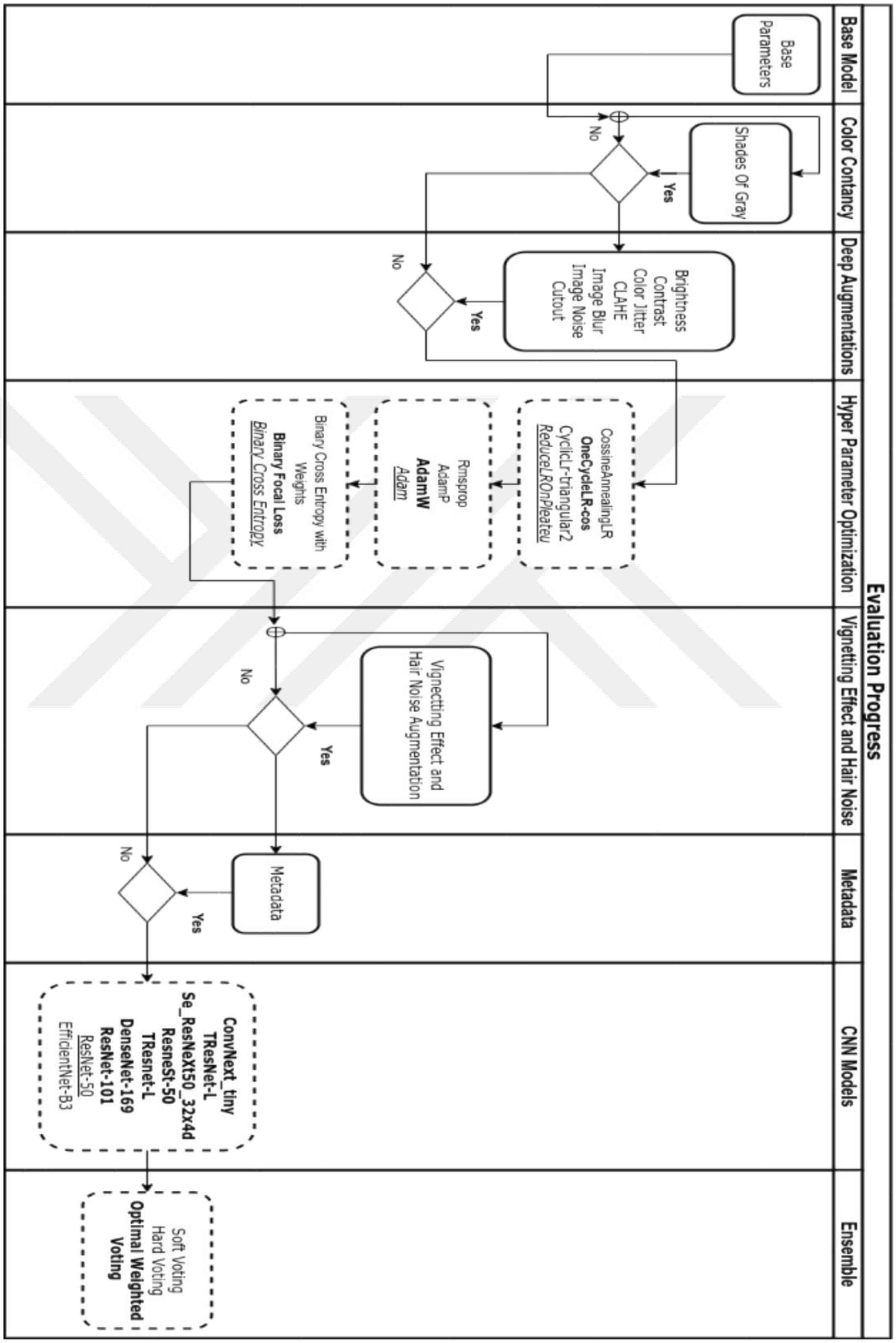


Figure 4.9 Evaluation progress

The parameters used in the base model and their reference values are given in the table 4.2 below.

Table 4.2 Base model parameters

Parameter	Value
CNN Backbone	ResNet-50
Epochs	30
Loss Function	Binary Cross Entropy (BCE)
Learning Rate Scheduler	Reduce Learning Rate On Pleateu
Optimizer	Adam
Learning Rate	0.00001
Early Stopping Patience	5
Batch Size	32

Base model parameters and why they were chosen are explained in separate subsections below.

4.3.1.1 CNN Backbone

We chose ResNet-50 as CNN backbone. ResNet-50, also known as Residual Network-50 with its 50 layers, is a deep neural network that can recognize intricate patterns and hierarchical data structures. Learning more expressive features is facilitated by this depth. The concept of residual connections was proposed by ResNet-50, which helps deeper networks deal with their degradation issue. By allowing gradients to pass straight through the network, these connections solve the vanishing gradient issue and make it possible to train deeper networks. The learned representations from ResNet-50 can be transferred and fine tuned for various computer vision applications, saving time and computational resources.

4.3.1.2 Loss Function

Convolutional neural networks frequently use the loss function Binary Cross Entropy (BCE) to solve binary classification issues [92]. It calculates the difference between the actual probability distribution of binary classes and the anticipated probability distribution. The mathematical definition of BCE is as follows:

$$BCE = \frac{1}{N}\sum_{i=1}^{N} -(y_i \times \log(p_i) + (1 - y_i) \times \log(1 - p_i)) \qquad (4.1)$$

where $\boldsymbol{N}$ is the number of samples, $\boldsymbol{p_i}$ is the predicted probability, $\boldsymbol{y_i}$ is the ground truth label (0 or 1) and **log** is the natural logarithm. Backpropagation and gradient descent optimization methods are used to modify the model's weights and biases in order to minimize the BCE loss during training. Getting the predicted and true class probabilities as close to each other as possible can improve classification performance. Due to its efficiency and simplicity, the BCE loss function is a popular loss function for convolutional neural networks doing binary classification tasks.

4.3.1.3 Optimizer

Adam (Adaptive Moment Estimation) optimizer [93] is a highly regarded optimization technique used in deep learning. It combines the benefits of momentum and RMSProp [94] to improve the weights of the neural network during training. The Adam optimizer has been shown to be effective in converging faster and more accurately than other optimization algorithms [28]. The Adam optimizer computes an exponentially decaying average of previous gradients and previous squared gradients of the weight variables during the training phase and changes the weights of a neural network accordingly. In order to help the optimization method converge more quickly, the momentum of the gradients is also included. The first moment and second moment of the past squared gradients of the weight variables are kept as an exponentially decaying average by the Adam optimizer. The mean and variance of the gradients are both estimated using the first moment and the second moment, respectively. In conclusion, the Adam optimizer tracks an exponentially decaying average of previous gradients and previous squared gradients, respectively, combining the benefits of momentum and RMSProp optimization techniques. The optimizer then uses these estimates to update the weight variables during the training process.

4.3.1.4 Learning Rate Scheduler

Many deep learning frameworks use the ReduceLROnPlateau algorithm, which lowers the learning rate of the optimizer when the loss function stops improving after a predetermined number of epochs [95]. This algorithm can be configured to monitor different metrics such as loss or validation accuracy and can be configured to adjust the learning rate in different ways, such as by a fixed factor or by a percentage of the current learning rate. A CNN's performance can be enhanced by lowering the learning rate on plateau since it will be able to avoid local minima and converge to a better optimum. However, it is important to be careful when using this technique as reducing the learning rate too aggressively can cause the model to converge too slowly or not at all. It is important to monitor the training progress carefully and adjust the parameters accordingly.

4.3.1.5 Basic Image Augmentations

The amount and diversity of training material for CNNs can be artificially increased using techniques called image augmentations. These techniques involve applying various transformations to the input images to create new versions of the same image with slightly different features. By doing so, the CNN can learn to generalize better and be more robust to various types of input variations. In this thesis, three different image augmentation techniques were used as a basis and to be used in the next steps.

- **Transpose:** CNNs frequently use the data augmentation method known as "image transpose augmentation" to enlarge and diversify the training dataset. It involves flipping the image along its diagonal axis, which swaps its rows and columns. By applying image transpose augmentation, the model can learn to recognize the same object or pattern from different viewpoints or orientations. This is particularly useful in image classification tasks where the object or pattern of interest may appear in different orientations or positions in the image. The technique was applied randomly by 50% chance to each image during training.
- **Horizontal Flip:** The input image is flipped horizontally along the y-axis, resulting in a new image that is a mirror image of the original image. The technique was applied randomly by 50% chance to each input image during training.
- **Vertical Flip:** The input image is flipped vertically along the x-axis, resulting in a new image, thus the top of the image becomes the bottom and the bottom becomes the top. The technique was applied randomly by 50% chance to each image.

4.4 Performance Evaluation Metrics

Performance evaluation metrics are used to quantify the effectiveness and accuracy of a machine learning model or algorithm [96]. These metrics are used to compare various models or algorithms as well as to assess how well the model is working on a certain dataset. The choice of evaluation metric depends on the problem domain and the specific goals of the machine learning project. A technique used in machine learning to assess the performance of the model is the confusion matrix [97].

A confusion matrix is used to evaluate the effectiveness of a classification model. On a set of data that was divided into various classes or categories, the matrix shows the number of accurate and inaccurate predictions the model made. The rows of the matrix reflect the actual classes of the data and the columns of the matrix show the anticipated classes by the model. The four cells of the matrix correspond to the results of the classification problem for true positive (TP), false positive (FP), false negative (FN) and true negative (TN) cells. The values in the matrix's diagonal correspond to the wrong classifications (FP and FN), while the values off the diagonal correspond to the correct classifications (TP and TN).

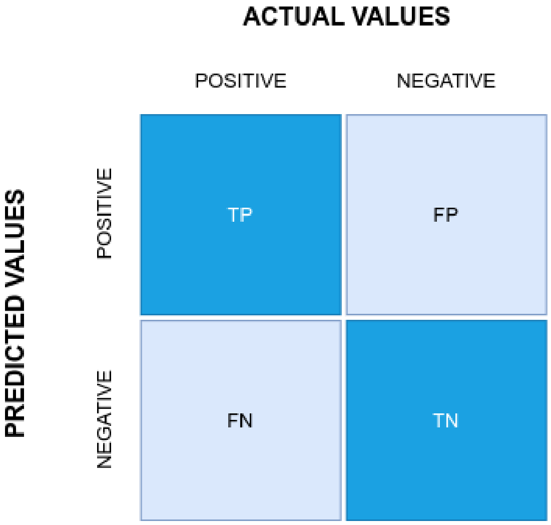


Figure 4.10 Confusion matrix

The terms in the confusion matrix's definitions are listed below.

True Positive (TP): The number of cases that the model accurately recognized as positive but are genuinely positive is known as the true positive value.

True Negative (TN): The number of samples that a classification model accurately predicted as negative and are therefore considered to be true negatives.

False Positive (FP): False positive is a prediction made by a model that claims a positive outcome when in fact the actual outcome is negative.

False Negative (FN): False negatives (FN) refer to instances in a confusion matrix where the model incorrectly predicts the negative class while the actual class is positive.

The confusion matrix, which provides a comprehensive view of the model's performance, can be used to determine F1-Score, recall , precision and accuracy.

Accuracy: In machine learning and classification tasks, accuracy is a performance evaluation metric that assesses the proportion of accurate predictions provided by a model relative to the total number of predictions. It can be mathematically stated as the proportion of accurate predictions to total predictions.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{4.2}$$

In other words, accuracy measures how well a model is able to correctly predict both positive and negative instances. While accuracy is a commonly used metric, it may not be the best measure of model performance in certain scenarios, such as imbalanced datasets, where distinct classes do not have an equal number of instances.

Sensitivity: Sensitivity is a performance statistic used to assess how well a binary classification model is working. It determines the percentage of true positives or the number of positive cases that the model correctly discovered, out of all the real positive cases in the dataset.

$$Sensitivity = \frac{TP}{(TP + FN)} \tag{4.3}$$

In situations where correctly recognizing all positive cases is essential, a high sensitivity means that the model can successfully detect a significant portion of positive cases. High sensitivity, meanwhile, can also lead to a lot of false positives—negative cases that are mislabeled as positive. Sensitivity should be used in conjunction with other measures like specificity and precision for a more full evaluation of the model's performance.

Specificity: Specificity is a performance indicator that gauges how well a model can recognize examples of the negative class. It is calculated by dividing the overall number of instances that are truly negative by the percentage of those that are.

$$Specificty = \frac{TN}{(TN + FP)} \tag{4.4}$$

A model's specificity measures how well it can avoid false positives and correctly identify negative examples. A higher specificity score signifies that the model correctly classifies negative cases as negative and has a low rate of false positives. A larger rate of false positives or the mistaken identification of negative occurrences as positive, is indicated by a lower specificity score, on the other side.

Precision: Precision is a performance evaluation indicator that expresses how many accurate positive predictions a model makes relative to all positive predictions. Or, to put it another way, accuracy is the ratio of true positives to all positive model predictions.

$$Precision = \frac{TP}{(TP + FP)} \tag{4.5}$$

Precision refers to the model's ability to prevent false positive predictions, which are situations in which the model predicts a positive result in error. A high precision score indicates that the model has a low percentage of false positives and is good at correctly predicting outcomes when they are in fact positive. In binary classification tasks, where the objective is to forecast either positive or negative outcomes, precision is frequently used.

Recall: Recall, an indicator used to assess performance, quantifies the percentage of real positive cases that a model correctly identifies as positive. To obtain the calculation, by the total of true positives and false negatives, divide the number of true positives.

$$Recall = \frac{TP}{(TP + FN)} \tag{4.6}$$

Measured by recall, a model's ability to identify each pertinent instance of a given class among all the examples that genuinely fit into that class. A high recall score shows that a model can recognize the majority of the important examples of a class, whereas a low recall score shows that the model misses many important instances.

F1-Score: F1-Score is a metric frequently used to assess the effectiveness of a binary classification model. It generates a single score, which is the harmonic mean of precision and recall and balances the trade-off between these two criteria.

$$F1 - Score = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{4.7}$$

When there is a trade-off between precision and recall and both are significant, the F1-Score is helpful. While a low F1-Score denotes that the model may be biased towards one metric or the other, a high F1-Score indicates that the model is performing well in both precision and recall. It is a helpful indicator for assessing how well models perform in situations when recall and precision are crucial.

Area Under Curve: The area under the curve (AUC) is a performance indicator that is frequently used to assess the effectiveness of binary classification models. The area under the curve is produced by graphing the true positive rate (sensitivity) vs the false positive rate

(1-specificity) at various threshold settings. An increase in the AUC value, which ranges from 0 to 1, indicates improved model performance. The AUC quantifies the probability that a model will rate a randomly selected positive example higher than a randomly selected negative example. An AUC of 0.5 indicates that the model only performs marginally better than guessing at classifications, whereas an AUC of 1.0 indicates perfect performance. Because it is unaffected by changes in the decision threshold, the AUC provides a single scalar number that summarizes the complete performance of the model and is a useful tool for comparing the performance of different models.

Chapter 5

Experimental Studies

In this section, many techniques mentioned in the training flow are presented with their explanations and comparative results. Along with the pros and cons, the order of application of the techniques also reveals the evaluating process. First, visual and comparative results of the effect of Shades Of Gray method with image preprocessing are given. Then, since the addition of data augmentation is used to solve the overfitting problem, techniques compatible with the dataset are preferred and comparative results are given in this section. In the next 3 steps, different methods for step scheduler, optimization and loss function as hyperparameter optimization step are given in the literature, including specific methods for this problem and also discussions. Then, the effect and results of two different image augmentation techniques, vignetting effect and hair noise, were emphasized in order to improve the model in accordance with the data set. The effect of using patients' clinical data and its impact on accuracy were then monitored. The model obtained with the best parameters obtained at these stages was also run with different CNN models and comparative results were given. Finally, it is aimed to further improve the results by using 6 model ensemble methods that show the best performance from 8 different models tested. While giving all these results, precision, recall, F1-Score and AUC values are given in a table comparatively. Again, these tables are based on the base model first and as the steps progress, in addition to the results obtained in that step, the models with the highest accuracy reached until that step are listed at the bottom. In addition to the table, if there is a situation that is expected to be observed in the step, for example, loss plot or figures such as confusion matrix are also presented in order to show the increase and decrease in the number of melanoma samples. Finally, the experimental studies ended with the comparative results of all models and the selection of the models with the highest scores.

5.1 Base Model Performance Results

Base model performance results are given in table 5.1.

Table 5.1 Base model performance results

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Base Model	93.66	94.13	0.9379	0.9290

AUC: Area Under Curve

Base model loss plot for Fold-0, which was used both for the base model and for all performance graphs after that, during training are given in figure 5.1. In addition, classification report is given in figure 5.2 and confusion matrix is given in figure 5.3.

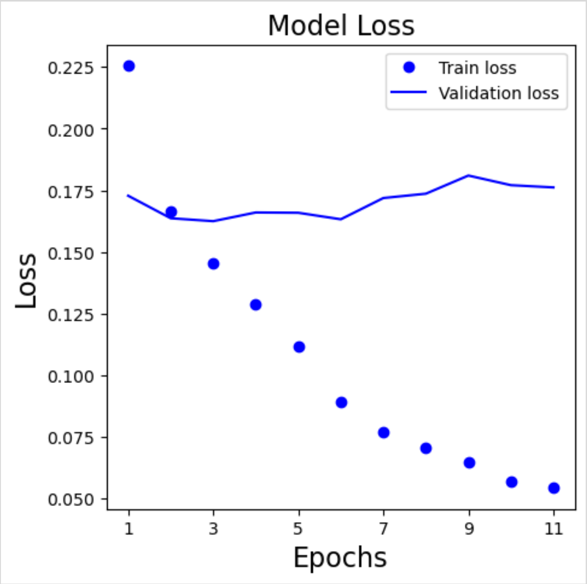


Figure 5.1 Base model loss plot during training

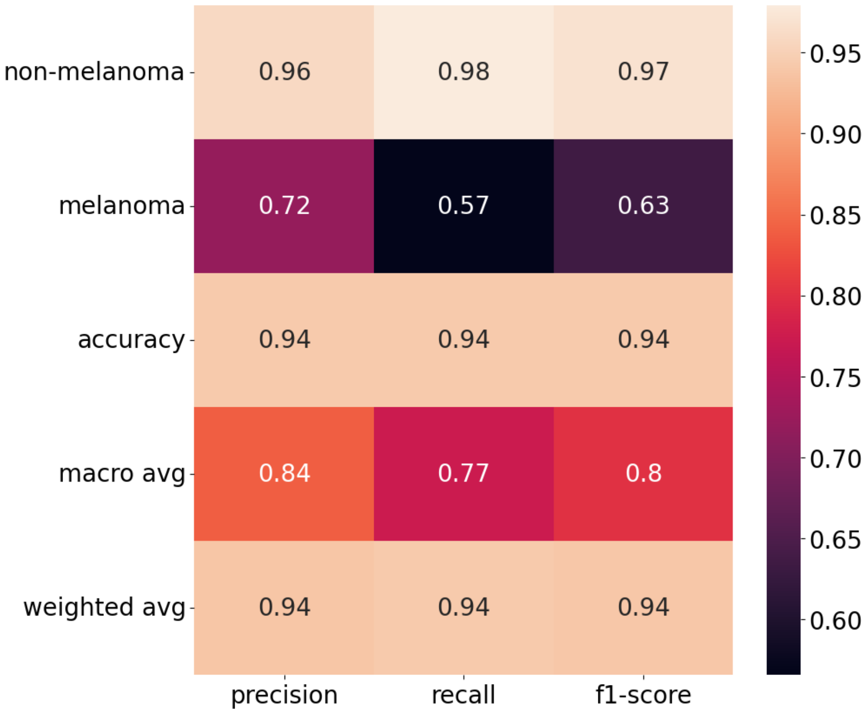


Figure 5.2 Base model classification report

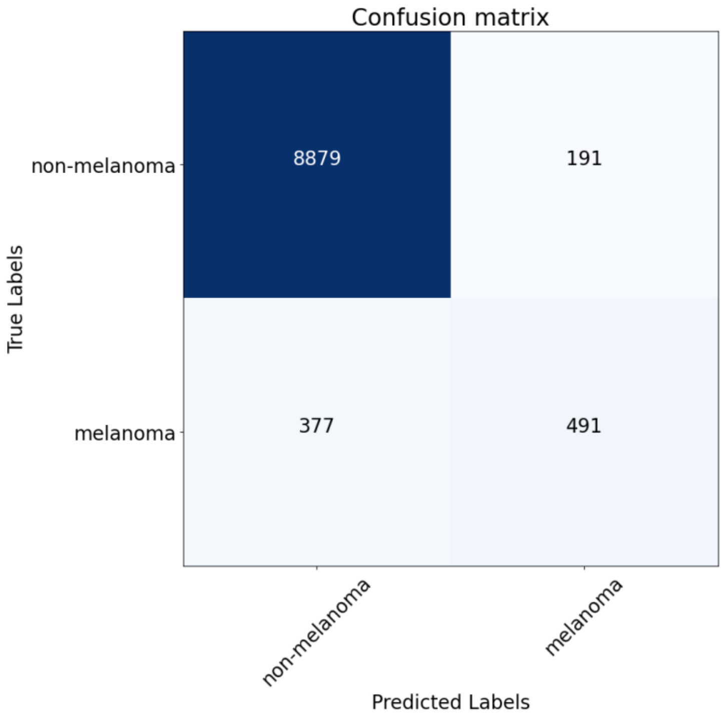


Figure 5.3 Base model confusion matrix

5.2 Step 1 : Color Constancy

The term "color constancy" describes how well the human visual system can distinguish between an object's true colors under various lighting situations. The goal of color constancy techniques in computer vision is to automatically change an image's colors to counteract the effects of illumination so that the perceived colors of the objects in the image are more accurate [98].

There are several techniques for maintaining color constancy, including the white patch, retinex theory-based, gray world assumption and Shades Of Gray techniques [99]. The gray world assumption method uses this assumption to modify the image's colors by assuming that the average color of the entire image is achromatic. The white patch method assumes that the brightest patch in the image corresponds to a white surface and uses this information to adjust the colors. The retinex theory-based methods simulate the neural processes that take place in the human visual system to estimate the reflectance of the surfaces in the image.

Shades Of Gray method is a simple and commonly used color constancy method that aims to correct the color of an image under different illuminations by neutralizing the illuminant's color. This method works by computing the average color of the image, which should be a neutral color under the assumption that the image contains objects that are not colored by the illumination, such as white or gray surfacesA color-corrected image is produced by scaling each of the image's color channels by a factor that equalizes the average value of the three channels.

In order to correct for variances brought on by various imaging devices or acquisition conditions, a technique known as color compensation is employed to modify the color distribution of images. The objective is to standardize the color look across images to increase comparability and the effectiveness of subsequent analytic operations. According to Barata et al. [100], using a color compensation strategy to lessen how the image acquisition setup affects the color attributes that are retrieved from images leads to improved performance for the classification of skin cancer.

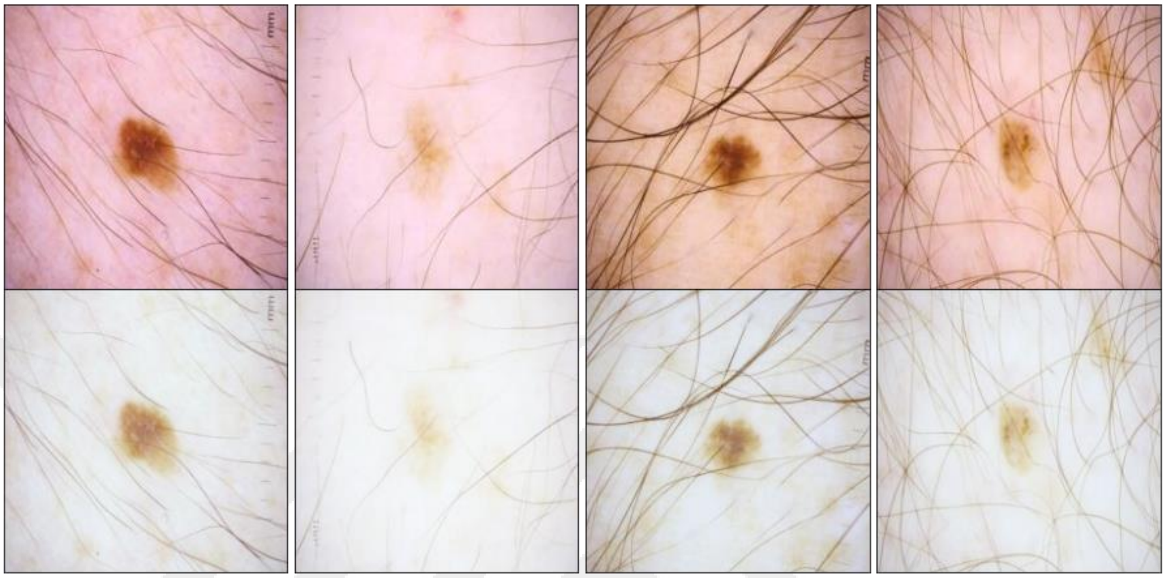


Figure 5.4 Original images at the top, Shades Of Gray method applied images at the bottom

As given in figure 5.4, it is seen how the 4 randomly selected sampling Shades Of Gray method have an effect on the selected images. By normalizing the color look of images, this technique serves to minimize the effects of color differences brought on by various acquisition settings or imaging instruments. This makes it possible to compare images more effectively, allowing for more accurate analysis and interpretation. Applying the Shades Of Gray method also had a significant impact on the accuracy and the comparative results with the base model are given in table 5.2.

Table 5.2 Comparison results of base model and Shades Of Gray algorithm applied

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Base Model	93.66	94.13	0.9379	0.9290
Step 1: Shades Of Gray	93.69	94.00	0.9380	**0.9299**

AUC: Area Under Curve

5.3 Step 2 : Deep Augmentation

By randomly applying these augmentations to training images, the network has a wider range of image variations, making it more robust and capable of better generalization to real-world scenarios [101].

- **Random Brightness and Contrast Augmentations:** Brightness augmentation randomly adjusts the overall brightness of an image, adding or subtracting a constant value from each pixel. This can simulate changes in lighting conditions and help the network learn to recognize objects under different lighting scenarios. The contrast of an image is randomly adjusted via contrast augmentation, which increases or decreases the contrast between light and dark pixels. This can make the network better able to discern small details and features in images and less sensitive to changes in image contrast when performing classification.
- **Color Jitter Augmentations:** Using the Color Jitter technique, the image's saturation, brightness, contrast and hue channels are randomly altered. It can be used to strengthen the model's resistance to changes in illumination conditions.
- **CLAHE Augmentations:** CLAHE is a method for enhancing contrast while keeping the image's overall brightness and color [102]. The CLAHE algorithm works by dividing the image into small rectangular sub-blocks and applying local histogram equalization to each of them. This technique helps to enhance the contrast of the image, especially in regions where there poor lighting conditions or shadows.
- **Image Blur Augmentations:** These techniques simulate the effect of image blur, which can occur due to various factors such as camera motion or defocus conditions. In this part, we used one of three image blur techniques with randomly. The first, motion blur, the technique is often used to simulate the effect of motion in real-world scenarios, such as images captured by a moving camera or images of moving objects. The second technique, known as median blur, involves replacing each pixel's value in an image with the median value of its nearby pixels. The final augmentation method is called Gaussian blur and it entails applying a Gaussian kernel to the image. This convolved and blurred the image by replacing each pixel's value with a weighted

average of its nearby pixels. The level of blur added to the image is determined by the Gaussian kernel's standard deviation.

- **Image Noise Augmentations:** In order to increase the generalization of the model, image noise augmentations include adding random noise to the input image [103]. We applied one of two random image noise approaches. The first is Gaussian noise, which is an additive noise type that adheres to a Gaussian distribution. By assigning random values drawn from a Gaussian distribution to each pixel, this kind of noise can be applied to an image. The second is known as ISO noise that develops when high ISO settings are applied when taking images. It can result in random patterns of pixel intensity variations that affect the overall quality of the image.
- **Cutout Augmentations:** Cutout is an image augmentation technique used in convolutional neural networks to improve the model's robustness to occlusions and increase its generalization capabilities [104]. The technique involves randomly selecting a square-shaped region within an image and replacing the pixel values in that region with zeros. Cutout randomly masks out a contiguous rectangular region of pixels from an image during training, effectively creating a "hole" in the image.

Albumentations [105] library was used to implement augmentation techniques of image augmentations. Because albumentations [106] offers fast and flexible solutions. Probabilities and details of the methods applied during the training are given in table 5.3.

Table 5.3 Augmentation Technical Details

Augmentation	Probability	Parameters
Random Brightness And Contrast	50%	Factor range = 0.2
Color Jitter	50%	Brightness=0.2, Contrast=0.2, Saturation=0.2, Hue=0.2
CLAHE	50%	threshold = (1,4) , Grid Size: (8,8)
Image Blur	50%	One Of (Median Blur(blur limit=5) , Gaussian Blur(blur limit=5) , Motion Blur(blur limit=5))
Image Noise	50%	One Of (ISO Noise(intensity=(0.1, 0.5), color_shift=(0.01, 0.05)) , Gauss Noise(variance range=(5,30))
Cutout	50%	Num_holes=8, max height and width of the hole = 8

Figure 5.5 shows the effects on the 6 selected images from the dataset and the 12 images after applying these images, in order to visually observe how it creates an effect on the images according to the probability and technical details.

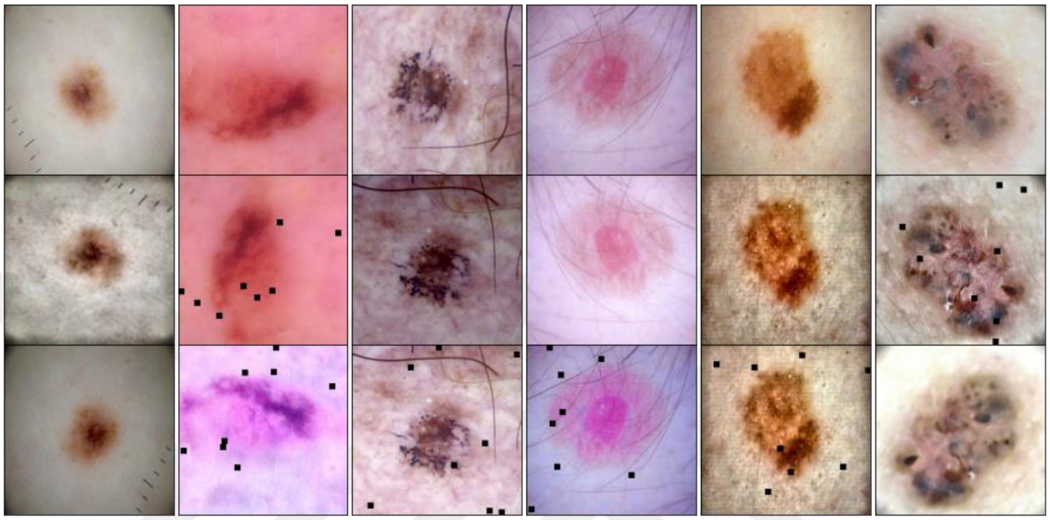


Figure 5.5 Original images at the top, randomly deep augmentations applied images at the second and the third row.

Comparative results obtained by applying deep augmentation techniques are presented in table 5.4. As seen in the table, the AUC value increased from 0.9299 to 0.**9336**. This proves that promising results can be obtained to increase accuracy while providing a difference in the training set during the training phase.

Table 5.4 Comparison results of previous models and deep augmentations applied

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Base Model	93.66	94.13	0.9379	0.9290
Step 1 : Shades Of Gray	93.69	94.00	0.9380	0.9299
Step 2: Deep Augmentations	93.72	94.18	0.9384	**0.9336**

AUC: Area Under Curve

The biggest problem we experienced in this problem was that overfitting occurred as a result of the incredible imbalance between the classes, which was seen on the left of the graphs in figure 5.6 below. In order to provide differentiation in the data set, which is one of the solution methods in order to combat overfitting, the successful effect of using image augmentation techniques in accordance with the data set is clearly seen in the graph on the right of the graphs in figure 5.6.

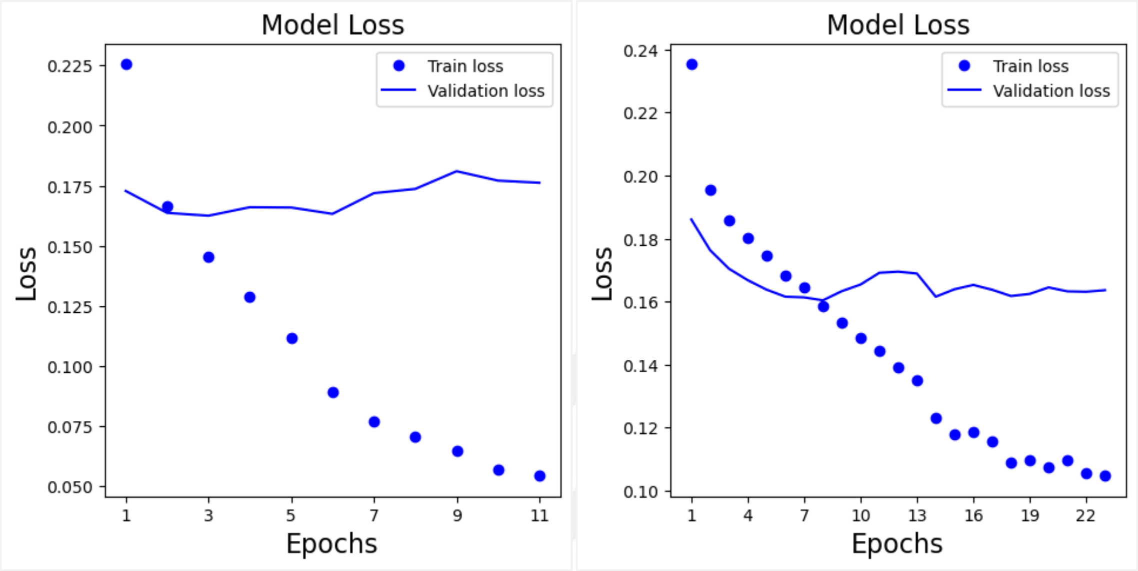


Figure 5.6 Loss plot for base model on the left and deep augmentation applied model on the right

As seen in figure 5.7, confusion matrix shows that true predicted melanoma labels increase from 491 to **511**.

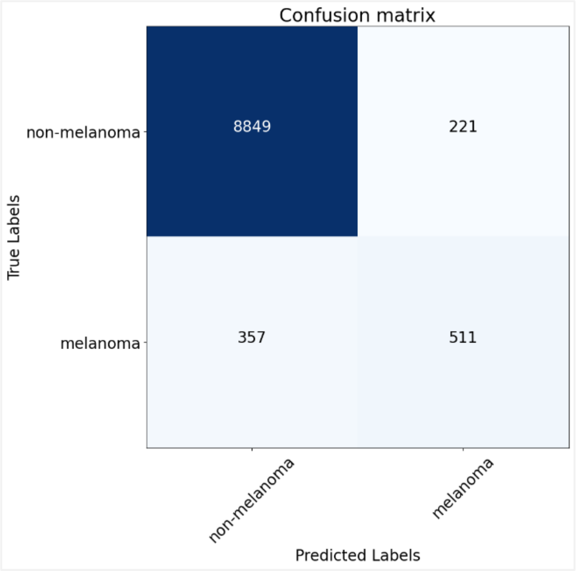


Figure 5.7 Confusion matrix for deep augmentations applied model

5.4 Step 3 : Scheduler

The learning rate is a hyperparameter that controls how much the weights of the model should change while being trained. There are methods for adjusting the learning rate during training to enhance model performance. During backpropagation, it establishes the size of the step that is taken in the gradient's direction. The model may take a very long time to converge to a satisfactory solution if the learning rate is too low. The model may overshoot the ideal answer and fail to converge if the learning rate is too high [107].

Learning rate scheduler is a technique that adjusts the learning rate during training. Learning rate schedulers automatically adjust the learning rate during training, typically by reducing or increasing it over time, to ensure efficient convergence of the model. By doing so, the model's performance can be enhanced and overfitting can be avoided. There are several types of learning rate schedulers, including step decay, exponential decay and cyclic learning rates. In this thesis, we tried CyclicLR-triangular2 [108], CosineAnnealingLR [109] and OneCycleLR-cos [110] learning rate techniques. If these are to be explained more technically,

- **CyclicLR-triangular2 learning rate scheduler:** CyclicLR-triangular2 is a learning rate scheduler used in convolutional neural networks that employs a cyclic learning rate policy. It is based on the idea of cyclic learning rates, which entails changing the learning rate during training in order to hasten convergence and maybe arrive at a more effective solution [108].

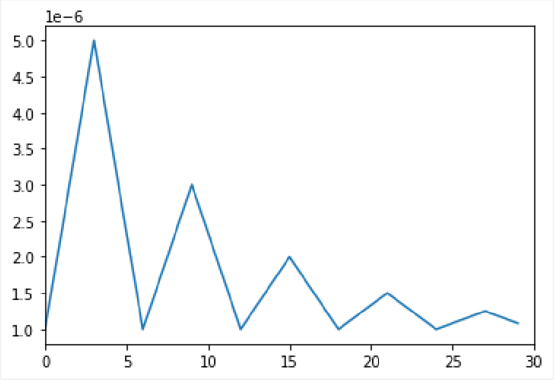


Figure 5.8 CyclicLR-triangular2 learning rate scheduler

As seen in figure 5.8, the learning rate climbs from an initial value to a maximum value in the first half of each cycle for the CyclicLR-triangular2 scheduler and then declines from the maximum value back to the initial value in the second half of each cycle. The scheduler uses a triangular waveform to achieve this cycle, hence the name "triangular2". The effect of using this scheduler in a CNN is that it can help the network converge faster and potentially reach a better solution compared to using a fixed learning rate. The cyclic nature of the learning rate helps the network avoid getting stuck in local optima and can help it explore different regions of the loss landscape. Additionally, the triangular waveform of the learning rate cycle provides a smooth transition between high and low learning rates, which can help prevent the network from making large updates that could destabilize the training process.

- **CossineAnnealing learning rate scheduler:** The learning rate of a convolutional neural network (CNN) is modified during the training process using the CosineAnnealingLR scheduler technique. The backpropagation optimizer's step size for updating the neural network's weights is determined by the learning rate [109].

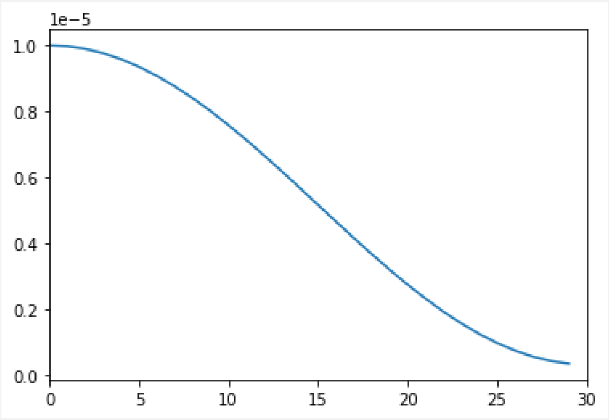


Figure 5.9 CossineAnnealing learning rate scheduler

As seen in figure 5.9, the CosineAnnealingLR scheduler reduces the learning rate in a cosine annealing manner, meaning that it gradually decreases the learning rate from an initial maximum value to a minimum value as the training progresses. This method helps to prevent overfitting and achieve better performance by allowing the network to explore the solution space more effectively. It functions by permitting the learning rate to be larger early in the training process when the weights are further away from their optimal values and then gradually reducing the learning rate as the weights get closer to their optimal values. The CosineAnnealingLR scheduler employs a cosine annealing schedule to smooth learning rate updates and lessen the possibility of exceeding the ideal weights. The number of epochs in a training cycle is a hyperparameter that affects the performance of the scheduler. The larger the number of epochs, the slower the decrease in the learning rate, allowing the network to explore the solution space more thoroughly.

- **OneCycleLR-cos learning rate scheduler:** Convolutional neural networks can use a method called the OneCycleLR-cos learning rate scheduler to regulate the learning rate while training [110]. This scheduler varies the learning rate in a cyclical pattern that consists of three phases: a gradual increase in the learning rate, a gradual decrease and a steep drop towards the end of training. The initial phase aims to quickly converge the model to a good solution, while the second phase enables the model to explore other regions of the parameter space. Towards the end of training, the last step aids in fine-tuning the model. The cosine annealing policy and the One Cycle policy are combined by the OneCycleLR-cos scheduler to adjust the learning rate.

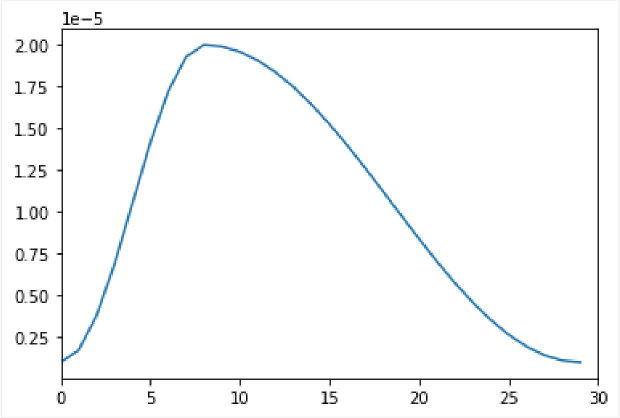


Figure 5.10 OneCycleLR-cos learning rate scheduler

As seen in figure 5.10, the One Cycle policy gradually increases and then decreases the learning rate, while the cosine annealing policy reduces the learning rate gradually towards the end of the training process. The OneCycleLR-cos scheduler provides an optimal learning rate schedule that ensures better convergence and faster training of the CNN. By dynamically modifying the learning rate based on the loss function and the number of iterations, this scheduler helps to prevent the model from being overfitted or underfitted.

Comparative performance results of the base model and CyclicLR-triangular2, CosineAnnealingLR, OneCycleLR-cos learning rate techniques are given in table 5.5.

Table 5.5 Comparison results of previous models and different learning rate schedulers applied

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Base Model	93.66	94.13	0.9379	0.9290
Step 1: Shades Of Gray	93.69	94.00	0.9380	0.9299
Step 2: Deep Augmentations	93.72	94.18	0.9384	0.9336
Step 3a: CyclicalLR-triangular2	93.37	93.99	0.9346	0.9278
Step 3b: CosineAnnealingLR	93.68	94.14	0.9381	0.9346
Step 3c: OneCycleLR-cos	94.00	94.46	0.9411	**0.9350**

AUC: Area Under Curve

As seen in table 5.5, OneCycleLR-cos learning rate scheduler got better AUC than other schedulers. A smooth learning rate schedule is provided by the OneCycleLR-Cos scheduler, which progressively raises the learning rate to a maximum value before gradually lowering it. Having a solid balance between initial learning that happens quickly and fine-tuning at the conclusion of training is made possible by this.

The loss plots of all models obtained during the training are given in figure 5.11.In fact, although there was a perfect match between the validation loss and the training loss with the CyclicalLR-triangular2 learning rate scheduler, the AUC metric value did not show success in the same way. The OneCycleLr-cos learning rate scheduler, on the other hand, seems to be good for the training stage.

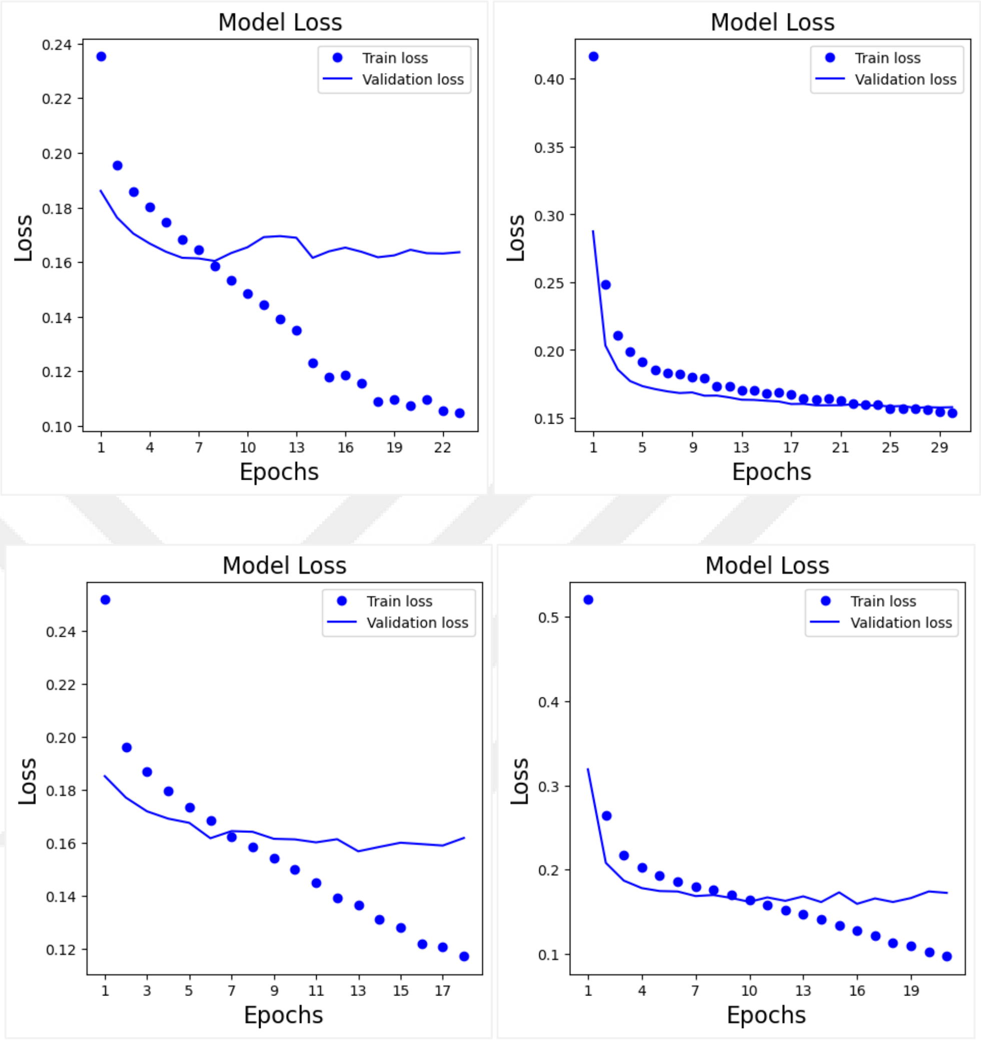


Figure 5.11 Loss plot for deep augmentation applied model on top left, Step 3a: CyclicalLR-triangular2 on top right, Step 3b: CosineAnnealingLR on bottom left, Step 3c: OneCycleLR-cos on bottom right

5.5 Step 4: Optimization

Optimizers are an essential component of the training process in convolutional neural networks. By reducing the loss function, which calculates the difference between the expected and actual output, they assist in updating the model parameters during training. In order to minimize the loss function and have it converge toward the best outcome, the optimizer modifies the model's weights and biases [111]. There are several optimizers used in CNNs, such as SGD [112], RMSprop [94], Adam [93], Nadam [113]. CNN's performance and training can be significantly impacted by the optimizer selected. In this thesis, in addition to Adam, we tried AdamP [114], AdamW [115] and RMSprop optimizers. If these are to be explained more technically,

- **RMSprop optimizer:** RMSprop (Root Mean Square Propagation) is an optimization approach used in deep learning to update a neural network's weights. The technique scales the learning rate using a moving average of the squared gradients. In order to avoid convergence problems, it seeks to prevent the learning rate from being either too high or too low. RMSprop divides the learning rate for each weight by a running average of the magnitudes of the most recent gradients. Due to the ratio of the magnitudes of the most recent gradients in each dimension, the gradient in each is scaled as a result. The technique also has a decay parameter that regulates how much the prior gradient magnitudes are forgotten, allowing it to adjust to shifting data distributions over time. Especially those involving sparse data deep learning applications, have found the RMSprop optimizer to be effective [94].
- **AdamP optimizer:** The AdamP optimizer expands on the Adam optimizer, which combines the AdaGrad's adaptive learning rate with momentum optimization's momentum. AdamP fixes the weight decay issue brought on by the L2 regularization by adding a new penalty term to the Adam optimizer. The weight decay or L2 regularization, which adds a penalty term to the loss function to deter excessive weights, prevents overfitting. L2 regularization, however, might cause weights to be pushed towards zero, which would delay learning and have a negative impact on the optimization process. AdamP addresses this issue by introducing a penalty term that is proportional to the gradient of the weight decay term, which allows the optimizer

to differentiate between the regularization penalty and the true gradient. This technique improves the optimization of scale-invariant weights, which are weights that have the same effect regardless of their scale and which are often used in convolutional neural networks [114]. Overall, AdamP is an extension of the Adam optimizer that improves the optimization of scale-invariant weights by correcting the weight decay problem caused by L2 regularization. It achieves this by introducing a penalty term that is proportional to the gradient of the weight decay term, which helps modify between the regularization penalty and the true gradient.

- **AdamW optimizer:** A variation of the Adam optimizer called the AdamW (Adam with Decay) optimizer is used to update the neural network's parameters while it is being trained. The weight decay term that has been added to the update rule is the primary distinction between AdamW and Adam. By encouraging the optimizer to choose a solution that is more generalizable to fresh data, this weight decay factor helps prevent overfitting [115]. Because the weight decay term is inversely related to the square of the weight values, heavier weights suffer a greater penalty than lighter weights. When training large neural networks with numerous parameters, the AdamW optimizer is very successful since it helps keep the network from overfitting the training data. It is frequently employed in deep learning applications, such as image recognition and natural language processing. In addition to the weight decay term, AdamW also has an adaptive learning rate that modifies the learning rate for each parameter in accordance with the gradient history. In comparison to conventional stochastic gradient descent techniques, this enables the optimizer to converge more quickly and consistently.

As seen in table 5.6, AdamW optimizer got better AUC than others. As seen in figure 5.12 AdamW has been shown to convergence generalization performance. Weight decay is incorporated right into the optimization process by AdamW. This lessens the influence of high parameter values and regularizes the weights of the model, preventing overfitting. It helps the model achieve a suitable balance between fitting the training data and avoiding overfitting by skillfully controlling the learning rates and including weight decay.

Table 5.6 Comparison results of previous models and different optimizers applied

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Base Model	93.66	94.13	0.9379	0.9290
Step 1: Shades Of Gray	93.69	94.00	0.9380	0.9299
Step 2: Deep Augmentations	93.72	94.18	0.9384	0.9336
Step 3c: OneCycleLR-cos	94.00	94.46	0.9411	0.9350
Step 4a: RMSprop	93.68	94.19	0.9371	0.9311
Step 4b: AdamP	93.95	94.32	0.9407	0.9344
Step 4c: AdamW	94.04	94.39	0.9416	**0.9369**

AUC: Area Under Curve

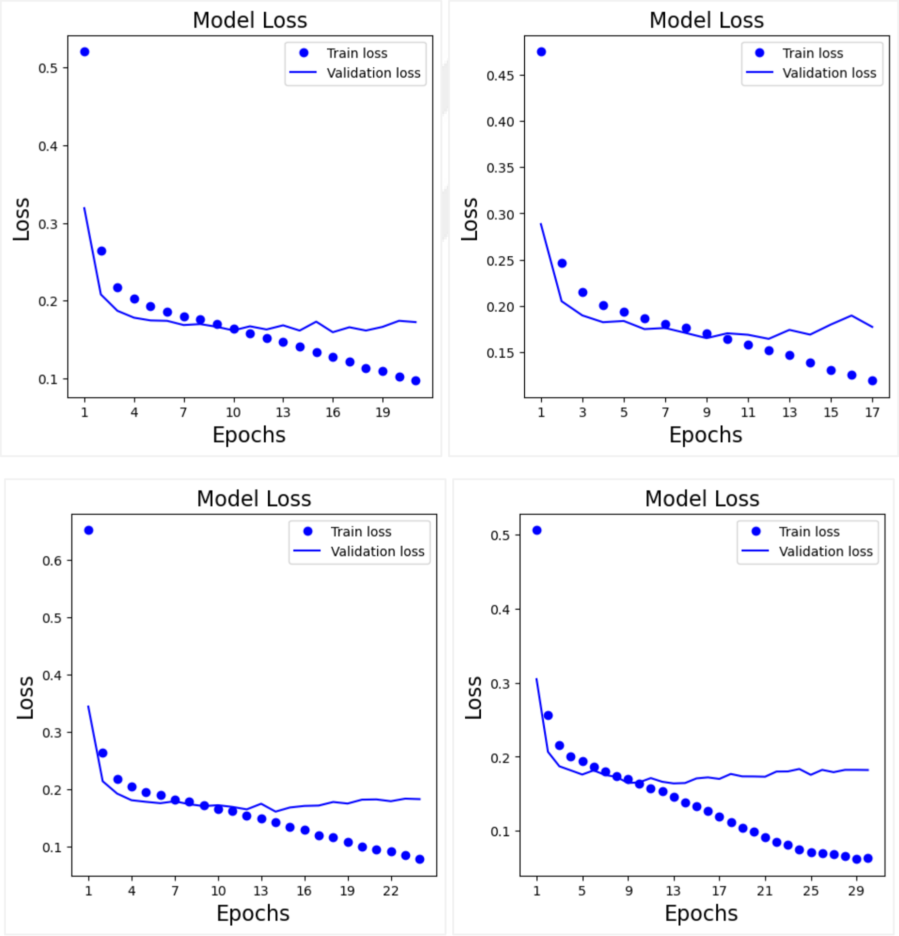


Figure 5.12 Loss plot for OneCycleLR-cos applied model on top left, Step 4a: RMSProp on top right, Step 4b: AdamP on bottom left, Step 4c: AdamW on bottom right

5.6 Step 5: Loss Function

Convolutional neural networks use a loss function to measure the difference between the expected output of the network and the actual output (ground truth). The goal of the model's training process is defined by the loss function. Reducing this loss function is the target of training a CNN, which is accomplished through backpropagation weight changes. In order to quantify how effectively the model is working, it measures the difference between the projected results and the labels from the ground truth. The model learns to predict events correctly by reducing the loss function.

- **Binary Cross Entropy with Weights:** A variant of the binary cross-entropy loss function frequently employed in CNNs for binary classification tasks is binary cross entropy with weights. The model can be influenced to focus more on correctly classifying positive cases by altering the weight allocated to the positive class. When dealing with imbalanced datasets, when one class has noticeably more samples than the other, the Binary Cross Entropy with Weights loss function in CNNs can be helpful [116]. The model can concentrate more on accurately predicting these samples and avoid being biased towards the majority class by giving the minority class larger weights. In addition to addressing the imbalanced datasets issue, this enhances the model's performance for the minority class.
- **Binary Focal Loss:** When performing binary classification tasks, a modified version of the Binary Cross-Entropy Loss function called Binary Focal Loss is widely used, especially when working with datasets that are unbalanced and may have one class that is significantly underrepresented. Lin et al. [117] introduced the Focal Loss function. It seeks to solve the issue of class imbalance by concentrating training on challenging examples that are confidently misclassified. By concentrating on cases that are difficult to classify, the binary focal loss function, a variation of the binary cross-entropy loss function, solves the problem of class imbalance. In order to prioritize the loss on incorrectly classified or challenging cases, it introduces a modulating factor termed the "focusing parameter" to downweight the loss contribution of well-classified examples.

Comparative performance results of loss function are given in table 5.7. By giving misclassified samples, especially those belonging to the minority class, greater weights, the focused loss function addresses how to handle class imbalance. It assists the model in concentrating on difficult cases and successfully balances the effects of several classes. In this sense, its success can be seen from the confusion matrix. In particular, the number of samples found in the melanoma class, which should be noted here, increased from 524 to **547**, which shows that Binary Focal Loss provides a consistent result in imbalance data sets.

Table 5.7 Comparison results of previous models and different loss functions applied

Experimented Model	**Precision (%)**	**Recall (%)**	**F1-Score**	**AUC**
Base Model	93.66	94.13	0.9379	0.9290
Applied with Shades Of Gray	93.69	94.00	0.9380	0.9299
Applied Deep Augmentations	93.72	94.18	0.9384	0.9336
Step 3c: OneCycleLR-cos	94.00	94.46	0.9411	0.9350
Step 4c: AdamW	94.04	94.39	0.9416	0.9369
Step 5a: wBCE	93.30	90.29	0.9135	0.9366
Step 5b: Binary Focal Loss	93.88	94.26	0.9401	**0.9373**

AUC: Area Under Curve

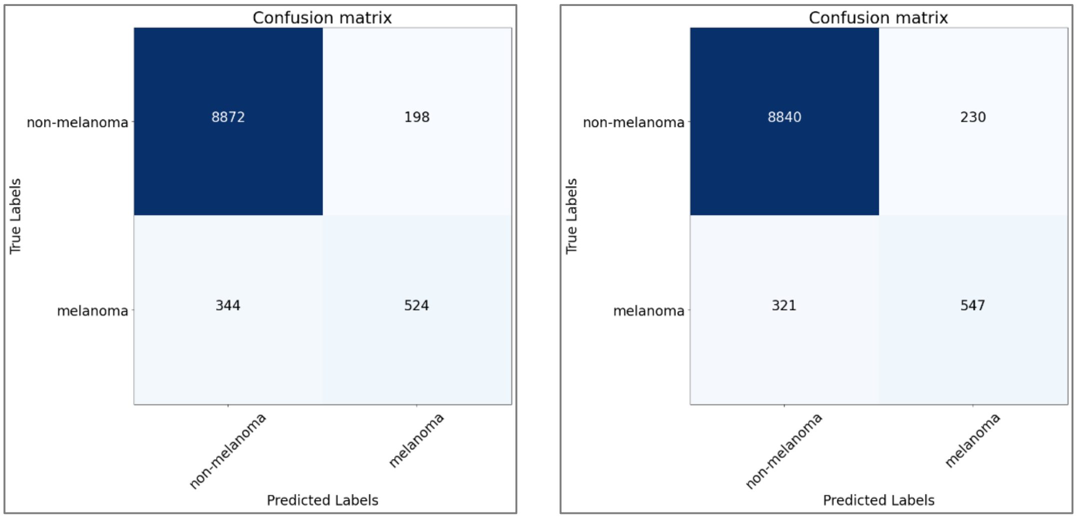


Figure 5.13 Confusion matrix for Step 4c:AdamW applied model on top left, Step 5b: Binary Focal Loss applied model on right

5.7 Step 6: Vignetting Effect

As we were examining the dataset and looking at the images, we noticed that some of the images had black areas around the center circle, as if these images were taken with a microscope, as seen in figure 5.14. But it was also seen that this round frame softened from the corners to the middle, which reminded us of the vignetting effect.

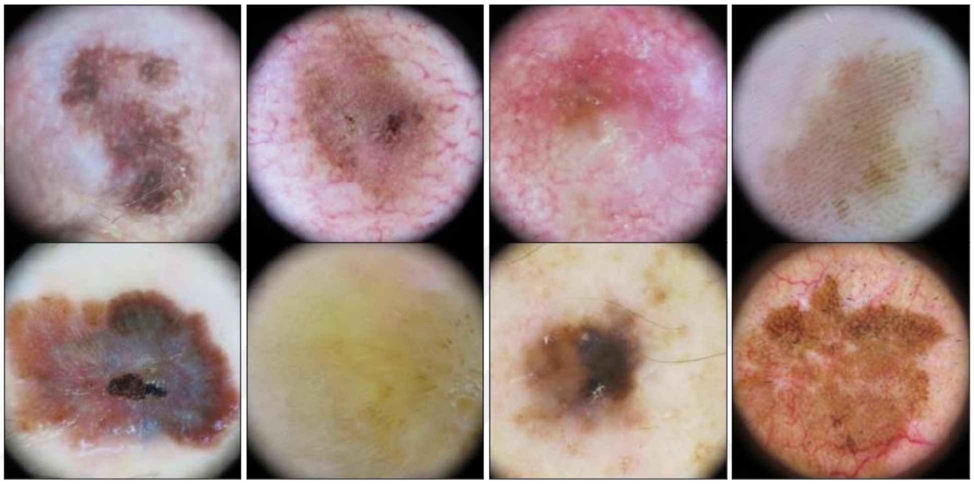


Figure 5.14 Some of the images had black areas around the center circle

When an image's light intensity falls toward its borders, a typical optical distortion known as vignetting results [118]. As a result, the image's edges may appear darker than its center. Vignetting can be obtained in image processing using various techniques, one of which is to apply a Gaussian distribution function. In this technique, a Gaussian distribution function is applied to the image, which simulates the light falloff that occurs in the image due to vignetting. The Gaussian function used in this technique has the following equation:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\ e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{5.1}$$

where $\boldsymbol{f(x)}$ represents the value of the Gaussian kernel at position (x,y) on the 2d image, $\boldsymbol{\sigma}$ is the standard deviation of the Gaussian function, $\boldsymbol{\pi}$ is the mathematical constant, $\boldsymbol{\mu}$ is mean of kernel and $\boldsymbol{e}$ is the exponential function. The Gaussian function is centered at the image center and has a standard deviation that determines the extent of the vignetting effect. The resulted image is created by multiplying the values of the Gaussian function by the image's original pixel values. As a result of the experience, we got the value of sigma as 70. The Gaussian function has a bell-shaped curve, with the peak in the image's center and values dwindling outward from it. When multiplied with the original pixel values, the Gaussian function reduces the intensity of the pixels towards the edges of the image, resulting in a corrected image with reduced vignetting. The original images and the images obtained after applying the vignetting effect filter are shown in figure 5.15.

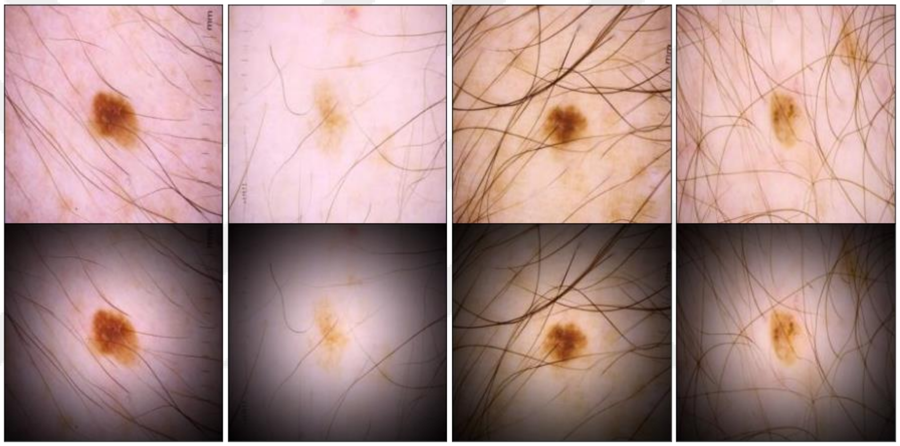


Figure 5.15 Original images at the top, vignetting effect applied images at the bottom

Applying the vignetting effect method also had a significant impact on the accuracy and the comparative results with the base model are given in table 5.8.

Table 5.8 Comparison results of previous models and vignetting effect filter applied

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Base Model	93.66	94.13	0.9379	0.9290
Step 1: Shades Of Gray	93.69	94.00	0.9380	0.9299
Step 2: Deep Augmentations	93.72	94.18	0.9384	0.9336
Step 3c: OneCycleLR-cos	94.00	94.46	0.9411	0.9350
Step 4c: AdamW	94.04	94.39	0.9416	0.9369
Step 5b: Binary Focal Loss	93.88	94.26	0.9401	0.9373
Step 6: Vignetting Effect	93.98	94.43	0.9408	**0.9378**

AUC: Area Under Curve

5.8 Step 7: Hair Noise

Experimental studies have shown that data augmentation techniques helped to achieve better results. They not only diversified the dataset but also improved the results. As we were observing the images in the dataset, we noticed that there are samples with body hair overlapping in the lesion area. There were actually two different approaches here. First, what would happen if these hairs were cleared from the samples. Second or opposite approach, what would happen if, taking into account the loss of information when these hairs are removed, we gathered some samples suitable for hair addition and randomly added these hairs to the images in data augmentation steps.

Hair removal is a crucial preprocessing step in image processing that improves the quality of the images and the precision of future analysis tasks, including the classification of melanoma. Numerous strategies for getting rid of hair have been given in the literature, including thresholding-based techniques, morphological filtering techniques and machine learning techniques [119, 120, 121, 122]. Setting a threshold value to distinguish the hair from the background and then deleting the hair pixels are the steps in threshold-based techniques. Morphological filtering methods use mathematical operations such as erosion, dilation and opening/closing to remove hair and other small objects in the image. Machine learning-based methods use algorithms such as SVMs or CNN to identify and remove hair pixels based on their features and characteristics.

One of the most popular techniques for hair removal is the "dull razor technique," which involves eliminating body hair using a dull razor blade [123]. This method has been applied as a pre-processing step to enhance the classification accuracy of skin lesions in melanoma diagnosis. The basic idea behind using this technique is to remove hair from the skin surface, which can cause artifacts and interfere with the accurate detection of melanoma lesions. In the study by Alizadeh et al. [124] was used the dull razor technique to remove hair from skin lesion images in the ISIC dataset. By using the dull razor technique, the authors were able to achieve a higher accuracy in melanoma classification compared to when the images were not pre-processed in this manner.

Improvement in classification is expected thanks to the use of dull razor technique, which is used in the image preprocessing stage and aims to improve the image and most of the hair noise is removed. The image needs to be preprocessed, with the majority of the human hair-related noise that it contains being eliminated.

The first stage in the dull razor process is to conduct a dilation operation, then an erosion phase, to reduce tiny details in the image. The difference between the original and processed images is then calculated. It uses an erosion process on the variance mask to lessen noise. In order to complete the process, the noise mask is used to replace the original image's pixels. Performance results by using this technique are given in the table 5.9 below.

Table 5.9 Comparison results of previous models and applied with dull razor technique for hair removal

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Base Model	93.66	94.13	0.9379	0.9290
Step 1: Shades Of Gray	93.69	94.00	0.9380	0.9299
Step 2: Deep Augmentations	93.72	94.18	0.9384	0.9336
Step 3c: OneCycleLR-cos	94.00	94.46	0.9411	0.9350
Step 4c: AdamW	94.04	94.39	0.9416	0.9369
Step 5b: Binary Focal Loss	93.88	94.26	0.9401	**0.9373**
Step 7a: Hair Remove	93.59	94.13	0.9370	0.9333

AUC: Area Under Curve

Although an improvement in accuracy was expected with dull razor technique for hair removal, contrary to expectations, AUC score was not improved. There could be many reasons to explain this result. The most important conclusion to be drawn should be, there were significant pixels in the cleared areas. In other words, as a result of performing an erosion process from the image, removing an area under and around the hair that is related or related to the disease, depending on the mask size, had a negative effect on accuracy. Comparative images of how the hair removal process affects the images and which masks are removed and the hairs are cleaned are shown in the figure 5.16 below.

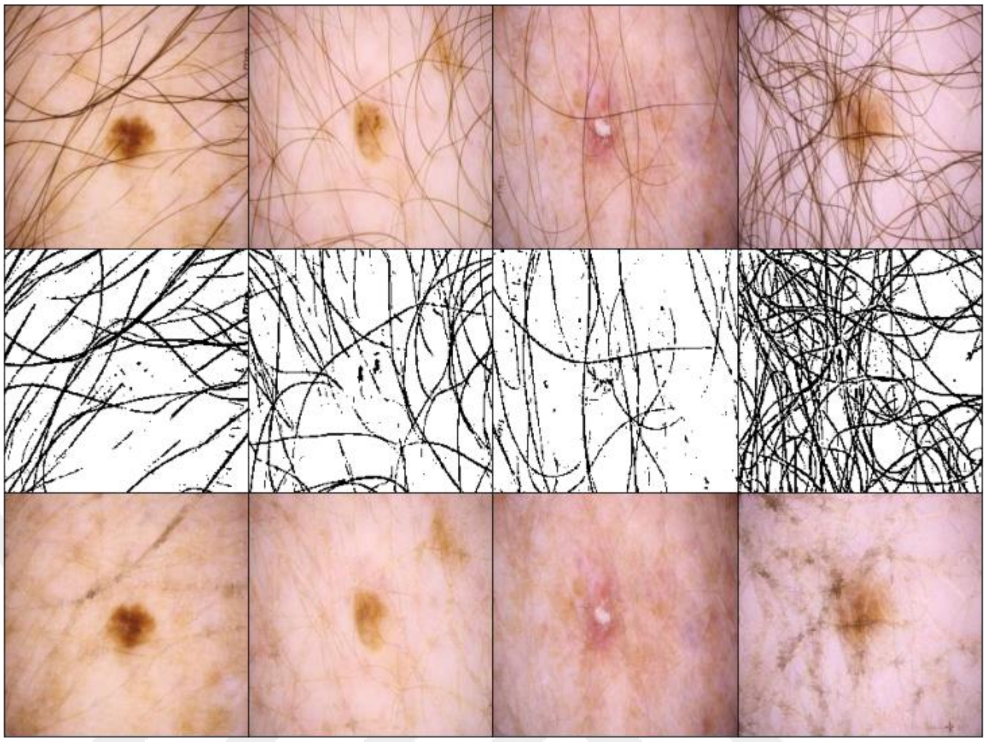


Figure 5.16 Original images at the top, mask images in the middle, hair removed images at the bottom

When the dataset is examined, it is seen that in some images the hairs are less pronounced or more, while in some images the hairs are more prominent and take up more space in the image. If successful results were not obtained in hair removal, then we continued the work by suggesting what would happen if we used some hairs as masks and randomly applied the masks to the images as a data augmentation technique. This time, an experimental investigation was done using a data augmentation technique to randomly add hair noise to the images rather than eliminating hair. The study's findings are shown in table 5.10 and the table below includes a comparison with hair removal. As a result, hair noise data augmentation technique was be used in the following stages, since adding hair noise rather than cleaning hair has a positive effect on performance.

Table 5.10 Comparison results of previous models and random hair noise augmentations applied

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Base Model	93.66	94.13	0.9379	0.9290
Step 1: Shades Of Gray	93.69	94.00	0.9380	0.9299
Step 2: Deep Augmentations	93.72	94.18	0.9384	0.9336
Step 3c: OneCycleLR-cos	94.00	94.46	0.9411	0.9350
Step 4c: AdamW	94.04	94.39	0.9416	0.9369
Step 5b: Binary Focal Loss	93.88	94.26	0.9401	0.9373
Step 7a: Hair Remove	93.59	94.13	0.9370	0.9333
Step 7b: Hair Noise	93.83	94.32	0.9395	**0.9379**

AUC: Area Under Curve

Comparative images of how the hair noise process affects the images and which masks are used for augmentations and how to augment images are shown in the figure 5.17 below.

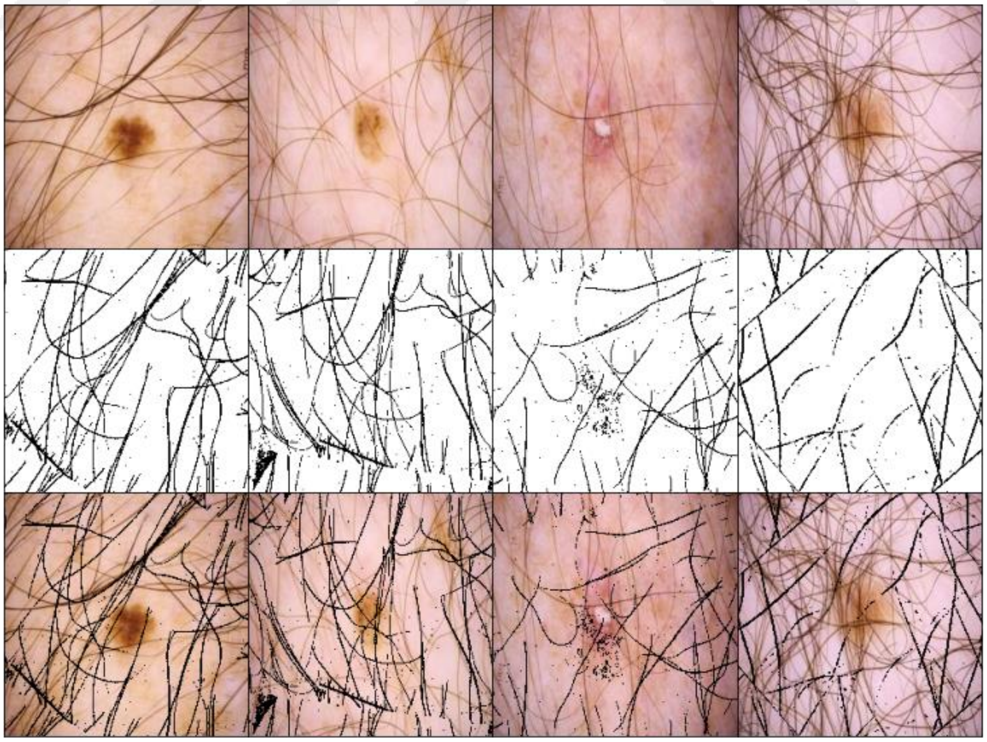


Figure 5.17 Original images on the top panel, mask images in the middle and images with hair noise added in the last row

5.9 Step 8: Vignetting Effect and Hair Noise

Experimental studies have shown that using vignetting effect filter and using hair noise data augmentations has reached better scores than the previous ones. For this reason, we wanted to see the effect of using both techniques in addition to data augmentation techniques and using these two techniques together on images randomly.

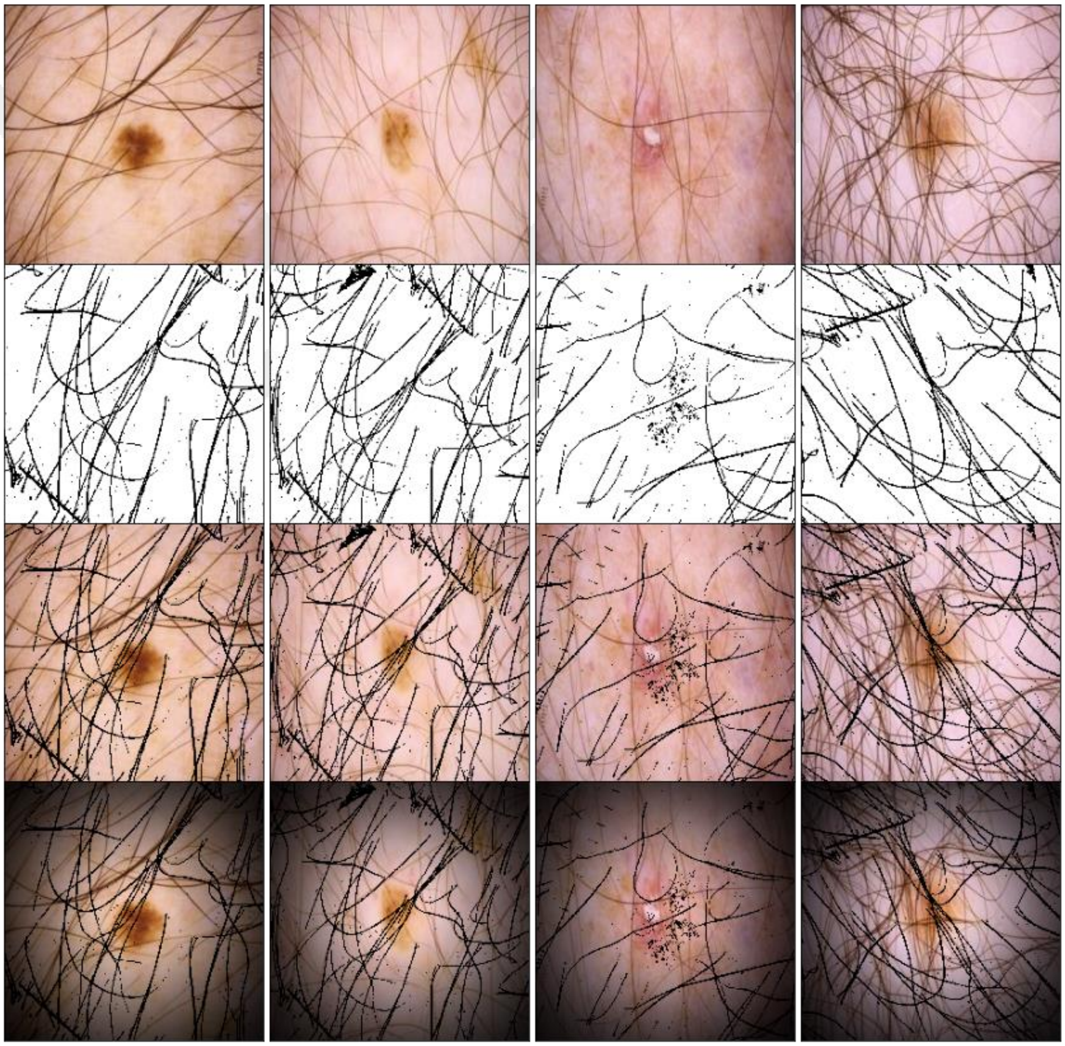


Figure 5.18 Original images in the first row, mask images in the second row, images with hair noise added in the third row and images with hair noise and vignetting effect added in the fourth row.

Comparative scores are given in table 5.11 and as can be seen from the table, using both techniques also stands out as an increase in the AUC metric.

Table 5.11 Comparison results of previous models and vignetting effect filter with random hair noise augmentations applied

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Base Model	93.66	94.13	0.9379	0.9290
Step 1: Shades Of Gray	93.69	94.00	0.9380	0.9299
Step 2 : Deep Augmentations	93.72	94.18	0.9384	0.9336
Step 3c: OneCycleLR-cos	94.00	94.46	0.9411	0.9350
Step4c: AdamW	94.04	94.39	0.9416	0.9369
Step 5b: Binary Focal Loss	93.88	94.26	0.9401	0.9373
Step 6: Vignetting Effect	93.98	94.43	0.9408	0.9378
Step 7b: Hair Noise	93.83	94.32	0.9395	0.9379
Step 8:Vignetting Effect and Hair Noise	93.85	94.23	0.9396	**0.9381**

AUC: Area Under Curve

5.10 Step 9: Metadata

Features of images and metadata fusion is the process of merging data that has been taken from both the visual information included in images and the related metadata. This fusion strategy tries to improve the performance of machine learning models or other data analysis tasks by utilizing the complimentary information offered by both modalities [125]. Metadata characteristics describe additional details linked to contextual information about images. The dataset includes information about the age, gender and anatomic site in context. Understanding the images and their accompanying semantics may benefit from the valuable context and insights provided by these metadata. Combining image and metadata information has various advantages, including,

- **Improved Performance:** The fusion approach can capture a more thorough representation of the data by merging information from many modalities, which enhances performance in image classification tasks.
- **Enhanced Robustness:** Metadata features can offer more context and semantics to the data, which may help to clarify any limitations or ambiguities in the visual material. The model can handle situations where the visual content alone may not be sufficient by utilizing both image and metadata information, making it more resilient to fluctuations in the data.
- **Improved Interpretability:** Combining image and metadata elements can produce insights that can be used to interpret decisions. It is simpler to comprehend the variables affecting the model's predictions or classifications when both visual and contextual information is taken into account.

5.10.1 Metadata Features Preparation

After ISIC 2019 and ISIC 2020 datasets were combined, only age, gender and anatomic site features were used as metadata. It is seen in the dataset that not all data are filled in these categories. First of all, when the gender category is examined, it is seen in figure 5.19 that 449 samples are not filled, that is, data is missing in non-a-number(nan).

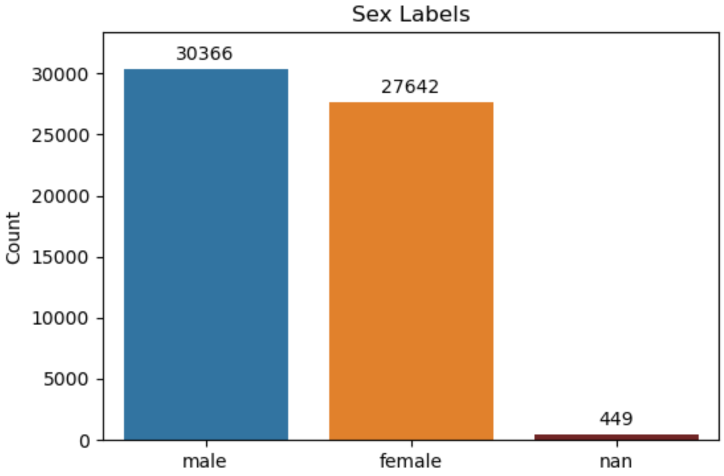


Figure 5.19 Different gender labels in the dataset

Then, when the anatomic site category is examined, it is seen in figure 5.20 that 3158 samples are again not filled, data is missing in non-a-number(nan).

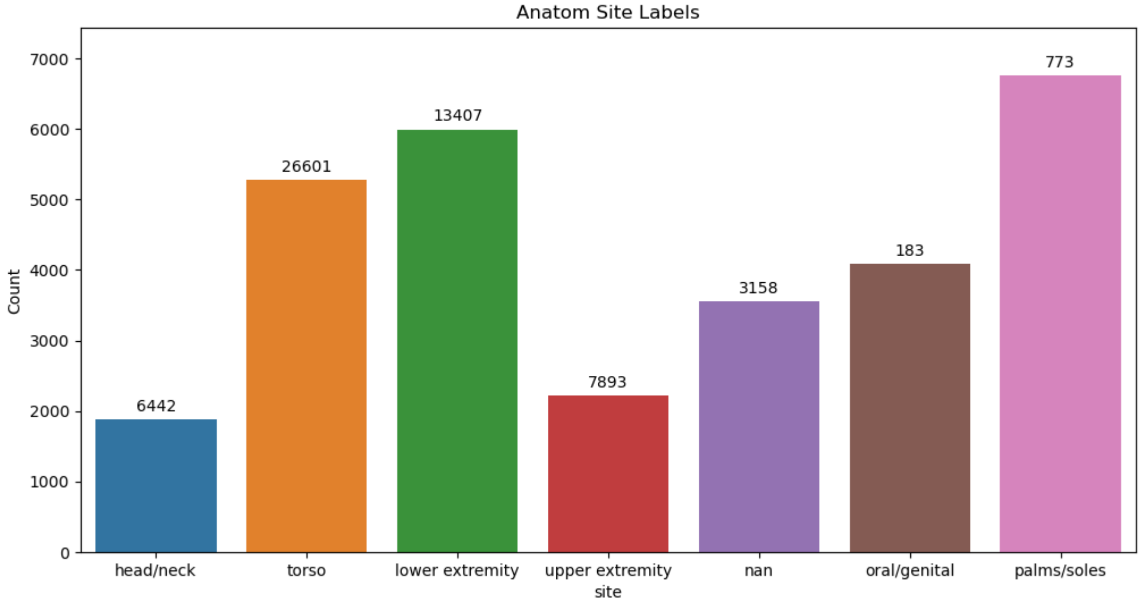


Figure 5.20 Different anatomic site labels in the dataset

Finally, when the age category is examined, it is seen in figure 5.21 that 505 samples are again not filled, data is missing in non-a-number(nan).

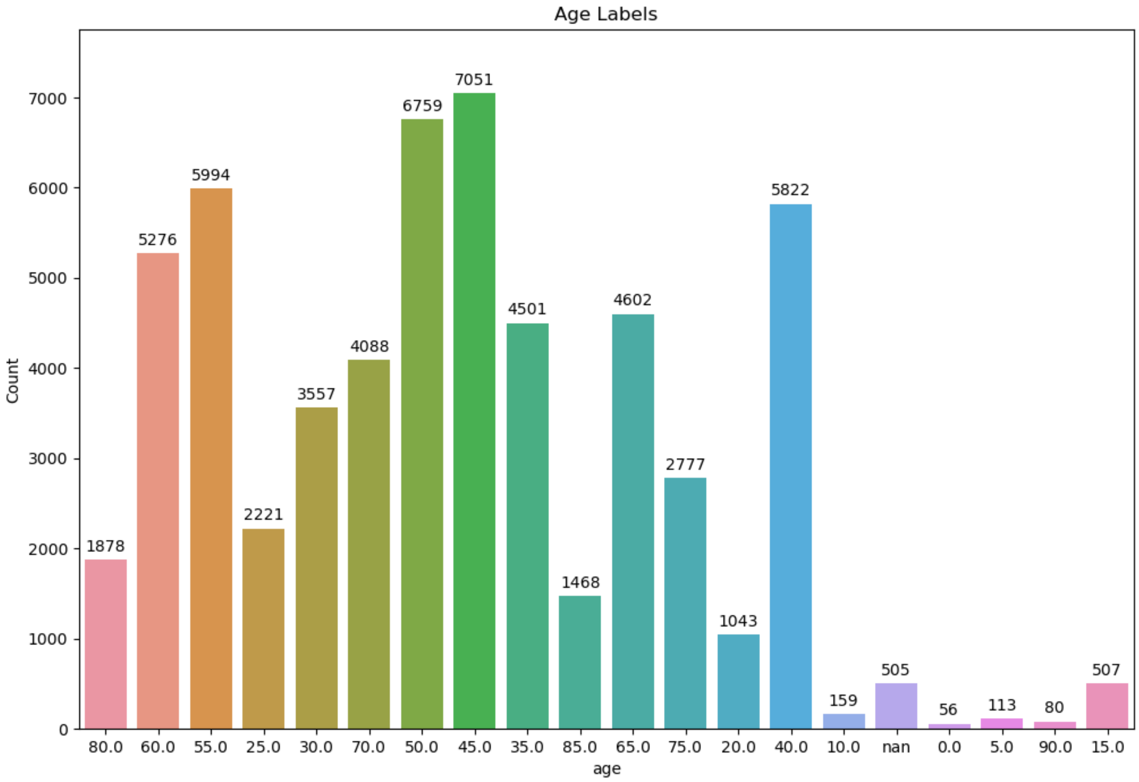


Figure 5.21 Different age labels in the dataset

It was necessary to combine different labels in these three categories. For the gender category, 3 labels consisting of male, female and non-a-number(nan) can be taken. Then, 7 labels consisting of head/neck, torso, lower extremity, upper extremity, oral/genital, palms/soles and nan can be taken for the anatomic site category. And finally, for the age category, 20 labels consisting of 0.0, 5.0, 10.0, 15.0, 20.0, 25.0, 30.0, 35.0, 40.0, 45.0, 50.0, 55.0, 60.0, 65.0, 70.0, 75.0, 80.0, 85.0, 90.0 and non-a-number(nan) can be taken.

Thus, a metadata conversion result consisting of 30 different categories in total was obtained. One-hot encoding method was used to convert categorical data to binary format [126]. Each category in the variable is represented by a binary vector in one-hot encoding, with the exception of the index corresponding to the category, which is set to one. All other elements in the vector are zero. Machine learning algorithms are able to handle and comprehend categorical data efficiently because of its binary vector representation.

5.10.2 Proposed Image Features and Metadata Fusion Model

Due to its capacity to automatically learn hierarchical and discriminative features from raw image data, CNNs have grown to be a prominent alternative for feature extraction in computer vision tasks. CNNs were created with the explicit purpose of utilizing the spatial correlations found in images and capturing regional patterns and structures. The spatial dimensions are downsampled by the global average pooling layers. Flattening is done for reshaping multidimensional feature maps to a one-dimensional vector. The image and metadata converted to one dimension are combined.

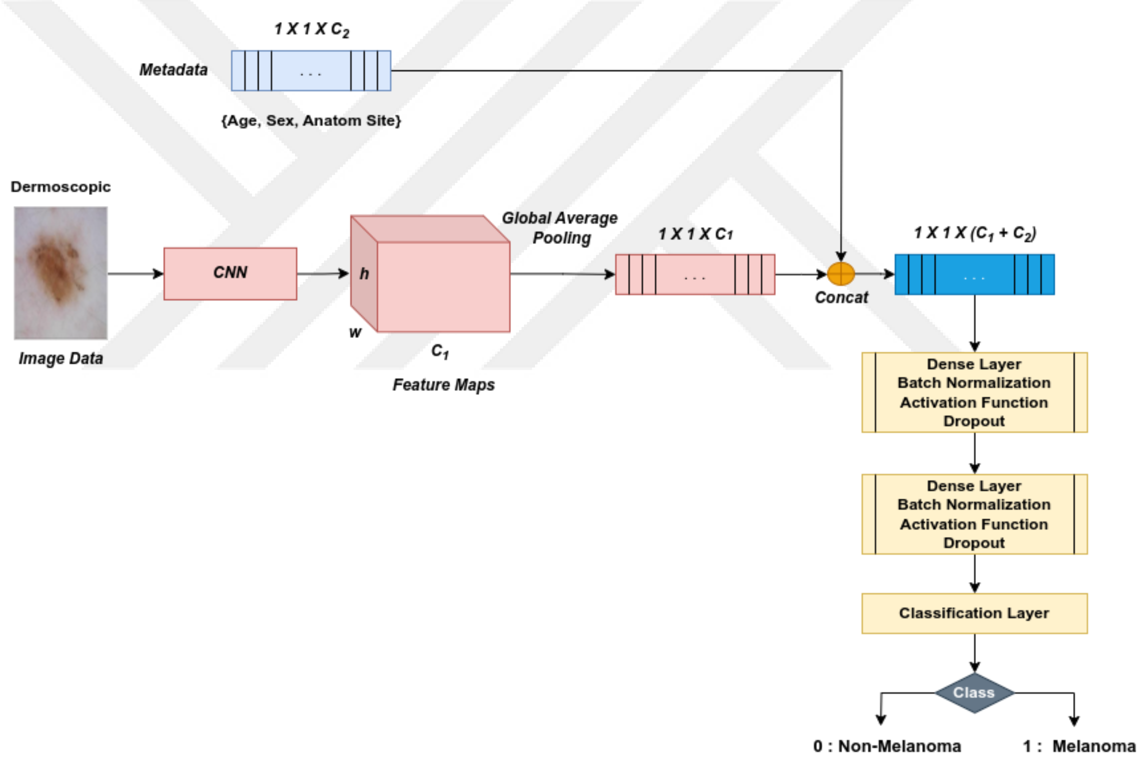


Figure 5.22 Conventional concatenation-based image and metadata fusion

The dense layer applies a set of weights to each input, followed by an activation function, using the flattened feature maps or the output from the previous layers as its input. The weights establish how much each input contributes to the dense layer's output. By minimizing internal covariate shift, batch normalization was employed to normalize the activations of a network. The network can learn complex patterns and produce nonlinear predictions thanks to the activation function's introduction of nonlinearity. In order to prevent neurons from overly depending on one another while making predictions, dropout was employed to limit co-adaptation between neurons. The classification layer was used for which generates predictions for melanoma or non-melanoma as output. Image features and metadata features are combined and the proposed model is given in figure 5.22.

Table 5.12 Comparison results of previous models and vignetting effect filter with random hair noise augmentations applied

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Base Model	93.66	94.13	0.9379	0.9290
Step 1: Shades Of Gray	93.69	94.00	0.9380	0.9299
Step 2: Deep Augmentations	93.72	94.18	0.9384	0.9336
Step 3c: OneCycleLR-cos	94.00	94.46	0.9411	0.9350
Step 4c: AdamW	94.04	94.39	0.9416	0.9369
Step 5b: Binary Focal Loss	93.88	94.26	0.9401	0.9373
Step 6: Vignetting Effect	93.98	94.43	0.9408	0.9378
Step 7b: Hair Noise	93.83	94.32	0.9395	0.9379
Step 8: Vignetting Effect and Hair Noise	93.85	94.23	0.9396	0.9381
Step 9: Metadata	93.96	94.33	0.9408	**0.9400**

AUC: Area Under Curve

Comparative scores are given in table 5.12 and as can be seen from the table, using metadata features stands out as a notable increase in the AUC metric.

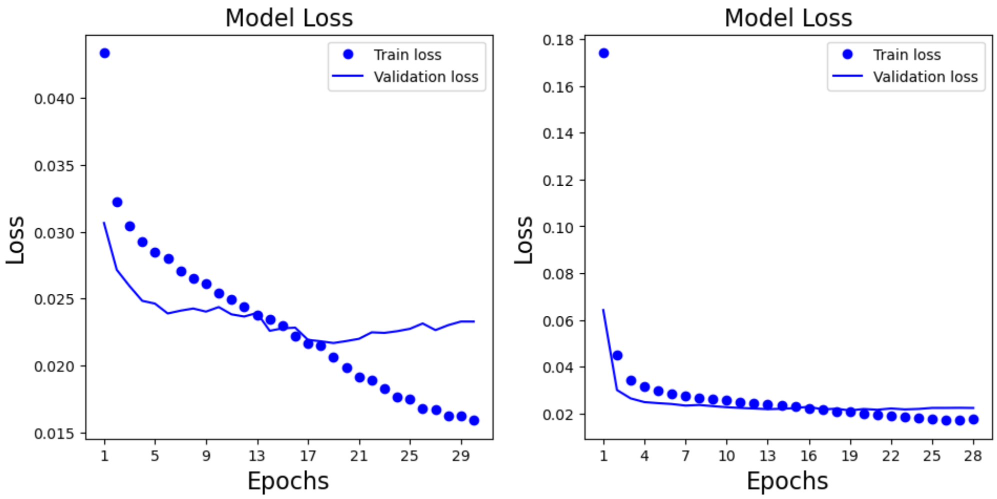


Figure 5.23 Step 8: Vignetting Effect and Hair Noise applied model on the left, Step 9: metadata applied model on the right

5.11 Step 10: Pretrained CNN Architectures

In this section, results of selected cnn models with or without metadata are given together, in addition to adequate explanations about other cnn architectures.

5.11.1 ResNet-101

In comparison to its predecessors, ResNet-101 extends the ResNet architecture by adding 101 levels, making it deeper and more expressive. The additional layers enable it to efficiently learn hierarchical representations and capture more complicated features.

Table 5.13 Comparison results of ResNet-50 and ResNet-101 models

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Step 8 : ResNet-50 without metadata	93.85	94.23	0.9396	0.9381
Step 9: ResNet-50 with metadata	93.96	94.33	0.9408	0.9400
Step 10a: ResNet-101 without metadata	93.89	94.27	0.9400	0.9388
Step 10b: ResNet-101 with metadata	94.04	94.45	0.9416	**0.9409**

AUC: Area Under Curve

As seen in table 5.13, as the CNN Backbone, ResNet-101 outperformed ResNet-50. Better results were obtained with the ResNet-101 architecture in both excluding and including metadata in terms of learning distinguishing features and demonstrating the efficiency of improving classification accuracy.

5.11.2 DenseNet-169

When compared to previous architectures, DenseNet-169 uses fewer parameters while maintaining or even enhancing performance. Because of the tight interconnectedness, features can be reused and data from lower layers can be transmitted straight to higher ones. Because of the decreased network redundancy and improved parameter efficiency, DenseNet-169 uses less memory and performs computations more quickly.

Table 5.14 Comparison results of ResNet-50 and DenseNet-169 models

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Step8: ResNet-50 without metadata	93.85	94.23	0.9396	0.9381
Step 9: ResNet-50 with metadata	93.96	94.33	0.9408	0.9400
Step 10c: DenseNet-169 without metadata	93.91	94.34	0.9404	0.9384
Step 10d: DenseNet-169 with metadata	94.03	94.45	0.9415	**0.9414**

AUC: Area Under Curve

As seen in table 5.14, as the CNN Backbone, DenseNet-169 outperformed ResNet-50. Better results were obtained with the DenseNet-169 architecture in both excluding and including metadata in terms of the vanishing gradient issue that deep neural networks frequently experience is reduced with DenseNet-169. In order to improve information transmission and solve the problem of vanishing gradients, the dense connection enables gradients to flow straight to prior layers.

5.11.3 The Squeeze-and-Excitation (SE) ResNeXt_50_32x4d

The ResNet and ResNeXt designs, which have already shown good performance in image classification tasks, serve as the foundation for the ResNeXt_50_32x4d architecture. The ability of the model to recognize intricate patterns and relationships within the data is further improved with the inclusion of the SE module. The network can learn more expressive representations and attain improved accuracy because of this expanded capacity.

Table 5.15 Comparison results of ResNet-50 and Se_Resnext50_32x4d models

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Step 8: ResNet-50 without metadata	93.85	94.23	0.9396	0.9381
Step 9: ResNet-50 with metadata	93.96	94.33	0.9408	0.9400
Step 10e: Se_Resnext50_32x4d without metadata	94.07	94.40	0.9419	0.9397
Step 10f: Se_Resnext50_32x4d with metadata	94.14	94.40	0.9425	**0.9427**

AUC: Area Under Curve

As seen in table 5.15, as the CNN Backbone, Se_Resnext50_32x4d outperformed ResNet-50. In addition to the Se_Resnext50_32x4d architecture achieving better results in both excluding and including metadata, the ResNeXt_50_32x4d design includes the SE module which adaptively recalibrates the feature maps and provides a channel attention mechanism giving it a custom focus ability by learning the relationships by channel and allows the network to focus on more informative features by giving more weight to relevant channels. This attention mechanism improves performance by increasing the discrimination capacity of the network.

5.11.4 ResNeSt-50

The Nested Residual Blocks concept is introduced in ResNeSt-50, which takes advantage of the dependencies between several feature groups to improve representation learning. These layered blocks allow the model to leverage multi-scale features and gather more fine-grained information, which improves performance.

Table 5.16 Comparison results of ResNet-50 and ResNeSt-50 models

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Step 8: ResNet-50 without metadata	93.85	94.23	0.9396	0.9381
Step 9: ResNet-50 with metadata	93.96	94.33	0.9408	0.9400
Step 10g: ResNeSt-50 without metadata	93.94	94.36	0.9408	0.9390
Step 10h: ResNeSt-50 with metadata	94.18	94.52	0.9429	**0.9415**

AUC: Area Under Curve

As seen in table 5.16, as the CNN Backbone, ResNeSt-50 outperformed ResNet-50. In addition to the ResNeSt-50 architecture achieving better results in both excluding and including metadata, ResNeSt-50 uses the Split-Attention Mechanism, which separates the input channels into various groups and computes attention across them. The network's representational strength is increased by this mechanism's ability to allow the model to focus only on educational channels. The split-attention method enables the model to capture a wider variety of distinguishing features.

5.11.5 EfficientNet-B3

A compound scaling technique is used by EfficientNet-B3 to scale the network's depth, width and resolution equally. In comparison to other architectures, this method improves accuracy with fewer parameters by striking a compromise between model size and performance. The model is able to perform at the cutting edge thanks to the compound scaling technique, which guarantees effective resource use.

Table 5.17 Comparison results of ResNet-50 and EfficientNet-B3 models

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Step 8: ResNet-50 without metadata	93.85	94.23	0.9396	0.9381
Step 9: ResNet-50 with metadata	93.96	94.33	0.9408	**0.9400**
Step 10i: EfficientNet-B3 without metadata	93.25	93.86	0.9340	0.9326
Step 10j: EfficientNet-B3 with metadata	93.47	94.03	0.9361	0.9374

AUC: Area Under Curve

As seen in table 5.17, as the CNN Backbone, EfficientNet-B3 performed worse than ResNet-50. While the performance comparison between the two architectures is dependent on specific tasks and datasets, some considerations can be made to explain why ResNet-50 might perform better in this scenario. ResNet-50 has 50 layers, but EfficientNet-B3 has less layers, making it a deeper network. A deeper network may occasionally be able to catch more intricate patterns and features, which would improve representation learning. For some tasks, the greater depth of ResNet-50 makes it possible to capture more hierarchical and abstract representations.

5.11.6 TResNet-L

The TResNet-L architecture is made up of residual blocks, each of which has two convolutional layers and a residual link. The network can discover long-range relationships thanks to the residual connections.

Table 5.18 Comparison results of ResNet-50 and TResNet-L models

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Step 8: ResNet-50 without metadata	93.85	94.23	0.9396	0.9381
Step 9: ResNet-50 with metadata	93.96	94.33	0.9408	0.9400
Step 10k: TResNet-L without metadata	94.08	94.52	0.9410	0.9398
Step 10l: TResNet-L with metadata	94.01	94.44	0.9412	**0.9427**

AUC: Area Under Curve

As seen in table 5.18, as the CNN Backbone, TResNet-L outperformed ResNet-50. In addition to the TResNet-L architecture achieving better results in both excluding and including metadata, both in terms of training and inference time, TResNet-L is effective. This is because it makes use of residual connections, which let it understand long-range dependencies without needing a lot of parameters.

5.11.7 ConvNeXt-tiny

ConvNeXt-tiny is based on the Transformer architecture, an extremely potent neural network design. ConvNeXt-tiny has successfully modified the Transformer architecture for computer vision tasks. The Transformer design has been proven to be particularly effective at natural language processing tasks.

Table 5.19 Comparison results of ResNet-50 and ConvNeXt-tiny models

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Step 8: ResNet-50 without metadata	93.85	94.23	0.9396	0.9381
Step 9: ResNet-50 with metadata	93.96	94.33	0.9408	0.9400
Step 10m: ConvNeXt-tiny without metadata	94.24	94.66	0.9432	0.9452
Step 10n: ConvNeXt-tiny with metadata	94.51	94.90	0.9459	**0.9479**

AUC: Area Under Curve

As seen in table 5.19, as the CNN Backbone, ConvNeXt-tiny outperformed ResNet-50. In addition to the ConvNeXt-tiny architecture achieving better results in both excluding and including metadata, It is highly effective, using less processing power and memory than other well-known CNN architectures and it is reasonably easy to deploy.

5.11.8 Select CNN Models

In this section, 8 CNN models with metadata and without metadata were experimented. As a result of the collective results of all models, the average of the performance metrics of the models, including and excluding the metadata, is given in the table below, sorted according to the AUC value and ordered from the highest to the lowest. is given in Table 5.20.

Table 5.20 Comparasion results of the average metrics of all CNN models

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Step 10i,j: EfficientNet-B3	93.36	93.95	0.9351	0.9350
Step 10: ResNet-50	93.91	94.28	0.9402	0.9391
Step 10a,b: ResNet-101	93.97	94.36	0.9408	0.9399
Step 10c,d: DenseNet-169	93.97	94.40	0.9409	0.9399
Step 10g,h: ResNeSt-50	94.06	94.44	0.9419	0.9403
Step 10e,f: Se_ResNeXt50_32x4d	94.11	94.40	0.9422	0.9412
Step 10k,l: TResNet-L	94.05	94.48	0.9411	0.9413
Step 10m,n: ConvNeXt-tiny	94.38	94.83	0.9446	**0.9466**

AUC: Area Under Curve

As a result of the ranking made according to the results, 6 models with the highest average AUC value from 8 models were selected to be used together in ensemble methods.

5.12 Ensemble Models

In the previous section, comparative performance results were obtained with many different models and the results of 8 different cnn models with and without metadata combined were compared and 6 different cnn models with high average AUC metric were selected. In this section, comparative accuracy results are given by using different ensemble methods in order to obtain better results by combining the predictions of these independent CNN models together. First of all, soft voting method was used with including or excluding metadata of 6 different cnn models and then hard voting and lastly optimal weighted voting from voting ensemble methods, which are also among the stacking ensemble methods principles were used.

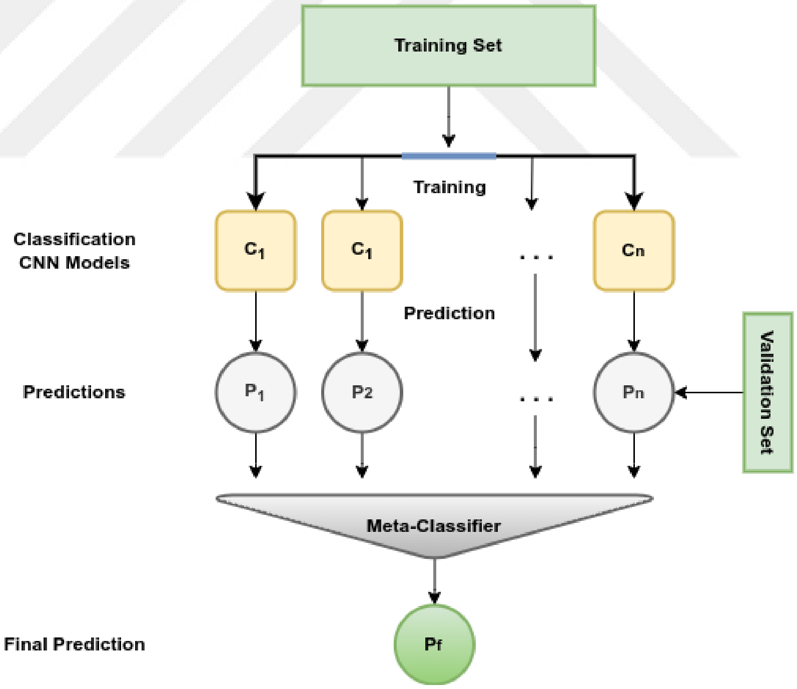


Figure 5.24 Stacked ensemble

As seen in figure 5.24, stacked ensemble is an ensemble learning method that combines the predictions of multiple individual models, called base models or learners, to make final predictions. This ensemble entail utilizing a model to figure out how to combine model predictions most effectively.

5.12.1 Soft Voting

Soft voting is a technique used in stacked ensembles to combine the predictions of the underlying models by taking into consideration the probabilities or confidence scores assigned to each class label [127]. Soft voting selects the class with the highest average probability as the ensemble prediction by averaging the predicted probabilities across the base models. In this section, the performance of the models without metadata, followed by the performance results of the models with metadata combined and the performance results of the models with both conditions are given separately.

Table 5.21 Comparison results of soft voting ensemble of selected CNN models without metadata

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Step 10c: DenseNet-169 without metadata	93.91	94.34	0.9404	0.9384
Step 10a: Resnet-101 without metadata	93.89	94.27	0.9400	0.9388
Step 10g: ResNeSt-50 without metadata	93.94	94.36	0.9408	0.9390
Step 10e: Se_ResNeXt50_32x4d without metadata	94.07	94.40	0.9419	0.9397
Step 10k: TResnet-L without metadata	94.08	94.52	0.9410	0.9398
Step 10m: ConvNeXt-tiny without metadata	94.24	94.66	0.9432	0.9452
Step 11a: Soft Voting Ensemble of Selected Models without metadata	94.83	95.10	0.9481	**0.9545**

AUC: Area Under Curve

As seen in table 5.21, soft voting ensemble of selected models without metadata outperformed than previous experimented models. The highest 0.9452 AUC score achieved in Step 10m up to this stage has increased to **0.9545** AUC score in this step.

Table 5.22 Comparison results of soft voting ensemble of selected CNN models with metadata

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Step 10c: DenseNet-169 with metadata	94.03	94.45	0.9415	0.9414
Step 10a: Resnet-101 with metadata	94.04	94.45	0.9416	0.9409
Step 10g: ResNeST-50 with metadata	94.18	94.52	0.9429	0.9415
Step 10e: Se_ResNeXt50_32x4d with metadata	94.14	94.40	0.9425	0.9427
Step 10k: TResnet-L with metadata	94.01	94.44	0.9412	0.9427
Step 10m: ConvNeXt-tiny with metadata	94.51	94.90	0.9459	0.9479
Step 11b: Soft Voting Ensemble of Selected Models with metadata	94.92	95.12	0.9489	**0.9555**

AUC: Area Under Curve

As seen in table 5.22, soft voting ensemble of selected models with metadata outperformed than previous experimented models, but also gives better AUC score than without metadata version. The highest 0.9479 AUC score achieved in Step 10m up to this stage has increased to **0.9555** AUC score in this step.

Table 5.23 Comparison results of soft voting ensemble of all selected CNN models

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Step 11a: Soft Voting Ensemble of Selected Models without metadata	94.83	95.10	0.9481	0.9545
Step 11b: Soft Voting Ensemble of Selected Models with metadata	94.92	95.12	0.9489	0.9555
Step 11c: Soft Voting Ensemble of Selected Models with and without metadata	94.97	95.21	0.9495	**0.9566**

AUC: Area Under Curve

Finally, as seen in table 5.23, comparison of soft voting method and models included and excluded from metadata is given. By combining the models with and without metadata, the AUC value reached **0.9566**, the highest score ever.

5.12.2 Hard Voting

Hard voting is a technique used in stacked ensembles to combine the predictions of the basic models by casting a majority vote [127]. In hard voting, each base model prediction is treated as a single vote and the class label with the most votes is chosen as the ensemble prediction. The class label that appears most frequently among the predicted class labels from the base models is chosen as the ensemble prediction.

Table 5.24 Comparison results of hard voting ensemble of all selected CNN models

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Step 11d: Hard Voting Ensemble of Selected Models without metadata	94.66	95.05	0.9457	0.9549
Step 11e: Hard Voting Ensemble of Selected Models with metadata	94.84	95.20	0.9480	0.9554
Step 11f: Hard Voting Ensemble of Selected Models with and without metadata	94.88	95.24	0.9484	**0.9567**

AUC: Area Under Curve

Although there is not much difference between hard voting and soft voting, it is seen in the table 5.24 that the AUC metric increases from 0.9566 to **0.9567**. These results help to prove the success of ensemble techniques.

5.12.3 Optimal Weighted Voting

In order to improve the performance of the ensemble as a whole, the approach of optimal weighted voting is employed when stacking ensembles to combine the predictions of many models in a weighted manner. This method aggregates the weighted predictions to get the final prediction, which indicates the relative relevance or competency of each model's prediction [128]. In order to identify the weights that maximize the ensemble's performance, the weights are chosen using optimization techniques. In this thesis, the prediction set of each model is multiplied by weights that add up to 1 and have 0.05 slice intervals between 0 and 1 and as a result, performance metrics are calculated again over these weights for the final prediction set. The aim here is to find the most optimal weight of the models.

Table 5.25 Comparison results of optimal weighted voting ensemble of all selected CNN models

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Step 11g: Optimal Weighted Voting Ensemble of Selected Models without metadata	94.92	95.26	0.9491	0.9553
Step 11h: Optimal Weighted Voting Ensemble of Selected Models with metadata	95.01	95.34	0.9503	0.9564
Step 11i: Optimal Weighted Voting Ensemble of Selected Models with and without metadata	95.06	95.38	0.9505	**0.9577**

AUC: Area Under Curve

At the same time, the mutually obtained classification report between metadata applied Step 10 model and Optimal Weighted Voting Ensemble of Selected Models with and without metadata applied model Step 11i is given in figure 5.25.

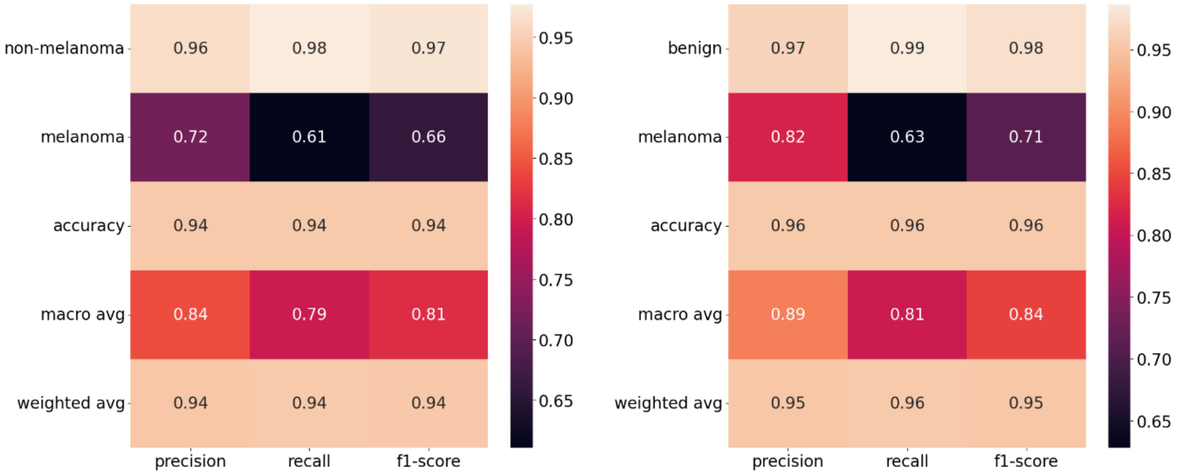


Figure 5.25 Classification report for Step 10:metadata on the left, optimal weighted voting ensemble of selected models with and without metadata on the right

At seen in figure 5.25, a significant increase was observed in almost every value. In particular, the melanoma weighted F1-Score increased from 0.66 to **0.71**. This shows us that there is a more sensitive model that distinguishes and recognizes the melanoma class more.

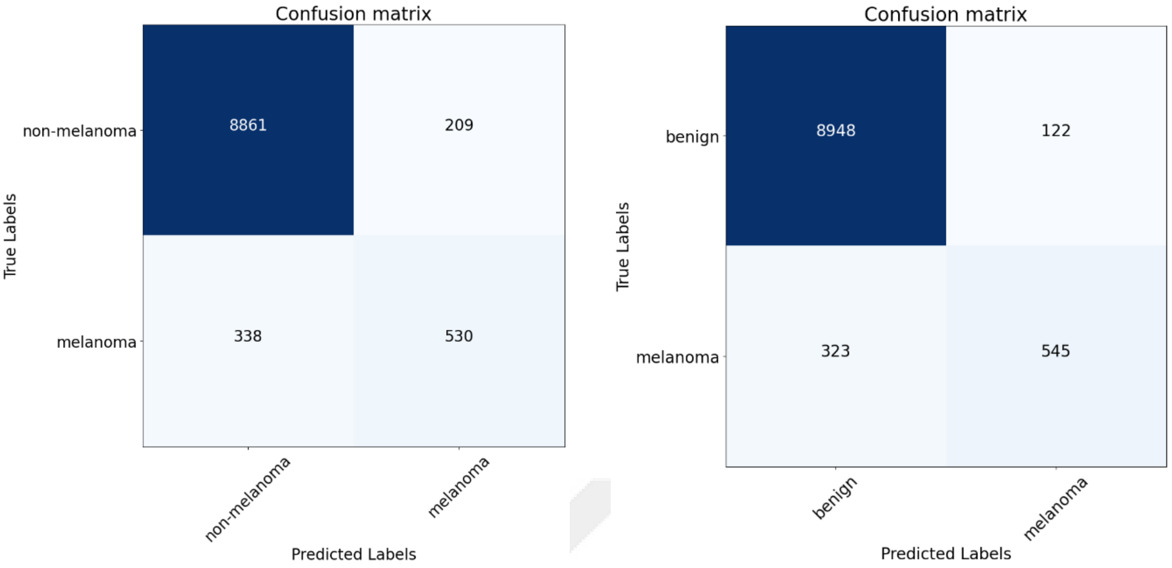


Figure 5.26 Confusion matrix for Step 10:metadata, optimal weighted voting ensemble of selected models with and without metadata on the right

At the same time, the mutually obtained confusion matrix between metadata applied Step 10 model and Optimal Weighted Voting Ensemble of Selected Models with and without metadata applied model Step 11i is given in figure 5.26. Again, a significant increase is observed in the part of the confusion matrix that is predicted as true negative of melanoma and this value is clearly seen to increase from 530 to **545**.

5.13 Comparative Results from All Steps

The comparison of all the results obtained up to this section is given in table 5.26. They are ranked according to process development.

Table 5.26 Comparison results all models with evaluation progress on validation set

Experimented Model	Precision (%)	Recall (%)	F1-Score	AUC
Base Model	92.41	94.13	0.9379	**0.9290**
Step 1: Shades Of Gray	93.69	94.00	0.9380	0.9299
Step 2: Deep Augmentations	93.72	94.18	0.9384	0.9336
Step 3c: OneCycleLR-cos	94.00	94.46	0.9411	0.9350
Step 4c: AdamW	94.04	94.39	0.9416	0.9369
Step 5b: Binary Focal Loss	93.88	94.26	0.9401	0.9373
Step 6: Vignetting Effect	93.83	94.32	0.9395	0.9379
Step 7: Hair Noise	93.98	94.43	0.9408	0.9378
Step 8:Vignetting Effect and Hair Noise	93.83	94.32	0.9395	0.9379
Step 9: Metadata	93.96	94.33	0.9408	0.9400
Step 10: ResNet-50	93.91	94.28	0.9402	0.9391
Step 10a,b: ResNet-101	93.97	94.36	0.9408	0.9399
Step 10c,d: DenseNet-169	93.97	94.40	0.9409	0.9399
Step 10g,h: ResNeSt-50	94.06	94.44	0.9419	0.9403
Step 10e,f: Se_ResNeXt50_32x4d	94.11	94.40	0.9422	0.9412
Step 10k,l: TResNet-L	94.05	94.48	0.9411	0.9413
Step 10m,n: ConvNeXt-tiny	94.38	94.83	0.9446	0.9466
Step 11i: Optimal Weighted Voting Ensemble of Selected Models with and without metadata	95.06	95.38	0.9505	**0.9577**

AUC: Area Under Curve

As can be seen in table 5.26, the accuracy of the developed models increases significantly. The AUC value, which was 0.9290 in the process that started with the base parameters, increased to **0.9577** as a result of many fine tunes and development processes. Especially with the use of ensemble methods, the improvement process has been moved to a better point.

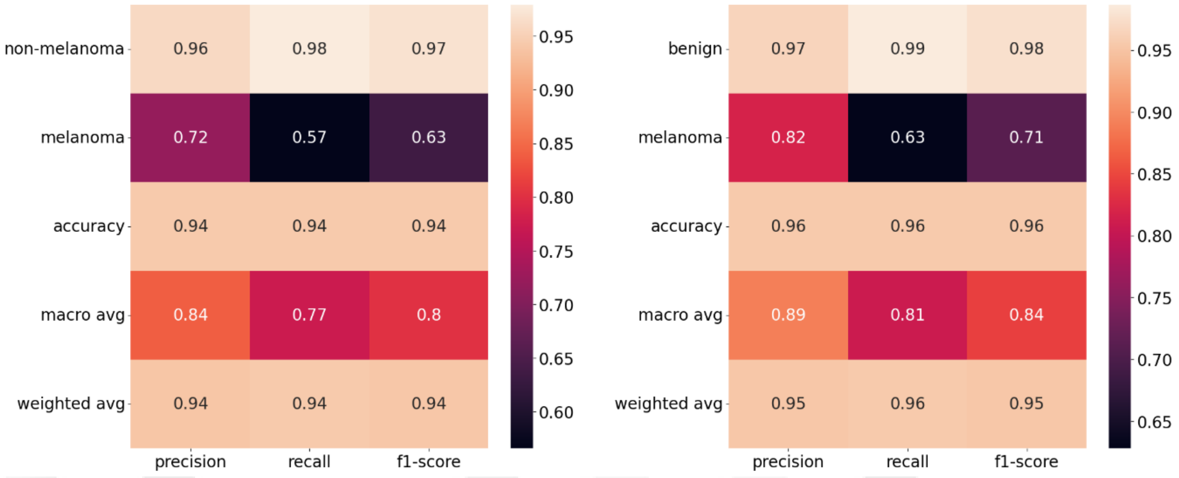


Figure 5.27 Classification report for base model on left, optimal weighted voting ensemble of selected models with and without metadata on the right

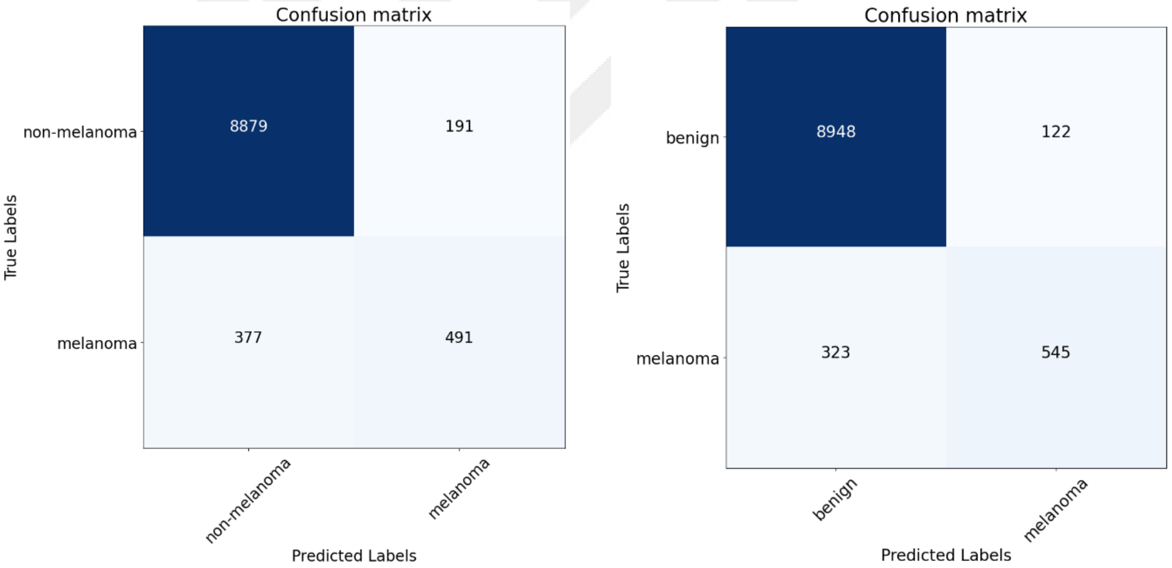


Figure 5.28 Confusion matrix for base model on the left, optimal weighted voting ensemble of selected models with and without metadata on the right

As can be seen in figure 5.27, thanks to the developed techniques, a significant increase was achieved in all metrics. For example, f1-score for the melanoma class increased from 0.63 to **0.71** and the recall value also increased from 0.57 to **0.63**. In addition, as can be seen in figure 5.28, the number of correctly predicted samples for the melanoma class increased from 491 to **545**. This proves that the developed techniques provide more accurate and precise accuracy on melanoma.

Chapter 6

Discussions

The 92% AUC achieved with the base model on the validation set for the first time has reached 95.77% success as a result of many technical and fine-tuning processes and ensemble strategies. Despite the fact that there are 2 classes and this is a binary class classification problem, the fact that there is a highly imbalance between the classes has emerged with the results obtained with the first base model, so the problem had to be evaluated from different aspects. The descriptions of the models representing each section and the average highest AUC scores from all folds obtained in these stages on validation sets are given in table 6.1.

Dermatoscopic images are essential for classifying melanoma, but adding more clinical data can improve the models' precision and clinical applicability. It is clearly seen in table 6.1 is that metadata can make a serious contribution to improving accuracy. It is possible to gain a thorough image of the disease through the integration of patient age, gender and anatomic site characteristics leading to individualized diagnosis and treatment planning. Ensemble approaches, which combine predictions from various CNN models, can increase classification accuracy while lowering the danger of overfitting. Ensemble model strategies have also been used to capitalize on the advantages of individiual models.

Until this section, the performance of the models was always based on the results obtained on the validation set. As explained before, the data was divided into 3 parts and 15% was the test set. Thanks to the performance results obtained on the validation set, fine tuning processes were carried out and how and by which methods better models were achieved step by step were explained in the previous sections. In order to see the generalization performance of the model at this stage, the performance of the models on the test set, which was never seen during the training process, is given in table 6.1.

Table 6.1 Comparison results all models with evaluation progress on validation and test set

Experimented Model	**AUC**	**AUC**
	On Validation Set	**On Test Set**
Base Model	**0.9290**	**0.9241**
Step 1: Shades Of Gray	0.9299	0.9243
Step 2: Deep Augmentations	0.9336	0.9320
Step 3c: OneCycleLR-cos	0.9350	0.9323
Step 4c: AdamW	0.9369	0.9330
Step 5b: Binary Focal Loss	0.9373	0.9358
Step 6: Vignetting Effect	0.9379	0.9368
Step 7: Hair Noise	0.9378	0.9372
Step 8:Vignetting Effect and Hair Noise	0.9379	0.9386
Step 9: Metadata	0.9400	0.9403
Step 10: ResNet-50	0.9391	0.9393
Step 10a,b: ResNet-101	0.9399	0.9402
Step 10c,d: DenseNet-169	0.9399	0.9394
Step 10g,h: ResNeSt-50	0.9403	0.9410
Step 10e,f: Se_ResNeXt50_32x4d	0.9412	0.9396
Step 10k,l: TResNet-L	0.9413	0.9402
Step 10m,n: ConvNeXt-tiny	0.9466	0.9467
Step 11i: Optimal Weighted Voting Ensemble of Selected Models with and without metadata	**0.9577**	**0.9575**

AUC: Area Under Curve

As can be seen from table 6.1, it is seen that the models developed with the proposed methods have received a significant improvement on both the validation set and the test set. The AUC value, which was 0.9241 on the test set in the process that started with the basic parameters, increased to **0.9575** as a result of many fine-tuning and development processes on the validation set. This proves that the generalization performance and abilities of the developed models are good and that the process specific to the problem has been successful.

Chapter 7

Conclusions and Future Prospects

7.1 Conclusions

With the advent of machine learning and deep learning techniques, the field of melanoma classification has made considerable strides. These methods have shown tremendous promise in increasing the precision and effectiveness of melanoma diagnosis, which will improve patient outcomes. These systems are capable of distinguishing between non-melanoma and malignant skin diseases by thoroughly analyzing massive datasets of skin images and utilizing sophisticated algorithms. Convolutional neural networks, a type of deep learning model used in machine learning, have produced encouraging results in the classification of melanoma. These algorithms are capable of accurately making predictions and automatically extracting pertinent characteristics from skin images. Furthermore, the performance of these models can be improved even further by using extra approaches like data augmentation, transfer learning and ensemble methods.

It is crucial to remember that these models should only be used as decision support tools and not as a substitute for qualified medical professionals. To guarantee the dependability and safety of these models in actual clinical settings, extensive examination and clinical validation are required. For increasing the precision and effectiveness of diagnosis, the combination of machine learning and deep learning techniques in melanoma classification shows tremendous promise. Further breakthroughs in melanoma classification and eventually better patient outcomes depend on ongoing research and development in this area as well as cooperation between clinicians and machine learning professionals.

7.2 Societal Impact and Contribution to Global Sustainability

Early identification is essential to enhancing patient outcomes in the fatal skin disease melanoma. To improve the treatment of patients, melanoma diagnosis must be accurate and made quickly. Machine learning models can help with early detection, prompt interventions, improved treatment planning and perhaps even save lives by correctly recognizing and classifying melanoma lesions. Even for expert dermatologists, melanoma diagnosis can be difficult and arbitrary. Machine learning models offer a systematic and objective method for classifying melanoma, lowering diagnostic variability and enhancing consistency among various healthcare organizations and areas.

Access to dermatology clinics or specialized dermatologists is scarce in many areas. By enabling access to precise and automated diagnosis in underserved areas, rural locales or regions with a shortage of healthcare professionals, machine learning models for melanoma classification can help close this gap. Machine learning algorithms for melanoma classification can be implemented into these platforms as telemedicine and mobile health apps become more and more prevalent. These models can be used on portable devices, making it possible for non-specialists to test for melanoma, such as general practitioners or healthcare professionals in far-off locations. This enables users to check skin lesions on themselves and obtain preliminary risk evaluations, encouraging them to seek medical assistance when necessary and fostering self-awareness.

As decision support tools, machine learning models can help dermatologists in their clinical work. These models can examine enormous datasets and offer extra insights to support dermatologists' decision-making on diagnosis and treatment, enhancing workflow effectiveness and lightening the load on medical professionals.

Healthcare professionals can optimize resource allocation, lower the need for unneeded biopsies and procedures and give high-risk patients priority for additional testing by utilizing machine learning models for melanoma classification. By reducing medical expenses, saving money and enhancing the effectiveness of the entire healthcare system, this strategy supports sustainable healthcare. These models can assist dermatologists in making more informed judgments and avoiding needless invasive treatments, which will lessen patient discomfort and healthcare expenses by offering a non-invasive and precise method of determining lesion malignancy.

7.3 Future Prospects

Although deep learning models have performed remarkably well at classifying melanoma, their decision-making procedures sometimes lack transparency. Research efforts are concentrated on creating interpretable and explainable deep learning models that can give medical professionals confidence in the model's judgment and insights into the characteristics and patterns used for classification.

It is essential to ensure the transferability and generalization of melanoma classification models across various populations and geographical areas because they are frequently trained and assessed on particular datasets. Future studies should concentrate on creating models that are flexible and effective across a range of groups, taking into account differences in skin tones, ethnicities and environmental factors.

Dermatologists can examine skin lesions in a three-dimensional virtual environment with the use of Virtual Reality (VR) and Augmented Reality (AR) technologies when melanoma classification is integrated with them. This could increase personnel's education and training while also improving diagnostic accuracy. Melanoma classification algorithms are continuously improved as a result of programs like the International Skin Imaging Collaboration (ISIC) and the exchange of annotated datasets, which encourage collaborative study, algorithm development and benchmarking. Thus, improving melanoma classification will depend heavily on collaboration between scientists, physicians, data scientists and as this is achieved, more precise systems may exist.

BIBLIOGRAPHY

[1] Melanom(Malign Melanom), https://www.derikanseri.org/melanom (01 June 2023)

[2] Lallas, A., Apalla, Z., Lazaridou, E., Ioannides, D. (2016). Dermoscopy. In Imaging in Dermatology (pp. 13-28). Academic Press.

[3] Lopez, A. R., Giro-i-Nieto, X., Burdick, J., Marques, O. (2017, February). Skin lesion classification from dermoscopic images using deep learning techniques. In 2017 13th IASTED international conference on biomedical engineering (BioMed) (pp. 49-54). IEEE.

[4] Codella, N. C., Nguyen, Q. B., Pankanti, S., Gutman, D. A., Helba, B., Halpern, A. C., Smith, J. R. (2017). Deep learning ensembles for melanoma recognition in dermoscopy images. IBM Journal of Research and Development, 61(4/5), 5-1.

[5] Bi, L., Kim, J., Ahn, E., Feng, D. (2017). Automatic skin lesion analysis using large-scale dermoscopy images and deep residual networks. arXiv preprint arXiv:1703.04197.

[6] Skin Cancer, https://medlineplus.gov/skincancer.html (01 June 2023)

[7] What is skin cancer?, https://www.cdc.gov/cancer/skin/basic_info/what-is-skin-cancer.htm (01 June 2023)

[8] Worldwide cancer data, https://www.wcrf.org/cancer-trends/worldwide-cancer-data/ (01 June 2023)

[9] Skin cancer facts & statistics, https://www.skincancer.org/skin-cancer-information/skin-cancer-facts/ (01 June 2023)

[10] Skin cancer - symptoms and causes, https://www.mayoclinic.org/diseases-conditions/skin-cancer/symptoms-causes/syc-20377605 (01 June 2023)

[11] Skin cancer - symptoms and causes - where skin cancer develops, https://www.mayoclinic.org/diseases-conditions/skin-cancer/symptoms-causes/syc-20377605#dialogId14503975 (01 June 2023)

[12] Be safe in the sun, https://www.cancer.org/cancer/risk-prevention/sun-and-uv.html (01 June 2023)

[13] The diagnostic process, https://www.ncbi.nlm.nih.gov/books/NBK338593/ (01 June 2023)

[14] What is dermatalogy?, https://dermnetnz.org/topics/what-is-a-dermatologist-what-is-dermatology (01 June 2023)

[15] Imaging in dermatology, https://www.scmsjournal.com/issues/view/imaging-in-dermatology/ (01 June 2023)

[16] Dermoscopy, https://dermnetnz.org/topics/dermoscopy (01 June 2023)

[17] Dermatoscopy, https://eledia.sumy.ua/diagnostics/dermatoscopy/#gallery-3 (01 June 2023)

[18] Dermoscopy overview and extradiagnostic applications, https://www.ncbi.nlm.nih.gov/books/NBK537131/ (01 June 2023)

[19] Kittler, H., Pehamberger, H., Wolff, K., Binder, M. J. T. I. O. (2002). Diagnostic accuracy of dermoscopy. The lancet oncology, 3(3), 159-165.

[20] Kaliyadan, F. (2016). The scope of the dermoscope. Indian Dermatology Online Journal, 7(5), 359.

[21] Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.

[22] Hastie, T., Tibshirani, R., Friedman, J. (2009). The elements of statistical learning: data mining, inference and prediction. Springer.

[23] Chapelle, O., Scholkopf, B., Zien, A. (2006). Semi-supervised learning. MIT Press.

[24] Sutton, R. S., Barto, A. G. (2018). Reinforcement learning, Second Edition: An introduction. MIT Press.

[25] Bishop, C. M. (1995). Neural Networks for Pattern Recognition. Oxford University Press.

[26] Introduction to Artificial Neural Networks (ANN), https://medium.com/analytics-vidhya/introduction-to-artificial-neural-networks-ann-5cf3b324204c (01 June 2023)

[27] Ahmadian, S., Khanteymoori, A. R. (2015, May). Training back propagation neural networks using asexual reproduction optimization. In 2015 7th Conference on Information and Knowledge Technology (IKT) (pp. 1-6). IEEE.

[28] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning. MIT Press.

[29] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

[30] Hinton, G. E., Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504-507.

[31] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014) Generative Adversarial Nets. Proceedings of the 27th International Conference on Neural Information Processing Systems, 2, 2672-2680.

[32] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Hassabis, D., et al. (2015). Human-level control through deep reinforcement learning. nature, 518(7540), 529-533.

[33] Zeiler, M. D., Fergus, R. (2014). Visualizing and understanding convolutional networks. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13 (pp. 818-833). Springer International Publishing.

[34] River Trail, http://intellabs.github.io/RiverTrail/tutorial/ (01 June 2023)

[35] Nair, V., Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10) (pp. 807-814).

[36] Maas, A. L., Hannun, A. Y., Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the 30th International Conference on Machine Learning (ICML-13) (Vol. 28, No. 1, pp. 3-11).

[37] Clevert, D. A., Unterthiner, T., Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289.

[38] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929-1958.

[39] Ioffe, S., Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning (ICML) (Vol. 37, pp. 448-456).

[40] Nielsen, M. A. (2015). Neural networks and deep learning (Vol. 25, pp. 15-24). San Francisco, CA, USA: Determination press.

[41] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning. MIT Press. Chapter 11: Practical Methodology.

[42] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning. MIT Press. Chapter 16: Transfer Learning.

[43] Illustration of transfer learning, https://www.researchgate.net/figure/Illustration-of-Transfer-Learning_fig2_345756958 (01 June 2023)

[44] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[45] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

[46] Hu, J., Shen, L., Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7132-7141).

[47] Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., Smola, A., et al. (2022). Resnest: Split-attention networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2736-2746).

[48] Tan, M., Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.

[49] Ridnik, A., Zhang, Y., Chen, J., Li, J., Liu, Z. (2021, January). TResNet: High performance GPU-dedicated architecture. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (pp. 1400-1409).

[50] Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., Xie, S. (2022). A convnet for the 2020s. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11976-11986).

[51] Dietterich, T. G. (2000, June). Ensemble methods in machine learning. In International workshop on multiple classifier systems (pp. 1-15). Berlin, Heidelberg: Springer Berlin Heidelberg.

[52] Freund, Y., Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119-139.

[53] Freund, Y., Schapire, R. E. (1996, July). Experiments with a new boosting algorithm. In icml (Vol. 96, pp. 148-156).

[54] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189-1232.

[55] Chen, T., Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

[56] Breiman, L. (1996). Bagging predictors. Machine Learning, 24(2), 123-140.

[57] Wolpert, D. H. (1992). Stacked generalization. Neural Networks, 5(2), 241-259.

[58] Melanoma warning signs and images, https://www.skincancer.org/skin-cancer-information/melanoma/melanoma-warning-signs-and-images/do-you-know-your-abcdes (01 June 2023)

[59] Cortes, C., Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

[60] Yuan, X., Yang, Z., Zouridakis, G., Mullani, N. (2006, August). SVM-based texture classification and application to early melanoma detection. In 2006 international conference of the IEEE engineering in medicine and biology society (pp. 4775-4778). IEEE.

[61] Gilmore, S., Hofmann-Wellenhof, R., Soyer, H. P. (2010). A support vector machine for decision support in melanoma recognition. Experimental dermatology, 19(9), 830-835.

[62] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[63] Janney, J.B., Roslin, S. Classification of melanoma from Dermoscopic data using machine learning techniques. Multimed. Tools Appl. 2020, 79, 3713–3728.

[64] Cover, H. (1953). Cover TM, Hart PE. Nearest neighbor pattern classification, IEEE Trans. Inf. Theory, 13(1), 21-27.

[65] John GH, Langley P. Estimating continuous distributions in bayesian classifiers. In: Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc. 1995; 338–345.

[66] Kavitha, J.C.; Suruliandi, A.; Nagarajan, D.; Nadu, T. Melanoma detection in dermoscopic images using global and local feature extraction. Int. J. Multimed. Ubiquitous Eng. 2017, 12, 19–28.

[67] Noel B. Linsangan and Jetron J. Adtoon. 2018. Skin Cancer Detection and Classification for Moles Using K-Nearest Neighbor Algorithm. In Proceedings of the 5th International Conference on Bioinformatics Research and Applications (ICBRA '18). Association for Computing Machinery, New York, NY, USA, 47–51.

[68] ISIC, https://www.isic-archive.com/ (01 June 2023)

[69] Balaji, V.; Suganthi, S.; Rajadevi, R.; Kumar, V.K.; Balaji, B.S.; Pandiyan, S. Skin disease detection and segmentation using dynamic graph cut algorithm and classification through Naive Bayes classifier. Measurement 2020, 163, 107922.

[70] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115–118.

[71] Adegun, A. A., Viriri, S. (2019). Deep learning-based system for automatic melanoma detection. IEEE Access, 8, 7160-7172.

[72] Mendonca, T.; Ferreira, P.M.; Marques, J.S.; Marcal, A.R.S.; Rozeira, J. PH2-A Dermoscopic Image Database for Research and Benchmarking. In Proceedings of the 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Osaka, Japan, 3–7 July 2013; pp. 5437–5440.

[73] Tschandl, P., et al. (2018). "Comparison of the accuracy of human readers versus machine-learning algorithms for pigmented skin lesion classification: an open, web-based, international, diagnostic study." The Lancet Oncology, 19(3), 328-337.

[74] Le, D.N.T., Le, H.X., Ngo, L., Ngo, H.T. Transfer learning with class-weighted and focal loss function for automatic skin cancer classification. arXiv 2020, arXiv:2009.05977.

[75] Codella, N. C., Gutman, D., Celebi, M. E., Helba, B., Marchetti, M. A., Dusza, S. W., Halpern, A., et al. (2018, April). Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018) (pp. 168-172). IEEE.

[76] Tschandl, P., Rosendahl, C., Kittler, H. (2018). Data descriptor: the HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Sci. Data, 5(1).

[77] Combalia, M., Codella, N. C., Rotemberg, V., Helba, B., Vilaplana, V., Reiter, O., Malvehy, J., et al. (2019). Bcn20000: Dermoscopic lesions in the wild. arXiv preprint arXiv:1908.02288.

[78] Rotemberg, V., Kurtansky, N., Betz-Stablein, B., Caffery, L., Chousakos, E., Codella, N., Soyer, H. P., et al. (2021). A patient-centric dataset of images and metadata for identifying melanomas using clinical context. Scientific data, 8(1), 34.

[79] Kassem, M. A., Hosny, K. M., Fouad, M. M. (2020). Skin lesions classification into eight classes for ISIC 2019 using deep convolutional neural network and transfer learning. IEEE Access, 8, 114822-114832.

[80] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A., et al. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).
[81] Gessert, N., Nielsen, M., Shaikh, M., Werner, R., Schlaefer, A. (2020). Skin lesion classification using ensembles of multi-resolution EfficientNets with meta data. MethodsX, 7, 100864.
[82] Karki, S., Kulkarni, P., Stranieri, A. (2021, February). Melanoma classification using EfficientNets and Ensemble of models with different input resolution. In 2021 Australasian Computer Science Week Multiconference (pp. 1-5).
[83] Kaur, R., GholamHosseini, H., Sinha, R., Lindén, M. (2022). Melanoma classification using a novel deep convolutional neural network with dermoscopic images. Sensors, 22(3), 1134.
[84] Tziomaka, M., Maglogiannis, I. (2021). Ensembles of deep convolutional neural networks for detecting melanoma in dermoscopy images. In Computational Collective Intelligence: 13th International Conference, ICCCI 2021, Rhodes, Greece, September 29–October 1, 2021, Proceedings 13 (pp. 523-535). Springer International Publishing.
[85] Jaisakthi, S.M.; Mirunalini, P.; Aravindan,C. , Appavu, R. (2023). Classification of skin cancer from dermoscopic images using deep neural network architectures. Multimedia Tools and Applications, 82(10), 15763-15778.
[86] About, https://www.isic-archive.com/mission (01 June 2023)
[87] ISIC Challenge: https://challenge.isic-archive.com/landing/2019/ (01 June 2023)
[88] Gonzalez, R.C., Woods, R.E. (2008). Digital Image Processing (3rd ed.). Pearson.
[89] Hastie, T., Tibshirani, R., Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference and Prediction (2nd ed.). Springer.
[90] Pratt, W.K. (2007). Digital Image Processing: PIKS Inside (3rd ed.). Wiley-Interscience.
[91] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.
[92] Bishop, C.M. (2006). Pattern Recognition and Machine Learning. Springer.
[93] Kingma, D. P., Ba, J. (2014). Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980.
[94] Tieleman, T., Hinton, G. (2012). RMSProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning.
[95] Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems (NIPS) (pp. 1097-1105).
[96] Japkowicz, N., Shah, M. (2011). Evaluating Learning Algorithms: A Classification Perspective. Cambridge University Press.
[97] Powers, D. M. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. Journal of Machine Learning Technologies, 2(1), 37-63.
[98] Land, E. H., McCann, J. J. (1971). Lightness and retinex theory. Journal of the Optical Society of America, 61(1), 1-11.
[99] Gijsenij, A., Gevers, T., Van De Weijer, J. (2011). Computational color constancy: Survey and experiments. IEEE transactions on image processing, 20(9), 2475-2489.

[100] Barata, C., Marques, J. S., Celebi, M. E. (2014, October). Improving dermoscopy image analysis using color constancy. In 2014 IEEE International Conference on Image Processing (ICIP) (pp. 3527-3531). IEEE.

[101] Perez, L., Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.04621.

[102] Pizer, S. M., Amburn, E. P., Austin, J. D., Cromartie, R., Geselowitz, A., Greer, T., Zuiderveld, K., et al. (1987). Adaptive histogram equalization and its variations. Computer Vision, Graphics and Image Processing, 39(3), 355-368.

[103] Shorten, C., Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, 6(1), 60.

[104] DeVries, T., Taylor, G. W. (2017). Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552.

[105] Albumentations: fast and flexible image augmentations, https://albumentations.ai/ (01 June 2023)

[106] Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A. A. (2020). Albumentations: fast and flexible image augmentations. Information, 11(2), 125.

[107] Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum and weight decay. arXiv preprint arXiv:1803.09820.

[108] Smith, L. N. (2017). Cyclical learning rates for training neural networks. In Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on (pp. 464-472). IEEE.

[109] Loshchilov, I., Hutter, F. (2016). SGDR: Stochastic gradient descent with warm restarts. In Proceedings of the International Conference on Learning Representations (ICLR).

[110] Smith, L. N., Topin, N. (2019, May). Super-convergence: Very fast training of neural networks using large learning rates. In Artificial intelligence and machine learning for multi-domain operations applications (Vol. 11006, pp. 369-386). SPIE.

[111] Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010 (pp. 177-186). Physica-Verlag HD.

[112] Robbins, H., Monro, S. (1951). A stochastic approximation method. The annals of mathematical statistics, 400-407.

[113] Dozat, T. (2016). Incorporating Nesterov Momentum into Adam. International Conference on Learning Representations (ICLR) Workshop.

[114] Heo, B., Chun, S., Oh, S. J., Han, D., Yun, S., Kim, G. , Ha, J. W., et al. (2020). Adamp: Slowing down the slowdown for momentum optimizers on scale-invariant weights. arXiv preprint arXiv:2006.08217.

[115] Loshchilov, I., Hutter, F. (2017). Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

[116] He, H., Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on knowledge and data engineering, 21(9), 1263-1284.

[117] Lin, T. Y., Goyal, P., Girshick, R., He, K., Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980-2988).

[118] Szeliski, R. (2022). Computer vision: algorithms and applications. Springer Nature.

[119] Abbas, Q., Celebi, M. E., García, I. F. (2011). Hair removal methods: A comparative study for dermoscopy images. Biomedical Signal Processing and Control, 6(4), 395-404.

[120] Abbas, Q., Garcia, I. F., Emre Celebi, M., Ahmad, W. (2013). A feature-preserving hair removal algorithm for dermoscopy images. Skin Research and Technology, 19(1), e27-e36.

[121] Maglogiannis, I., Delibasis, K. (2015, August). Hair removal on dermoscopy images. In 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 2960-2963). IEEE.
[122] Toossi, M. T. B., Pourreza, H. R., Zare, H., Sigari, M. H., Layegh, P., Azimi, A. (2013). An effective hair removal algorithm for dermoscopy images. Skin Research and Technology, 19(3), 230-235.
[123] Lee, T., Ng, V., Gallagher, R., Coldman, A., McLean, D. (1997). Dullrazor®: A software approach to hair removal from images. Computers in biology and medicine, 27(6), 533-543.
[124] Alizadeh, S. M., Mahloojifar, A. (2021). Automatic skin cancer detection in dermoscopy images by combining convolutional neural networks and texture features. International Journal of Imaging Systems and Technology, 31(2), 695-707.
[125] Ningrum, D. N. A., Yuan, S. P., Kung, W. M., Wu, C. C., Tzeng, I. S., Huang, C. Y., Wang, Y. C., et al. (2021). Deep learning classifier with patient's metadata of dermoscopic images in malignant melanoma detection. Journal of Multidisciplinary Healthcare, 877-885.
[126] Kotsiantis, S. B., Kanellopoulos, D., Pintelas, P. E. (2006). Data preprocessing for supervised leaning. International journal of computer science, 1(2), 111-117.
[127] Rokach, L. (2010). Ensemble-based classifiers. Artificial intelligence review, 33, 1-39.
[128] Zhou, Z. H. (2012). Ensemble methods: foundations and algorithms. CRC press.

CURRICULUM VITAE

2002 – 2007	B.Sc., Computer Engineering, Erciyes University, Kayseri, TURKEY
2020 – 2023	M.Sc., Electrical and Computer Engineering, Abdullah Gül University, Kayseri, TURKEY

SELECTED PUBLICATIONS AND PRESENTATIONS

There are no publications and presentations yet.