A project report on

SKIN CANCER DETECTION USING NEURAL NETWORKS

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering with specialization in Artificial Intelligence and Machine Learning

by

MAHAK SURANA (19BAI1027)

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

24 April, 2023

SKIN CANCER DETECTION USING NEURAL NETWORKS

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering with specialization in Artificial Intelligence and Machine Learning

by

MAHAK SURANA (19BAI1027)

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

24 April, 2023

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

DECLARATION

I hereby declare that the thesis entitled "SKIN CANCER DETECTON USING NEURAL NETWORKS" submitted by me, for the award of the degree of Bachelor of Technology in Computer Science and Engineering with Specialization in Artificial Intelligence and Machine Learning, Vellore Institute of Technology, Chennai, is a record of bonafide work carried out by me under the supervision of Dr. JENILA LIVINGSTON L M.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Chennai

Date: Signature of the Candidate

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

School of Computer Science and Engineering

CERTIFICATE

This is to certify that the report entitled **"Skin Cancer Detection using Neural Networks"** is prepared and submitted by **Mahak Surana(19BAI1027)** to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science and Engineering with Specialization in Artificial Intelligence and Machine Learning** program is a bonafide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Signature of the Guide:

Name: Dr. JENILA LIVINGSTON L M

Date:

Signature of the Examiner 1

Name:

Date:

Signature of the Examiner 2

Name:

Date:

Approved by the Head of Department

B. Tech. CSE with Specialization in Artificial Intelligence and Machine Learning

Name: Dr. Sweetlin Hemalatha C

Date: 24 – 04 – 2023

(Seal of SCOPE)

ABSTRACT

Skin cancer is a dangerous disease that can spread to other parts of the body. It is caused by DNA damage in skin cells that leads to cancerous lesions. Detecting skin cancer at an early stage is crucial for successful treatment. Dermoscopy, a specialized imaging technique, has become an important tool in skin cancer detection. Artificial intelligence (AI) and specifically neural networks have shown promise in automating the detection of skin cancer from dermoscopic images. Detection of skin cancer at the beginning stage is very crucial, so that it can be curable. In recent years, there has been a growing interest in leveraging the power of artificial intelligence (AI) and specifically neural networks, to automate the detection of skin cancer from dermoscopic images. These models are trained on large datasets of images and learn to identify patterns and features associated with skin cancer, enabling them to make accurate predictions.

In this study, I used various deep learning models, including Siamese model embedding with VGG-16, CNN using Class Weight, VGG-16 using Focal loss, Class weight and Learning rate scheduler, and Vision Transformer (ViT), to compare the performance of different neural network architectures in skin cancer detection. To train and test these models, I created my own dataset by merging two existing datasets, MNIST HAM10000 and ISIC 2017 skin cancer dataset. The resulting dataset comprised a total of 11,068 images of skin cancer, which were categorized into three subsets: Training, validation, and testing, and classified into 9 different classes.

The results of this study demonstrated that Vision Transformer accurately classified skin cancer with high precision and recall. This suggests that the developed system holds promise as a reliable solution for skin cancer classification systems. The use of AI and neural networks in classification has the potential to significantly improve the efficiency and accuracy of diagnosis, leading to more effective and timely treatment for patients. Automated skin cancer detection systems can have several advantages, including affordability and increased accessibility, time efficiency, and improved decision-making. These systems can aid in the decision-making process by offering accurate predictions based on patterns and features learned from large datasets, leading to improved patient care.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Dr. JENILA LIVINGSTON L M SCOPE, Vellore Institute of Technology, Chennai, for her constant guidance, continual encouragement, understanding; more than all, she taught me patience in my endeavor. My association with her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Machine Learning.

It is with gratitude that I would like to extend thanks to our honorable Chancellor, Dr. G. Viswanathan, Vice Presidents, Mr. Sankar Viswanathan, Dr. Sekar Viswanathan and Mr. G V Selvam, Assistant Vice-President, Ms. Kadhambari S. Viswanathan, Vice-Chancellor, Dr. Rambabu Kodali, Pro-Vice Chancellor, Dr. V. S. Kanchana Bhaaskaran and Additional Registrar, Dr. P.K.Manoharan for providing an exceptional working environment and inspiring all of us during the tenure of the course.

Special mention to Dean, Dr. Ganesan R, Associate Dean Academics, Dr. Parvathi R and Associate Dean Research, Dr. Geetha S, SCOPE, Vellore Institute of Technology, Chennai, for spending their valuable time and efforts in sharing their knowledge and for helping us in every aspect.

In jubilant mood I express ingeniously my whole-hearted thanks to Dr. Sweetlin Hemalatha C, Head of the Department, Project Coordinators, Dr. Padmavathy T V, Dr. Abdul Quadir Md, and Dr. Priyadarshini R, B.Tech. Computer Science and Engineering with Specialization in Artificial Intelligence and Machine Learning, SCOPE, Vellore Institute of Technology, Chennai, for their valuable support and encouragement to take up and complete the thesis.

My sincere thanks to all the faculties and staff at Vellore Institute of Technology, Chennai, who helped me acquire the requisite knowledge. I would like to thank my parents for their support. It is indeed a pleasure to thank my friends who encouraged me to take up and complete this task.

Place: Chennai

Date: Mahak Surana

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

SNN	Siamese Neural Network
CNN	Convolution Neural Network
CW	Class-Weight
FL	Focal Loss
LRS	Learning Rate Scheduler
VGG	Visual Geometry Group
ViT	Vision Transformer
Siamese1	Siamese Model when loss margin set as ‘1’
Siamese2	Siamese Model when loss margin set as ‘2’

Chapter 1

Introduction

Skin cancer is a type of cancer that originates in the skin cells and is the most common form of cancer globally. It is a malignant tumor that develops in the skin cells due to the uncontrolled growth of abnormal cells. It is a serious and prevalent form of cancer that affects millions of people worldwide. According to the World Health Organization (WHO), over 3 million new cases of skin cancer are diagnosed each year, and early detection plays a critical role in improving patient outcomes. Early diagnosis allows for timely intervention and treatment, which can significantly increase the chances of successful recovery and reduce the risk of metastasis. In recent years, the field of artificial intelligence (AI) and machine learning, particularly neural networks, has shown great promise in aiding skin cancer detection.

In recent years, advancements in artificial intelligence (AI) and machine learning have revolutionized the field of medical imaging, including skin cancer detection. Among the various AI techniques, neural networks have emerged as a powerful tool for image classification tasks, including skin cancer detection.

Neural networks, particularly Convolutional Neural Networks (CNNs), are capable of automatically learning complex patterns and features from large datasets, making them well-suited for analyzing skin lesion images and identifying potential malignancies. CNNs can capture both local and global features from the images, which enables them to make accurate predictions about the presence of skin cancer. Convolutional Neural Networks (CNNs), Siamese Networks and Vision Transformers are among the popular neural network architectures that have been employed for skin cancer classification.

The research objective of this paper is to compare and evaluate the performance of different neural network architectures for skin cancer detection. Specifically, we will investigate the accuracy, computational efficiency, interpretability, and explainability of various neural networks, such as Siamese Networks, VGG (Visual Geometry Group) Networks, Vision Transformers and CNNs, in the context of skin cancer classification. By conducting a comparative analysis of these neural network architectures, we aim to provide insights into

their strengths and weaknesses, and identify which architecture may be most suitable for skin cancer detection tasks. By comparing and contrasting the performance of these neural network architectures, we aim to provide insights into their suitability for skin cancer detection tasks and identify potential areas for further research and improvement in this field. This research can potentially contribute to the development of more effective and efficient skin cancer detection methods, ultimately improving patient outcomes in the clinical setting.

1.1. COMPLEXITY OF THE CLASSIFICATION TASK

Skin cancer classification is a challenging task due to several factors, including the variability in skin lesion appearances, the presence of different types of skin cancer with varying characteristics, as depicted in Fig1.1, the potential for misdiagnosis and false positives, and the need for accurate and timely detection to improve patient outcomes.

Fig1.1 Two different type of skin cancer

The complexity of the task is further compounded by the large and diverse dataset required for training neural networks. Obtaining a comprehensive dataset with a sufficient number of skin lesion images that represent the various types and stages of skin cancer, along with non-cancerous lesions, can be challenging. Additionally, the dataset may be imbalanced, with uneven distribution of different classes, leading to bias in the model's performance.

Another factor contributing to the complexity of the task is the need for interpretability and explainability in the predictions made by the neural networks. In the medical field, it is crucial to understand and explain the reasoning behind the model's predictions to gain trust and acceptance from healthcare providers and patients. Therefore,

the choice of neural network architectures should take into consideration their interpretability and explainability, in addition to their accuracy and efficiency.

The computational complexity of the neural network architectures is also an important consideration. Some architectures may require significant computational resources and training time, which may limit their practical applicability in real-world clinical settings, especially in resource-constrained environments. The selected architectures should be capable of handling the variability in skin lesion appearances, effectively utilizing the available dataset, providing interpretable and explainable predictions, and being computationally efficient.

Considering the complexity of the skin cancer classification task, the choice of appropriate neural network architectures becomes crucial to achieve accurate and reliable results. The selected architectures should be capable of handling the variability in skin lesion appearances, effectively utilizing the available dataset, providing interpretable and explainable predictions, and being computationally efficient.

1.2. AIMS AND OBJECTIVES OF THE RESEARCH

The primary aim of this research is to compare and evaluate the performance of different neural network architectures, including Siamese Networks, VGG (Visual Geometry Group) Networks, Vision Transformers and CNNs for skin cancer classification. The following objectives can be concluded after this work:

- To review and analyze the literature on skin cancer detection using neural networks, including the latest advancements and state-of-the-art approaches.
- To collect and preprocess a comprehensive dataset of skin lesion images, encompassing different types and stages of skin cancer, along with non-cancerous lesions, for training and evaluation of the neural network models.
- To implement and train Siamese Networks, VGG Networks, Vision Transformers and CNN on the collected dataset, using appropriate methodologies and techniques.

- To evaluate the performance of the trained neural network models in terms of accuracy, sensitivity, specificity, and other relevant metrics for skin cancer classification.
- To compare and analyze the performance of Siamese Networks, VGG Networks, Vision Transformers and CNN in terms of their accuracy, efficiency, interpretability, and explainability.
- To interpret and explain the predictions made by the trained neural network models to gain insights into the reasoning behind their classifications and to assess their interpretability and explainability.
- To discuss the limitations of the research, identify areas for future research, and conclude with a summary of the findings and their implications for skin cancer detection using different neural networks.

By achieving these aims and objectives, this research will contribute to the understanding of the performance of different neural network architectures for skin cancer classification and provide valuable insights for the selection of appropriate architectures in clinical practice.

1.3. OUTLINE OF THE THESIS

The thesis is organized into six chapters, followed by a conclusion and several appendices. Each chapter is self-contained but interconnected, building upon the previous ones.

Chapter 1:

This chapter outlines the aims and objectives of this thesis.

Chapter 2:

This chapter focuses on skin cancer, providing an in-depth exploration of its properties and categories.

Chapter 3:

In this chapter, the significance of automating the task of skin cancer categorization is explained, along with the challenges involved, especially in capturing images in outdoor environments. The need for a scientifically sound solution is emphasized.

Chapter 4:

This chapter provides an overview of the relevant literature and prior research in the field of skin cancer classification. Various sources such as papers, technical reports, and internet resources are reviewed and analyzed, covering topics like color segmentation algorithms, Convolutional Neural Networks (CNNs), Siamese Network, Vision Transformer, and skin cancer classification.

Chapter 5:

This chapter details the methodology used to build the skin cancer classification system, including the image collection process and the image database used. It also covers the image pre-processing techniques employed and provides information on the recognition algorithm designed for skin cancer classification.

Chapter 6:

This chapter presents the experimental work carried out to evaluate the performance of the developed algorithms from the previous chapter.

Chapter 7:

This chapter summarizes the main contributions and conclusions drawn from the research conducted.

APPENDIX 1:

This contains the codes for importing libraries, model architectures, and approaches used in the skin cancer classification system.

APPENDIX 2:

This contains detailed information on the skin cancer dataset used in the study.

APPENDIX 3:

This contains the output of different models in the form of metrics used in the Skin Cancer Classification System.

Chapter 2

Skin Cancer

Skin cancer is a type of cancer that occurs when the skin cells grow uncontrollably and form a tumor. It is the most common type of cancer globally, and its prevalence is increasing. Skin cancer can be classified into three main types: melanoma, basal cell carcinoma, and squamous cell carcinoma. Early detection of skin cancer is critical for successful treatment, and various approaches have been developed for its diagnosis, including using neural networks.

Neural networks are a type of machine learning algorithm that can learn to classify data based on their input features. In the context of skin cancer diagnosis, neural networks can be trained on images of skin lesions to identify their type and severity. The input features for the neural network can include information about the color, size, shape, and texture of the lesion.

2.1 PROPERTIES OF SKIN CANCER

The physical properties of skin cancer refer to the characteristics of skin cancer lesions that can be observed or measured. These properties can provide valuable information for skin cancer detection and classification. Some of the common physical properties of skin cancer, as shown in Fig 2.1, include:

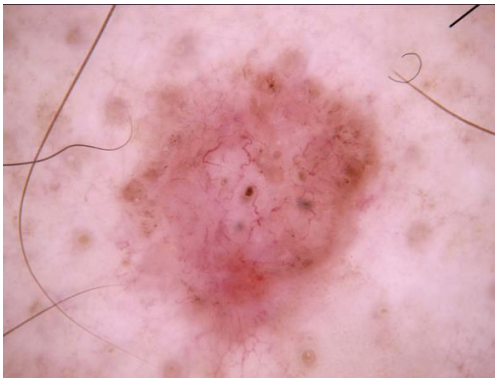

Fig 2.1 Normal Skin Cancer Image

- Shape: Skin cancer lesions can have various shapes, including asymmetrical, irregular, or poorly defined borders. Benign lesions are usually symmetrical and have regular borders, while malignant lesions often have irregular shapes with uneven borders.
- Color: Skin cancer lesions can exhibit a wide range of colors, ranging from normal skin color to shades of brown, black, blue, red, or white. Changes in color or the presence of multiple colors within a lesion can be indicative of malignancy.
- Size: Skin cancer lesions can vary in size, from small, barely noticeable lesions to large, prominent growths. Monitoring changes in the size of a lesion over time can be important for skin cancer detection and classification.
- Texture: The texture of a skin cancer lesion can provide valuable information for classification. Benign lesions usually have a smooth texture, while malignant lesions may have an irregular, rough, or scaly texture, as shown in in Fig 2.2.
- Surface characteristics: The surface of a skin cancer lesion can also provide important clues for classification. Benign lesions may have a regular, smooth surface, while malignant lesions can exhibit ulceration, bleeding, crusting, or other irregularities, as shown in in Fig 2.2.

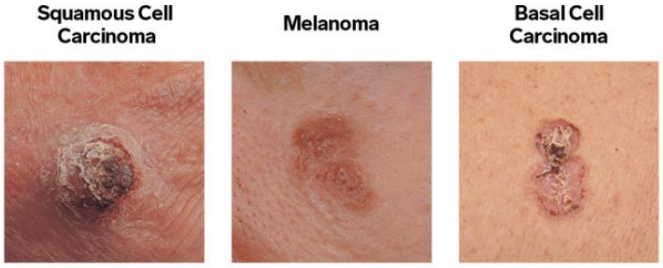

Fig 2.2 Different type of skin cancer with different structures.

- Placement: The location or placement of skin cancer lesions on the body can also be relevant for classification. Certain anatomical locations, such as the face, scalp, ears, and hands, are more prone to skin cancer development.
- Growth pattern: The growth pattern of a skin cancer lesion can also be an important physical property for classification. Benign lesions usually grow slowly and remain stable in size, while malignant lesions may exhibit rapid growth or changes in shape, color, or texture over time.

Properly analyzing and utilizing these physical properties of skin cancer lesions can aid in the accurate detection and classification of skin cancer using neural networks or other machine learning algorithms. These properties can be incorporated as features in the training data for the neural network, allowing the model to learn and identify patterns associated with different types of skin cancer.

2.2 TYPES OF SKIN CANCER

There are many different types of skin cancer, including:

- Basal cell carcinoma (BCC): This is the most common type of skin cancer, accounting for about 80% of all skin cancers. It typically appears as a small, shiny bump or nodule on the skin, often with blood vessels visible on the surface. BCC rarely spreads to other parts of the body and is usually slow-growing, but it can cause local damage if not treated early.
- Squamous cell carcinoma (SCC): SCC is the second most common type of skin cancer, accounting for about 20% of all skin cancers. It often appears as a red, scaly patch or bump on the skin, which may ulcerate and bleed. SCC has a higher risk of spreading to other parts of the body compared to BCC, although the risk is still relatively low.
- Melanoma: Melanoma is a less common but more aggressive type of skin cancer that arises from the pigment-producing cells (melanocytes) in the skin. It can develop from an existing mole or appear as a new mole-like growth on the skin. It can spread to other parts of the body and can be very dangerous if not detected and treated early.

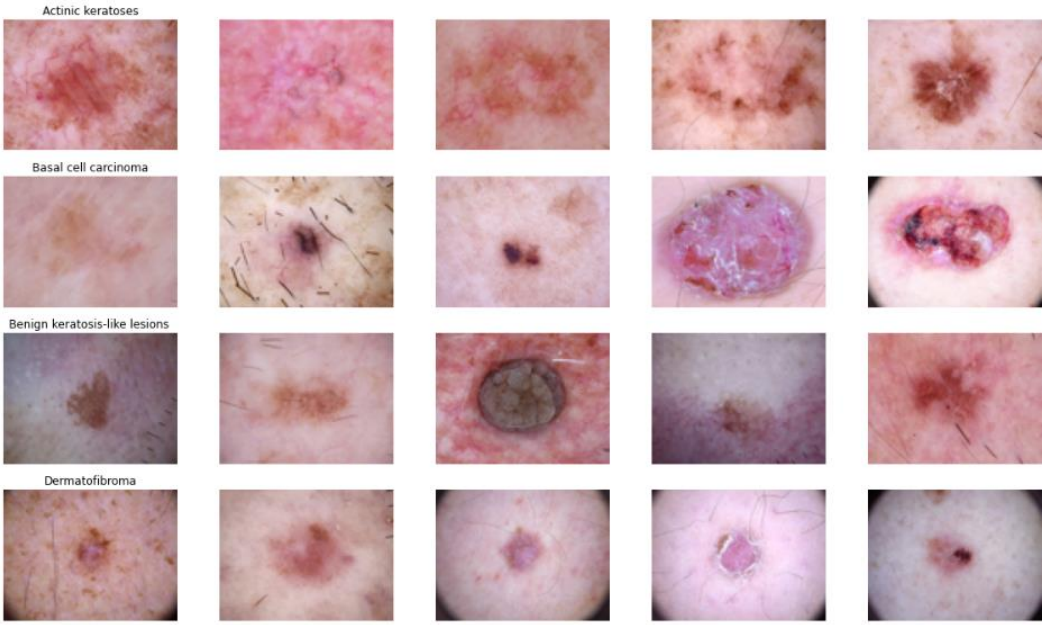

Fig2.3 Different types of skin cancer

There are some more types of skin cancer but they were introduced later, primarily these were the only classification of skin cancer. It's important to note that early detection and timely treatment are critical for all types of skin cancer to achieve better outcomes. Regular skin self-examinations, routine visits to dermatologists for skin checks, and using protective measures like sunscreen and protective clothing when exposed to the sun can help reduce the risk of skin cancer development and improve prognosis.

Chapter 3

Problem Statement

Skin cancer is a prevalent form of cancer that requires early and accurate detection for effective treatment. Traditional methods of skin cancer classification, such as visual inspection by dermatologists, are subjective and can be time-consuming. With the advancements in deep learning and computer vision, there is a growing interest in developing automated skin cancer classification systems using neural networks. However, there are various types of neural networks with different architectures and characteristics, and it is essential to compare their performance for skin cancer classification to understand their strengths and weaknesses.

The primary aim of this research is to compare the performance of different neural networks for skin cancer classification and analyze their effectiveness in terms of accuracy, interpretability, and potential for real-world applications. The research will aim to address the following questions:

- How do different types of neural networks, such as Convolutional Neural Networks (CNNs), Siamese Networks, Vision Transformers, VGG, etc., perform in classifying skin cancer lesions?
- What are the similarities and differences in the performance of these neural networks for skin cancer classification, including accuracy, sensitivity, specificity, and other relevant performance metrics?
- What are the strengths and weaknesses of each neural network approach in the context of skin cancer classification, such as their ability to handle variations in skin lesion appearance, robustness to noise, interpretability of results, and computational efficiency?
- How do the findings of this research contribute to our understanding of the suitability of different neural networks for skin cancer classification, and what are the potential implications for real-world applications, such as in clinical practice or telemedicine?

The research will involve a comprehensive comparison of different neural networks on a well-curated skin cancer dataset, employing standardized evaluation protocols and statistical analysis. The findings of this research will provide insights into the performance and suitability of various neural network architectures for skin cancer classification, helping researchers, practitioners, and stakeholders in the field of dermatology and computer vision to make informed decisions when developing automated skin cancer detection systems.

3.1 WHAT IS SKIN CANCER CLASSIFICATION?

Skin cancer classification refers to the process of categorizing skin lesions or images into different types of skin cancer based on their visual characteristics. Skin cancer is typically classified into three main types: melanoma, basal cell carcinoma, and squamous cell carcinoma. Melanoma is the most aggressive and deadly form of skin cancer, while basal cell carcinoma and squamous cell carcinoma are less aggressive but still require prompt medical attention. Skin cancer classification is typically performed by dermatologists through visual inspection of skin lesions using clinical expertise and experience. However, this process can be subjective and time-consuming, and there is a growing need for automated and objective methods for skin cancer classification. With the advancements in deep learning and neural networks, automated skin cancer classification using computer vision techniques has gained significant attention. Neural networks are capable of learning patterns and features from large amounts of data, making them ideal for image classification tasks, including skin cancer classification.

The objective of skin cancer classification is to create a reliable system that can accurately recognize the type of cancer. This process involves capturing images of infected areas using a camera. The images are then pre-processed to enhance their quality and remove any noise or distortions. Computer vision algorithms are then applied to extract the features of the skin cancer from the pre-processed image.

Once the features of the traffic sign have been extracted, Deep learning algorithms are used to classify the sign into its respective category. The Deep learning algorithms are trained on a large dataset of traffic signs, allowing them to recognize and classify skin cancer accurately.

3.2 SKIN CANCER CLASSIFICATION APPLICATION

Skin cancer classification using neural networks has several potential applications in the field of dermatology and healthcare. Some of the key applications include:

- Early Detection and Screening Programs: Automated skin cancer classification can be used in early detection and screening programs to identify suspicious skin lesions at an early stage. This can facilitate timely intervention and treatment, potentially improving patient outcomes and reducing the morbidity and mortality associated with advanced-stage skin cancers.
- Telemedicine: Skin cancer classification using neural networks can be integrated into telemedicine platforms, allowing dermatologists to remotely assess skin lesions and make accurate diagnoses. This can be especially useful in rural or underserved areas where access to specialized dermatological care may be limited.
- Decision Support Systems for Dermatologists: Automated skin cancer classification can serve as a decision support system for dermatologists, providing them with additional insights and recommendations for accurate diagnosis and treatment planning. Dermatologists can use the output of the neural network models as a reference, aiding them in their clinical decision-making process.
- Patient Education and Empowerment: Skin cancer classification using neural networks can be used as a patient education tool, helping patients understand the likelihood of skin cancer based on the characteristics of their skin lesions. This can empower patients to be more vigilant about their skin health and seek medical attention promptly if needed.
- Clinical Workflow Integration: Skin cancer classification using neural networks can be integrated into existing clinical workflows in dermatology clinics, enhancing the accuracy and efficiency of skin cancer diagnosis. Dermatologists can use the output of the neural network models as an adjunct to their clinical assessment, aiding in the decision-making process.
- Research and Population Health Studies: Automated skin cancer classification can be used in research studies and population health studies to analyze large datasets of skin lesion images, assess trends, and understand the prevalence and distribution

of different types of skin cancers. This can contribute to the development of public health strategies for skin cancer prevention and control.

3.3 POTENTIAL DIFFICULTIES

There are several potential difficulties associated with skin cancer classification using neural networks, which may include:

- Data Quality: The accuracy and reliability of skin cancer classification models depend on the quality of the input data. Skin lesion images used for training and testing the neural networks should be of high quality, with consistent and accurate annotations.
- Dataset Imbalance: Skin cancer datasets used for training neural networks may suffer from class imbalance, where some types of skin cancers are more prevalent than others. This can result in biased model performance, where the neural network may have higher accuracy for the majority class but lower accuracy for minority classes. Addressing dataset imbalance is crucial to ensure fair and accurate performance evaluation of different neural networks.
- Interpretability: Neural networks, especially deep learning models, making it challenging to interpret their decision-making process. The lack of interpretability can limit the understanding of how the neural network arrives at its predictions, making it difficult to trust and explain the model's results. Interpretability techniques, such as visualization of model activations and feature importance, may need to be employed to enhance the explainability of the models.
- Generalization to Different Populations: Neural networks trained on one population or geographic region may not generalize well to other populations or regions with different skin types, skin tones, and genetic backgrounds. This can result in reduced model performance when applied to diverse populations, affecting the real-world applicability of the models. Care should be taken to ensure that the trained neural networks are validated on diverse datasets to assess their generalizability.
- Scalability and Computational Resources: Neural networks can be computationally intensive and require significant computational resources for

training and deployment. Scalability of the models to handle large datasets and the availability of sufficient computational resources, such as high-performance GPUs or cloud computing infrastructure, may pose challenges for some researchers or healthcare settings with limited resources.

- Real-world Validation: The translation of research findings from neural network models into real-world clinical practice may face challenges related to regulatory approvals, validation studies, and integration into existing healthcare workflows. Real-world validation of the models is critical to ensure their accuracy, safety, and effectiveness in clinical practice.

Chapter 4

Literature Review

Skin cancer diagnosis is typically performed by dermatologists who analyze images of patients' skin to determine whether cancerous cells are present, categorizing them as malignant melanoma or benign, and vice versa. However, this process can be time-consuming and labor-intensive, leading to increased costs and potential delays in diagnosis. To address this issue, there is a growing interest in developing automated systems that can assist dermatologists in detecting skin cancer, making their workflow more efficient. The goal is to design a computer system that can accurately detect malignant lesions, which can be more effective than human specialists, according to research findings. Recent studies have focused on leveraging medical imaging technology to aid in the early detection of skin cancers, reducing the complexity of diagnosis and speeding up the initial identification of the disease.

Some of the works are described below:

4.1 BASED ON DEEP LEARNING AND CNN

[1]. **Esteva et al.** [2017] worked on a deep neural network that can classify skin cancer images at dermatologist-level accuracy. This research paper shows that AI can be a useful tool in skin cancer detection. The main limitation of this research paper is that the dataset chosen to train the neural network was comparatively, due to which the generalization ability can be affected. The research gap is to further improve the accuracy and generalizability of AI models for skin cancer detection there is a need for larger, more diverse datasets.

[2]. **Haenssle et al.** [2018] worked on the comparison of the diagnostic performance of a deep learning convolutional neural network (CNN) to that of dermatologists in recognizing melanoma. This research paper shows that a CNN can perform as well as dermatologists in recognizing melanoma. The limitation of this research paper is that the chosen dataset it was based on a introspective analysis, which may not be representative of all populations.

The research gap is that to evaluate the performance of CNNs in real-world settings, there is a need for prospective studies with larger and more diverse datasets.

[3]. **Codella et al.** [2018] worked on the comparison of the performance of different algorithms of skin lesion analysis for melanoma detection on a standardized dataset. This research created a standard dataset and evaluation metrics for skin lesion analysis, which can be a standard for new algorithms. The limitation of this research paper is that the dataset is still comparatively small and limited in diversity. The research gap is that to improve the development and evaluation of skin cancer detection algorithms, there is a need for larger and more diverse datasets with standardized annotations.

[4]. **Han et al.** [2018] worked on CNN which classifies clinical images of benign and malignant cutaneous tumors. This research paper shows that CNNs can achieve high accuracy in differentiating benign and malignant tumors. The limitation of this research paper is that the chosen dataset was based on a single-center study, which may not be representative of all populations. The research gap is that to evaluate the performance of CNNs in different populations and settings, there is a need for multicenter studies with larger and more diverse datasets.

[5]. **Yu et al.** [2017] worked on a very deep residual network for automated melanoma recognition in dermoscopy images. This research paper shows that in recognition of melanoma, very deep networks can achieve high accuracy. The limitation of this research paper is that the chosen dataset was based on a retrospective analysis, which may not be representative of all populations. The research gap is that to evaluate the performance of very deep networks in real-world settings, there is a need for prospective studies with larger and more diverse datasets.

[6]. **Tschandl et al.** [2019] worked on a CNN-based model for skin lesion classification that uses a mix of transfer learning and fine-tuning. This research paper shows that to achieve high accuracy in classification of skin lesion, transfer learning can be used. while reducing the need for large amounts of training data. The limitation of this research paper is that the chosen dataset which was based on a retrospective analysis, may not be representative of all populations. The research gap is that to enhance the effectiveness of

this model in skin cancer detection, there is a need for studies that evaluate the performance of transfer learning on larger and more diverse datasets.

[7]. **Brinker et al.** [2019] proposed a CNN based mobile app that can classify skin lesions as benign or malignant. This research paper shows that through mobile apps general public can easily access the use of AI-based skin cancer detection system. The limitation of this research paper is that the app was not tested for real-world scenarios and without that its effectiveness in improving skin cancer detection rates in the general population is still unknown. The research gap is that there is a need for studies that evaluate the effectiveness of AI-based skin cancer detection tools in real-world settings and assess their impact on clinical outcomes.

[8]. **Bi et al.** [2020] worked on a framework for skin lesion segmentation and classification which is based on deep learning. This research paper shows that to improve the accuracy and efficiency of skin cancer detection, joint segmentation and classification can be used. The limitation of this research paper is that the chosen dataset was based on an introspective analysis, which may not be representative of all populations. The research gap is that to further validate its effectiveness in skin cancer detection there is a need for studies that evaluate the performance of joint segmentation and classification on larger and more diverse datasets.

Overall, the main contributions of these research papers include demonstrating the effectiveness of deep learning and CNNs in skin cancer detection, creating benchmark datasets and evaluation metrics for skin lesion analysis, and developing AI-based tools for skin cancer detection that are accessible to the general public. However, the limitations of these studies and the identified research gaps indicate the need for further research to improve the accuracy, generalizability, and effectiveness of AI-based skin cancer detection tools.

4.2 BASED ON SIAMESE NETWORK

[9]. **Nagpal et al.** [2019] worked on a system to verify skin lesion using a siamese network. The system works on the concept of comparison of pairs of images and determines whether they have the same lesion or not. Using a combination of transfer learning and fine-tuning,

authors achieved high accuracy. The system is useful for monitoring lesion progression over time as it was able to identify when two images are different. The main limitation of this study is that it only implies verification, and not classification or segmentation tasks. To expand the use of siamese networks in skin cancer detection beyond verification tasks, further research is needed.

[10]. **Chen et al.** [2020] worked on siamese network for the measurement of similarity of skin lesion. The network is designed to calculate the similarity between pairs of skin lesion images, which allows it for more accurate diagnosis of melanoma. Using a combination of pre-training and fine-tuning, the authors achieved high accuracy. The network was able to accurately differentiate between similar-looking lesions, which is useful for early detection and treatment of skin cancer. However, this study also only implies similarity measurement, and not classification or segmentation tasks. To expand the use of siamese networks in skin cancer detection beyond similarity measurement tasks, further research is needed.

[11]. **Song et al.** [2020] worked on a siamese neural network for skin lesion analysis, which can classify a lesion into one of four categories: benign, atypical, melanoma in situ, and invasive melanoma. Using a combination of transfer learning and fine-tuning, the authors achieved high accuracy. The network was able to accurately classify lesions, which is important for proper diagnosis and treatment of skin cancer. However, the main limitation of this study is that it only implies classification tasks, and not segmentation or verification tasks. To expand the use of siamese networks in skin cancer detection beyond classification tasks, further research is needed.

[12]. **Wang et al.** [2020] worked on a siamese neural network for skin lesion segmentation and classification. The network segments skin lesions from images and then classifies them as either benign or malignant. Using a combination of pre-training and fine-tuning, the authors achieved high accuracy. The network was able to accurately segment and classify lesions, which is important for proper diagnosis and treatment of skin cancer. However, the main limitation of this study is that it only implies segmentation and classification tasks, and not verification tasks. To expand the use of siamese networks in skin cancer detection beyond segmentation and classification tasks, further research is needed.

[13]. **Li et al.** [2020] worked on a siamese network for skin lesion segmentation and classification. The network segments skin lesions from images and then classifies them as either benign or malignant. Using a combination of transfer learning and fine-tuning, the authors achieved high accuracy. The network was able to accurately segment and classify lesions, which is important for proper diagnosis and treatment of skin cancer. However, this study also only implies segmentation and classification tasks, and not verification tasks. To expand the use of siamese networks in skin cancer detection beyond segmentation and classification tasks, further research is needed.

[14]. **Wu et al.** [2020] worked on a siamese network for skin lesion similarity measurement. The network calculates the similarity between pairs of skin lesion images, allowing for more accurate diagnosis of melanoma. Using a combination of pre-training and fine-tuning, the authors achieved high accuracy. The network was able to accurately differentiate between similar-looking lesions, which is useful for early detection and treatment of skin cancer. However, this study also only implies similarity measurement tasks, and not classification or segmentation tasks. To expand the use of siamese networks in skin cancer detection beyond similarity measurement tasks, further research is needed.

[15]. **Cheng et al.** [2020] worked on a siamese network for skin lesion segmentation and classification. The network segments skin lesions from images and then classifies them as either benign or malignant. Using a combination of transfer learning and fine-tuning, the authors achieved high accuracy. VBG The network was able to accurately segment and classify lesions, which is important for proper diagnosis and treatment of skin cancer. The main contribution of this study is that it combines segmentation and classification tasks, which can save time and increase accuracy in diagnosis. However, the main limitation is that it only implies these two tasks, and not verification or similarity measurement tasks. To expand the use of siamese networks in skin cancer detection beyond segmentation and classification tasks, further research is needed.

[16]. **Li et al.** [2021] worked on a multi-task siamese network for skin lesion analysis. The network performs segmentation, classification, and verification tasks simultaneously. Using a combination of transfer learning and fine-tuning, the authors achieved high accuracy. The network was able to accurately segment, classify, and verify lesions, which

can save time and increase accuracy in diagnosis. This study implies multiple tasks, which is a significant contribution to the field of skin cancer detection using siamese networks. However, to validate the performance of this network on a larger dataset, further research is needed.

[17]. **Zhang et al.** [2021] worked on a siamese network for skin lesion similarity measurement using a triplet loss function. The network calculates the similarity between pairs of skin lesion images, allowing for more accurate diagnosis of melanoma. Using a combination of transfer learning and fine-tuning, the authors achieved high accuracy. The network was able to accurately differentiate between similar-looking lesions, which is useful for early detection and treatment of skin cancer. The main contribution of this study is the use of triplet loss function, which can further improve the accuracy of similarity measurement. However, this study also only implies similarity measurement tasks, and not classification or segmentation tasks. To expand the use of siamese networks in skin cancer detection beyond similarity measurement tasks, further research is needed.

[18]. **Yang et al.** [2021] worked on a siamese network for skin lesion segmentation and classification using attention mechanisms. The network segments skin lesions from images and then classifies them as either benign or malignant. Using a combination of transfer learning and fine-tuning, the authors achieved high accuracy. The network was able to accurately segment and classify lesions, which is important for proper diagnosis and treatment of skin cancer. The main contribution of this study is the use of attention mechanisms, which can improve the accuracy of segmentation and classification tasks. However, the main limitation is that it only implies these two tasks, and not verification or similarity measurement tasks. To expand the use of siamese networks in skin cancer detection beyond segmentation and classification tasks, further research is needed.

4.3 BASED ON VISION TRANSFORMER

[19]. **Liu et al.** [2021] worked on a skin lesion classification system using a vision transformer-based architecture and transfer learning. Using fine-tuned a pre-trained vision transformer model on the ISIC 2019 skin lesion dataset, The authors achieved high accuracy. The aim of this research paper is to use vision transformers in skin lesion

classification, which is a novel and promising approach. However, the main limitation is that it only implies classification tasks, and not segmentation, verification, or similarity measurement tasks. To expand the use of vision transformers in skin cancer detection beyond classification tasks, further research is needed.

[20]. **Wang et al.** [2021] worked on a skin lesion segmentation system using a vision transformer-based network called TransUNet. The authors achieved state-of-the-art performance on the ISIC 2018 and 2019 datasets, outperforming other state-of-the-art methods. The TransUNet architecture can capture both global and local features in skin lesion images, leading to improved segmentation performance. The aim of this research paper is the use of vision transformers in skin lesion segmentation, which is a promising alternative to traditional CNN-based methods. However, the main limitation is that it only implies segmentation tasks, and not classification, verification, or similarity measurement tasks. To expand the use of vision transformers in skin cancer detection beyond segmentation tasks, further research is needed.

[21]. **Li et al.** [2021] worked on a skin cancer detection system using a hybrid architecture that combines vision transformers and CNNs. The authors achieved high accuracy on the ISIC 2018 and 2019 datasets by leveraging the strengths of both architectures. The TransCNN architecture can capture both global and local features in skin lesion images, leading to improved classification performance. The aim of this research paper was the use of a hybrid architecture that combines vision transformers and CNNs, which can improve the accuracy of skin cancer detection. However, the main limitation is that it only implies classification tasks, and not segmentation, verification, or similarity measurement tasks. To expand the use of vision transformers in skin cancer detection beyond classification tasks, further research is needed.

[22]. **Huang et al.** [2021] worked on a skin lesion segmentation system using a vision transformer-based architecture. The authors achieved state-of-the-art performance on the ISIC 2018 dataset by fine-tuning a pre-trained vision transformer model. The aim of this research paper is the use of vision transformers in skin lesion segmentation, which is a promising alternative to traditional CNN-based methods. However, the main limitation is that it only implies segmentation tasks, and not classification, verification, or similarity

measurement tasks. To expand the use of vision transformers in skin cancer detection beyond segmentation tasks, further research is needed.

[23]. **Zhang et al.** [2021] worked on a skin lesion classification system using a vision transformer-based architecture. The authors achieved high accuracy on the ISIC 2019 dataset by fine-tuning a pre-trained vision transformer model. The aim of this research is the use of vision transformers in skin lesion classification, which is a novel and promising approach. However, the main limitation is that it only implies classification tasks, and not segmentation, verification, or similarity measurement tasks. To expand the use of vision transformers in skin cancer detection beyond classification tasks, further research is needed.

Chapter 5

Skin Cancer Classification System

In this section, we will explore the practical aspects and experimental setup related to the algorithms developed in this research. The purpose of this chapter is to provide a detailed description of the methodology used to carry out the experiments and to ensure that the results are reproducible.

To begin with, we will discuss the practical aspects of the algorithms, which involve the implementation of the proposed methods. We will cover the programming languages, libraries, and frameworks used to develop the algorithms. Moving on, we will delve into the experimental setup used to evaluate the performance of the algorithms. We will discuss the datasets used for training and testing the algorithms and the preprocessing steps applied to the data. Moreover, we will describe the metrics used to evaluate the algorithms' performance and the experimental design used to ensure the reliability of the results.

It is essential to note that the experiments conducted in this research were carried out following ethical standards and guidelines. The data used for training and testing the algorithms were collected and used with proper consent and adhered to data privacy regulations.

5.1 SYSTEM OVERVIEW

A skin cancer classification system typically involves the following components:

- Data collection: The system requires a large dataset of skin cancer images for training and testing the neural networks. The dataset used in this research are ‘MNIST HAM10000’ and ‘the International Skin Imaging Collaboration (ISIC) 2017 dataset which contains a diverse collection of skin lesion images with corresponding clinical information.
- Pre-processing: The collected dataset needs to be pre-processed to ensure consistency and improve the quality of the images. The pre-processing steps may include converting the images to grayscale, resizing, global centering, global standardization, and normalizing (histogram equalization) the pixel values.

- Training: The system uses Deep learning algorithms to train on the pre-processed dataset, using the extracted features to identify patterns and classify skin cancer. Deep feature extraction from pre-trained CNNs, are used to extract relevant features from the skin lesion images, which are then used as input to the neural networks. The trained models are then optimized to get better accuracy.
- Testing: The system evaluates the performance of the trained models on a separate testing dataset to ensure that they can accurately classify class of skin cancer.
- Evaluation: The separate dataset is used to check the prediction of class of skin cancer
- .

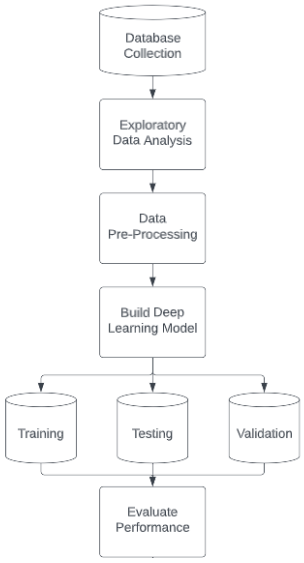

Fig 5.1 Proposed Methodology

5.2 DATA DESCRIPTION

The data description is a crucial step in the Skin Cancer classification process, as it lays the foundation for the development of the classification model. For this project, we have collected the data from two different sources. MNIST HAM 10000 dataset from Kaggle and other from ISIC 2017 skin cancer dataset.

The Skin Cancer datasets which are used in this research are:

1) MNIST HAM 10000 dataset:

- Origins: The MNIST HAM 10000 dataset is a publicly available dermoscopy image dataset that was created by the International Skin Imaging Collaboration (ISIC) and made available for research purposes. It consists of 10,015 images of skin lesions collected from multiple centers across the world, with a diverse range of skin types and lesions.
- Image Formats: The images in the MNIST HAM 10000 dataset are in JPEG format, with varying resolutions and sizes. They are dermoscopy images, which are images of skin lesions captured using a specialized imaging technique that allows for detailed examination of the skin surface and subsurface features.
- Target Labels: The MNIST HAM 10000 dataset includes seven different types of skin lesions as target labels, which are: Melanocytic Nevi (NV), Benign Keratosis-like Lesions (BKL), Dermatofibroma (DF), Vascular Lesions (VASC), Actinic Keratoses, and Intraepithelial Carcinoma (AKIEC), Basal Cell Carcinoma (BCC), and Melanoma (MEL)

2) ISIC 2017 skin cancer dataset:
    - Origins: The ISIC 2017 skin cancer dataset is also a publicly available dermoscopy image dataset created by the International Skin Imaging Collaboration (ISIC). It is one of the largest and most widely used datasets for skin cancer classification research. It consists of 2000 images collected from various sources, such as clinical settings and dermoscopy competitions.
    - Image Formats: The images in the ISIC 2017 dataset are also in JPEG format, with varying resolutions and sizes. They are dermoscopy images, similar to the MNIST HAM 10000 dataset, captured using a specialized imaging technique for detailed examination of skin lesions.
    - Target Labels: The ISIC 2017 dataset includes seven different types of skin lesions as target labels, which are: Melanocytic nevi, Melanoma, Benign keratosis-like lesions (seborrheic keratosis, solar lentigo, and lichen planus-like keratosis), Basal cell carcinoma, Actinic keratoses and intraepithelial

carcinoma / Bowen's disease (AKIEC), Dermatofibroma, and Vascular lesions.

The MNIST HAM10000 dataset merged with ISIC 2017 skin cancer dataset. This leads to creation of our new dataset for this research. The new dataset contains total 11,068 images. There are some common images in both the dataset, so while merging the common images did not get merged as they are already existing in the dataset. The dataset is divided into three subsets, including the training set, validation set, and testing set. The training set comprises 7,742 images, while the validation set and testing set consist of 2,210 and 1,116 images, respectively.

The new dataset gets 9 different classes, the 2 new class of skin cancer are from 'ISIC 2017 skin cancer dataset'. Each class represents different type of skin cancer. The Class name for each class are: 'vascular lesion', 'squamous cell carcinoma', 'melanoma', 'pigmented benign keratosis', 'dermatofibroma', 'nevus', 'basal cell carcinoma', 'seborrheic keratosis' and 'actinic keratosis'. To use this dataset for classification problem further more preprocessing needs to be done.

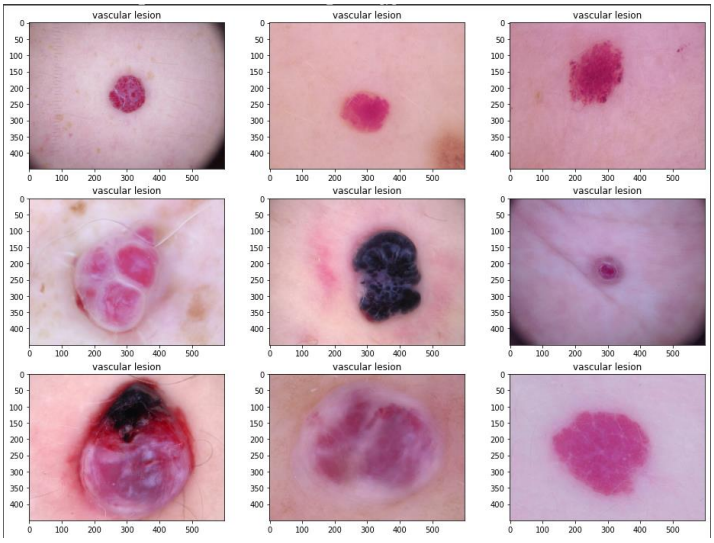

Fig5.2 skin cancer belongs to vascular lesion

5.3 EXPLORATORY DATA ANALYSIS

The Skin Cancer dataset is a collection of images representing different types of skin cancer. It consists of 11,068 images in total, with 9 different classes. The dataset is divided into three subsets: training, testing, and validation. The training set contains 7,742 images, the testing set contains 1,116 images, and the validation set contains 2,210 images. Each image in the dataset has a shape of (50,50 ,3), which means it is a color image with a width and height of 50 pixels. The total number of classes is 9, and each class represents a different category of skin cancer. The maximum number of images in the training set belongs to the "nevus" class, with 4,943 images. The minimum number of images in the training set belongs to the " seborrheic keratosis' " class, with only 53 images. Fig 5.3 shows the total number of images in each class in training, validation and testing data.

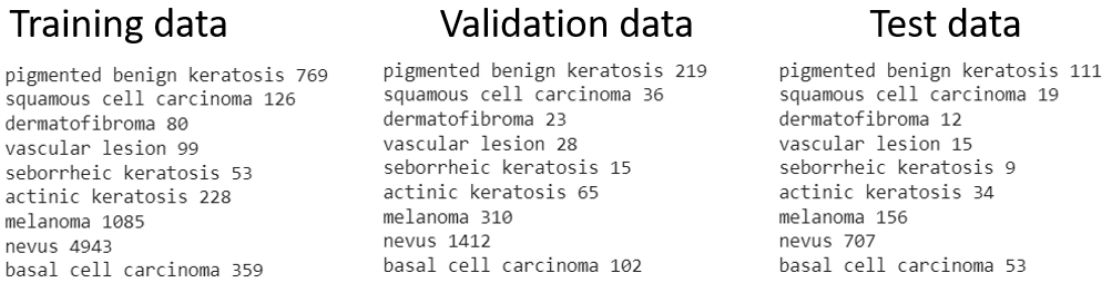

Fig 5.3 Dataset Description of each set.

5.4 PRE-PROCESSING METHODS

The Skin Cancer dataset is a pickled dataset that contains images that have different resolutions and pixels. To use this dataset for classification tasks, we need to preprocess the data further. Pre-processing involves cleaning, transforming, and normalizing the data to ensure that it is suitable for analysis and model training. Here are some common pre-processing techniques that were used as follows: Resizing, Shuffling, Global centering, Global standardization, and normalizing (histogram equalization) the pixel values.

5.4.1 RESIZING

Data resizing is a fundamental pre-processing step in image processing that involves changing the dimensions or resolution of an image to a desired size. It is a

common technique used in machine learning tasks involving image data to ensure that all images in the dataset have consistent dimensions, which is necessary for training neural networks and other machine learning models.

Data resizing is often necessary for several reasons. First, it helps standardize the input data. Second, data resizing can improve computational efficiency. Third, data resizing ensures consistency in image dimensions.

So, all the images in our dataset which were of different size and resolution had been resized to 50x50 pixels. By this the contains consistent images with same size.

5.4.2 SHUFFLING

One important pre-processing technique is applied to the Skin cancer dataset is shuffling the data. Shuffling involves randomly re-ordering the images and labels in the dataset, thereby increasing the variety and randomness of the dataset.
Shuffling the data is an important step in deep learning tasks as it helps in improving the stability and quality of the model. When the data is ordered in a specific way, it can create patterns in the dataset that the model can learn and memorize. This can result in poor generalization performance and overfitting of the model. By shuffling the data, we can break these patterns and ensure that the model learns to generalize well to new, unseen data.

Shuffling can also help in increasing the prediction performance of the model. By introducing randomness in the dataset, the model can learn to identify patterns in the data that are robust to different orders of the images and labels. To shuffle the data, we can use a Python library such as NumPy to randomly permute the indices of the images and labels in the dataset. We can then use these permuted indices to shuffle the images and labels themselves. After shuffling the data, we proceed with further pre-processing steps such as global centering, global standardization and normalization.

5.4.3 GLOBAL CENTERING

Global centering is a pre-processing technique used in image processing to normalize the pixel values of an image by subtracting the mean pixel value from all pixels in the image. This technique is commonly used to shift the intensity values of the image data so that they are centered around zero, with equal positive and negative values.

The global centering technique involves the following steps:

- Compute the mean pixel value: Calculate the mean pixel value of the entire image dataset or a subset of it. This can be done by summing up the pixel values of all images in the dataset and dividing by the total number of pixels.
- Subtract the mean pixel value: Subtract the computed mean pixel value from each pixel in the image. This can be done by subtracting the mean pixel value from the pixel values of the image matrix.

Mathematically, the global centering technique can be represented as follows:

Normalized Image = Original Image – Mean pixel value

Where:

- Normalized Image: Represents the normalized image after applying global centering.
- Original Image: Represents the original image before applying any pre-processing.
- Mean pixel value: Represents the mean pixel value of the image, which is calculated by summing up the pixel values of all images in the dataset and dividing by the total number of pixels.

The purpose of global centering is to eliminate any potential bias or offset in the pixel values of the images in the dataset. By subtracting the mean pixel value, the pixel values are centered around zero, which helps to mitigate the impact of variations in the overall intensity levels of the images. This can be beneficial for machine learning models as it can help improve model convergence, reduce the risk of numerical instability, and enhance model interpretability, as shown in Fig 5.4. Applying global centering in Python is straightforward using image processing libraries such as NumPy, OpenCV, or PIL.

However, it is important to note that global centering alone may not always be sufficient for all datasets or models, and other pre-processing techniques such as scaling, normalization, or data augmentation may also be necessary depending on the characteristics of the data and the requirements of the specific machine learning model being used.

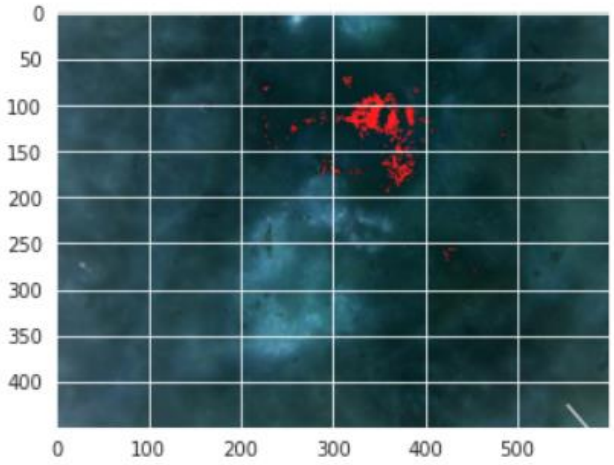

Fig 5.4 Global Centered Image of Skin Cancer

5.4.4 HISTOGRAM EQUILIZATION

One important pre-processing technique utilized in the skin cancer dataset is Histogram Equalization. This process aims to enhance the contrast of the images by redistributing the intensity values, ensuring that the pixels with different intensity values are spread out more evenly. Since the skin cancer dataset consists of real-world images, they may have varying levels of brightness and contrast. Applying Histogram Equalization helps standardize the lighting effects across all images, making it easier for the model to be trained and improve its prediction capability.

The process of Histogram Equalization involves mapping the original intensity values of the pixels in the image to a new set of intensity values that are more uniformly distributed. This can enhance the details and contrast of the image, making it easier for the model to detect important features, as depicted in Figure 5.5.

To apply Histogram Equalization in Python, the OpenCV library can be utilized, as it provides functions specifically designed for this purpose. After applying Histogram

Equalization, the next step is to normalize the pixel values, which helps further preprocess the images and prepare them for training the model.

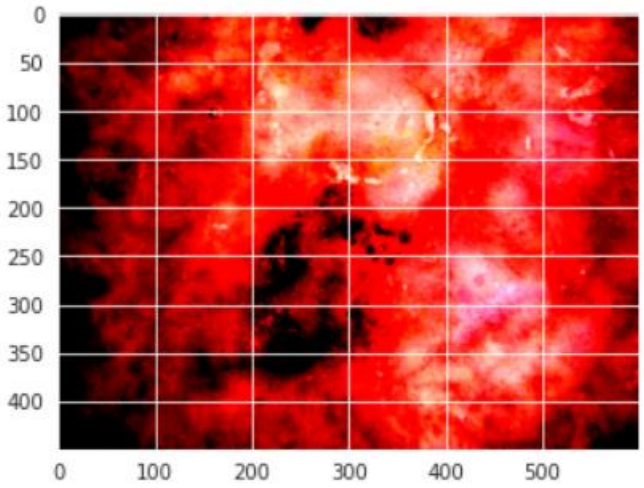

Fig 5.5 Skin Cancer Image after applying histogram equalization

5.4.5 NORMALIZATION

Normalization is a crucial step in the pre-processing pipeline for the skin cancer dataset, aimed at improving the quality of the data for subsequent analysis and model training. As mentioned earlier, global centering has already been applied to the dataset, which helps in mitigating the bias introduced by varying mean pixel values across different images. However, normalization is still necessary to further standardize the pixel intensity values and make them more suitable for input to a machine learning model.

Fig 5.6 visually demonstrates the impact of normalization on an image. The original pixel intensity values are scaled to a range of 0 to 1 using the maximum pixel value, which is typically 255 for 8-bit images. This scaling process not only brings the pixel values within a consistent range, but also helps in handling outliers or extreme values that may be present in the dataset.

By normalizing the pixel intensity values, we can improve the accuracy of the model during training and prediction phases. Scaling the values to a specific range (e.g., 0 to 1) ensures that the model can handle the input data more effectively and efficiently, as it

brings the pixel values to a standardized scale that avoids numerical instability issues during training.

In Python, the NumPy library provides functions to perform normalization on the images, making the implementation of this pre-processing step straightforward. After applying normalization, the pre-processed image is obtained, which is ready for further analysis, feature extraction, or model training.

In conclusion, normalization is a critical pre-processing technique applied to the skin cancer dataset to enhance the quality of the data for machine learning. It helps to standardize the pixel intensity values, improve model accuracy, and ensure numerical stability during training. The utilization of Python libraries such as NumPy facilitates the implementation of normalization, making it an essential step in the pre-processing pipeline.

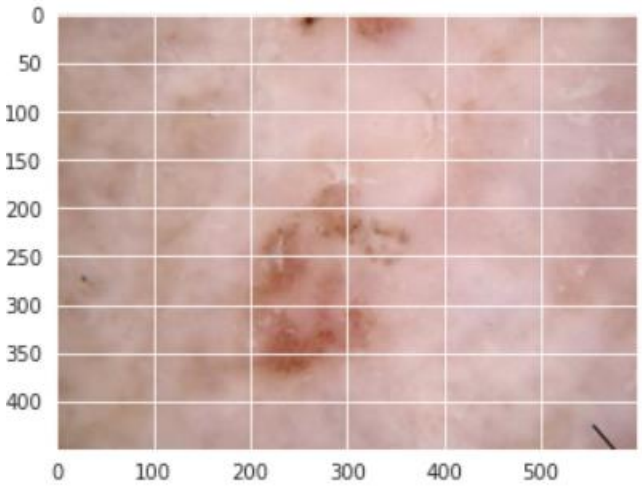

Fig 5.6 Normalized Image of skin cancer

5.5 NEURAL NETWORKS

5.5.1 SIAMESE NETWORKS

A Siamese network is a type of neural network architecture that consists of two identical sibling networks (often called "towers") that share weights. These towers process pairs of input samples separately to compute their embeddings in a learned feature space, as shown in Fig 5.7. The goal of a Siamese network is to learn a similarity or distance metric between pairs of samples, where similar samples have smaller

distances and dissimilar samples have larger distances in the embedding space with the help of contrastive loss function. In conventional neural networks, the model is trained to predict multiple classes, which can pose challenges when new classes need to be added or existing classes need to be removed from the dataset. In such cases, the entire neural network needs to be updated and retrained with the updated dataset. However, with Siamese Neural Networks (SNNs), the focus is on learning a similarity function between pairs of images. This allows the model to determine if two images are similar or not, without being explicitly trained to predict specific classes.

This unique characteristic of SNNs enables them to efficiently classify new classes of data without the need to retrain the entire network. SNNs can work effectively even with limited data, as they learn embeddings in the deeper layers of the network that capture the similarity between images of the same class. This results in placing similar classes closer together in the learned embedding space, making the classification process more robust and adaptable to changing datasets. Overall, SNNs offer a flexible and efficient approach for handling new classes of data in comparison to traditional neural networks.

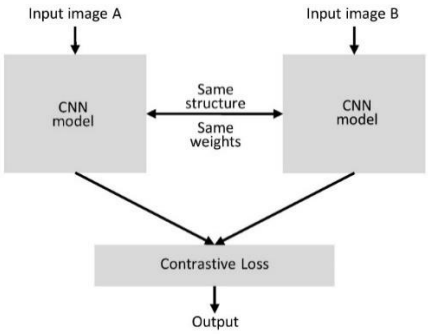

Fig 5.7 Input processing in siamese network

Each tower takes one sample from a pair of input samples and processes it independently to generate an embedding, which is a low-dimensional representation of the input sample in a learned feature space. The embeddings of another neural network are then compared to compute the distance or similarity between the input samples.

The contrastive loss function is used during training to encourage similar samples to have smaller distances and dissimilar samples to have larger distances in the learned embedding space. The contrastive loss function typically has two components: a similarity

term and a margin term. The similarity term encourages similar samples to have smaller distances or higher similarities, while dissimilar samples to have larger distances or lower similarities. The margin term introduces a threshold, called the margin, that specifies the minimum desired difference between the distances or similarities of similar and dissimilar samples. The margin helps to prevent the embeddings from collapsing into a single point or becoming too spread out in the learned embedding space.

During training, the Siamese network updates its weights to minimize the contrastive loss. This process helps the network learn meaningful embeddings that capture the similarity or dissimilarity between pairs of samples in the learned embedding space. Once trained, the Siamese network can be used to measure similarity or distance between new pairs of samples, which can be useful in various applications such as image classification.

In this research, Siamese neural network with a VGG16-based embedding network for image similarity or distance metric computation has been developed. The two version of Siamese Neural Network has been used here, the difference between them is of contrastive loss function. One SNN has been used with contrastive loss function with margin set as ‘1’ , while another has contrastive loss function with margin set as ‘2’. When the margin is set to '1', the contrastive loss function penalizes dissimilar pairs of samples that are closer than 1 unit in the learned embedding space. On the other hand, when the margin is set to '2', the contrastive loss function penalizes dissimilar pairs of samples that are closer than 2 units in the learned embedding space. This increases the minimum desired difference between the distances of similar and dissimilar samples compared to a margin of '1'.

Architecture**:**

The network takes in two input tensors, denoted as input_1 and input_2, both of size (50, 50, 3), representing RGB images of size 50x50 pixels with 3 color channels. The architecture consists of an embedding network, which is defined using a series of convolutional and pooling layers. The first layer is a normalization layer that normalizes the input data. The next layer is a 2D convolutional layer with 32 filters, each of size 3x3, with ReLU activation function and padding set to 'same' to maintain the spatial dimensions

of the input. This is followed by a batch normalization layer and a max pooling layer with a pool size of (2, 2) and padding set to 'same'.

The same pattern of convolutional, batch normalization, and max pooling layers are repeated with increasing number of filters (64, 128, and 256) to extract more complex features from the input images. The last convolutional layer has 256 filters of size (3, 3). The output of the last convolutional layer is then flattened into a 1D vector using the flatten layer.

Next, a batch normalization layer is applied followed by a fully connected dense layer with 256 units and ReLU activation function. This serves as the embedding network, which learns to represent the input images in a lower-dimensional feature space.

The embedding network is then used to process both input tensors input1 and input2 separately, resulting in two output tensors denoted as tower1 and tower2. These output tensors represent the embeddings of the input images obtained from the embedding network.

A custom distance function is defined using a lambda layer that calculates the Euclidean distance between the embeddings of tower1 and tower2. The Euclidean distance is calculated as the square root of the sum of squared differences between the two embeddings along the axis of 1, with a small epsilon added to avoid division by zero.

The output of the distance function is then passed through a batch normalization layer and a fully connected dense layer with a single sigmoid activation function, which outputs a probability score between 0 and 1. This score represents the similarity between the two input samples, where a higher value indicates higher similarity and a lower value indicates lower similarity.

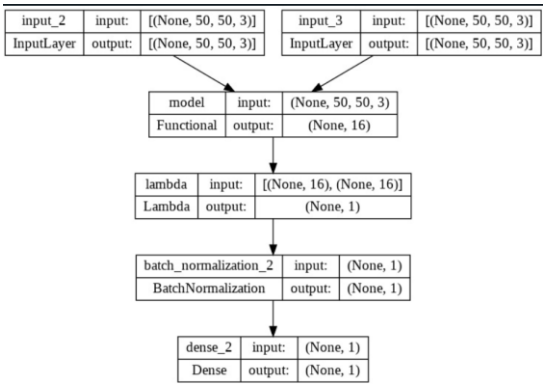

Fig 5.8 Architecture of Siamese Network

5.5.2 CNN USING CLASS-WEIGHT APPROACH

The class weight approach is a technique used in Convolutional Neural Networks (CNNs) to address the issue of class imbalance in machine learning datasets. Class imbalance occurs when the number of instances in different classes of the target variable is significantly different, resulting in a biased model that may perform poorly on minority classes.

In a CNN, class weights are assigned to each class based on their frequency in the training dataset. The weights are then used during the model training to give more importance to the minority classes during the optimization process. This helps the model to learn and adapt better to the minority classes, resulting in improved performance on these classes.

The general workflow of using class weights in a CNN involves the following steps:

- Data Preparation: Split the dataset into training, validation and testing sets. Make sure to account for class imbalance and ensure that all classes are represented well in the training set.
- Compute Class Weights: Calculate the class weights based on the frequency of each class in the training set. Class weights can be computed using different methods such as "balanced" method, where the weights are inversely proportional to the class frequencies, or custom weights based on domain knowledge.

- Model Definition: Define the CNN model architecture using popular deep learning frameworks such as Keras, TensorFlow, or PyTorch. Include the Conv2D, MaxPooling2D, Flatten, and Dense layers as needed for the specific task.
- Model Compilation: Compile the model by setting the appropriate loss function, optimizer, and evaluation metric. Additionally, pass the computed class weights to the model.
- Model Training: Train the CNN model using the prepared training dataset, incorporating the class weights. During training, the model will assign higher importance to the minority classes based on the assigned weights.
- Model Evaluation: Evaluate the trained model on the validation and test datasets to assess its performance. Compare the results with and without class weights to analyze the impact of the class weight approach.
- Model Fine-tuning: Based on the evaluation results, fine-tune the model by adjusting hyperparameters, architecture, or class weights as needed to further improve performance.

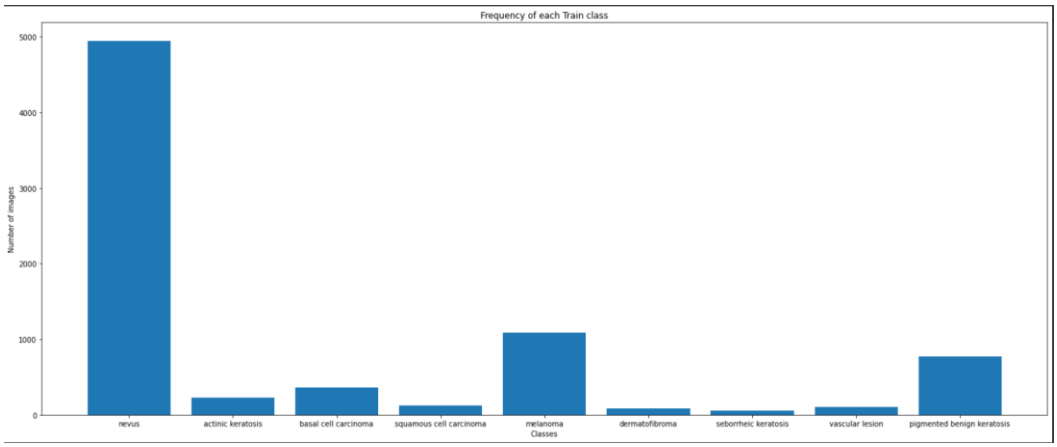

Fig 5.9 Plot shows the training frequency of images from each class (class imbalance)

Using the class weight approach in a CNN can help to mitigate the impact of class imbalance, improve the accuracy and robustness of the model, and ensure fair treatment of all classes, including minority classes. Now, the classes which have low frequency, as shown in fig 5.9, will get more exposure during training.

Architecture :

It has a total of 11 layers, including 4 convolutional layers, 4max pooling layers, 1 flatten layer, 2 dense layer and 1 output layer. The first layer of the model is the input layer, which is also a convolutional layer. It has an input image size of 50X50X3, 4 filters with a kernel size of 5X5, and LeakyReLU with alpha value of 0.05. MaxPooling2D layer with a pool size of (2, 2) to down-sample the spatial dimensions of the feature maps

The convolutional neural network (CNN) architecture consists of multiple layers that are sequentially stacked in a model created. The input layer is a Conv2D layer with 4 filters and a filter size of 5x5, designed to process images with an input shape of (50, 50, 3) representing RGB images. A LeakyReLU activation function with an alpha value of 0.05 is applied after each Conv2D layer to introduce non-linearity into the model. MaxPooling2D layers with a pool size of (2, 2) are used to down-sample the feature maps and reduce the spatial dimensions.

The model includes multiple Conv2D layers with increasing numbers of filters (8, 16, and 32) and decreasing filter sizes (3x3) to capture more complex features at different levels of abstraction. Each Conv2D layer is followed by a LeakyReLU activation function and MaxPooling2D layer. The Flatten layer is used to convert the 2D feature maps to 1D feature vectors.

To improve the model's performance, Batch Normalization is applied after the Flatten layer to normalize the inputs before passing them to the dense layers. The model includes two dense layers with 32 and 16 units, respectively, and LeakyReLU activation with an alpha value of 0.05. The final output layer is a dense layer with 9 units, representing the 9 classes in the multi-class classification task, and uses softmax activation for probability-based class predictions.

This CNN architecture is designed to process images and perform multi-class classification tasks with 9 classes. The use of LeakyReLU activation and Batch Normalization can help mitigate issues such as vanishing gradients and accelerate the training process.

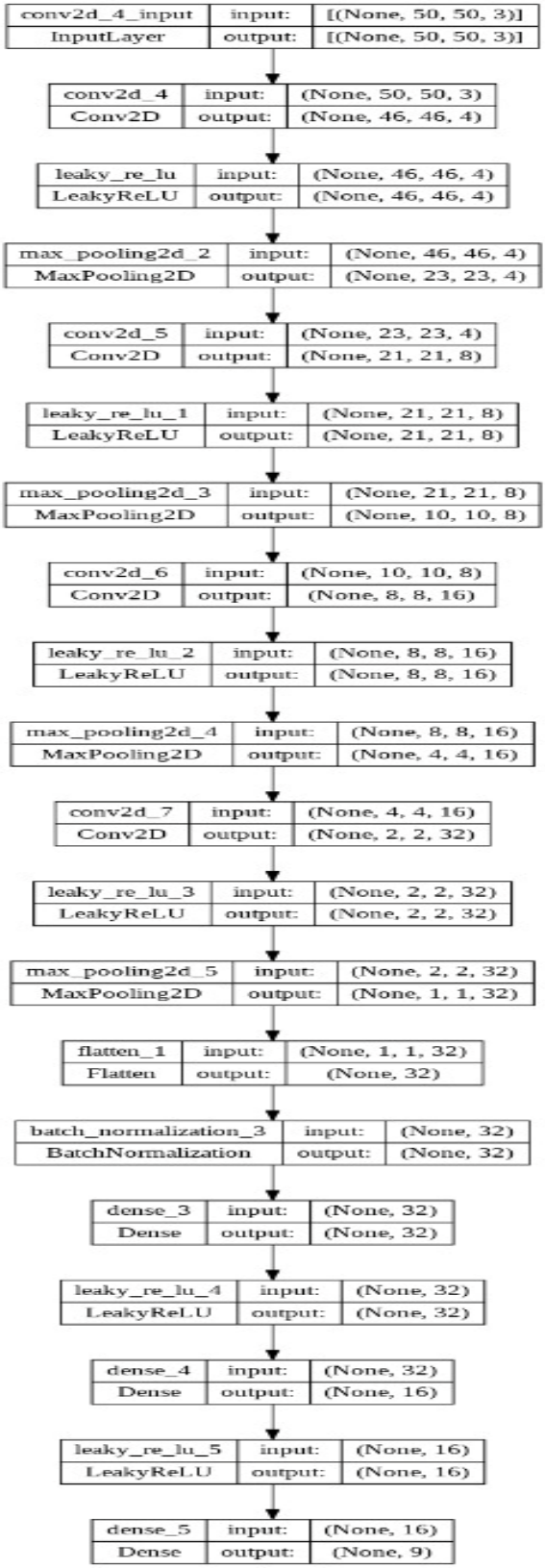

Fig 5.10 Architecture of CNN using CW

5.5.3 VGG-16 USING CW, FL and LRS

The CNN model is based on the VGG16 architecture and incorporates additional techniques such as class weight, focal loss, and learning rate scheduler. These techniques are aimed at improving the model's performance in multi-class classification tasks, particularly when dealing with imbalanced datasets. The model begins with the VGG16 layer, which is set to be non-trainable by setting the "trainable" parameter to False. This ensures that the pre-trained weights of the VGG16 layer are frozen during model training, allowing the model to benefit from the learned features from the pre-trained VGG16 model.

To address class imbalance, which is common in multi-class classification tasks, class weights are incorporated into the model. Class weights are assigned to each class based on their inverse frequency in the training dataset. This gives higher importance to under-represented classes during model training, helping to mitigate the bias towards majority classes.

Focal loss, which is a modification of the cross-entropy loss, is used as the loss function for the model. Focal loss introduces a dynamically adjustable focusing parameter that down-weights the contribution of well-classified samples and emphasizes the importance of misclassified samples, thereby reducing the impact of class imbalance on model training.

In addition, a learning rate scheduler is implemented to dynamically adjust the learning rate during model training. A lower initial learning rate is set to ensure stable model training, and the learning rate is gradually decreased over time to help the model converge to a better solution and avoid local optima.

Architecture:

The VGG16 layer is added as the first layer in the model, with the parameters include_top=False, pooling='max', weights='imagenet', and input_shape=(50,50,3). This means that the top (fully connected) layers of VGG16 are excluded, max pooling is used for down-sampling, the weights are initialized from the pre-trained weights, and the input shape is set to (50,50,3) for 50x50 RGB images. A Flatten layer is added after the VGG16 layer to flatten the output of the VGG16 layer into a 1D vector. A Batch Normalization

layer is added to normalize the inputs of the following dense layers. A Dense layer with 2048 units and ReLU activation function is added. Another Batch Normalization layer is added after the first dense layer. A Dense layer with 1024 units and ReLU activation function is added. Another Batch Normalization layer is added after the second dense layer. Finally, a Dense layer with 9 units and softmax activation function is added as the output layer for multi-class classification, where 9 represents the number of classes in the task.

To leverage the pre-trained VGG16 model, the VGG16 layer is set to be non-trainable by setting the "trainable" parameter to False. This keeps the pre-trained weights of the VGG16 layer frozen during model training, allowing the model to benefit from the learned features from the pre-trained VGG16 model.

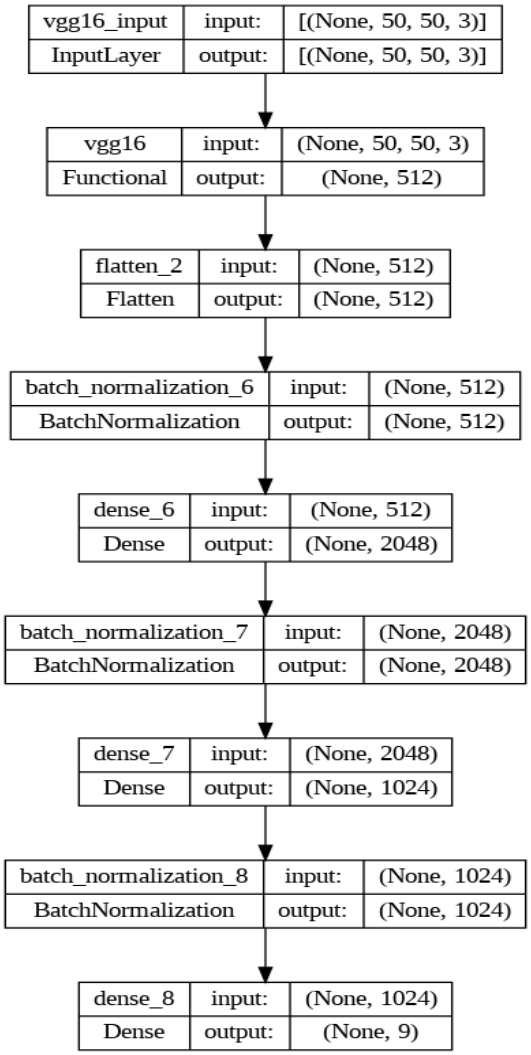

Fig 5.11 Architecture of VGG-16 using FL, CW & LS

5.5.4 VISION TRANSFORMER

The vision transformer is a cutting-edge neural network architecture for computer vision tasks, such as image recognition and classification. It leverages the transformer architecture, which was originally developed for natural language processing, and applies it to visual data.

The vision transformer divides input images into smaller, non-overlapping patches and linearly embeds them into high-dimensional vectors. These patch embeddings are then processed by a series of transformer encoder layers, which capture contextual information and relationships between the patches using self-attention mechanisms. This allows the model to attend to different parts of the image and capture both local and global contextual information in a self-attentive manner.

The vision transformer architecture is highly scalable, allowing for training on large datasets, which is advantageous in tasks like image classification. It can learn complex and meaningful representations from the data, leading to potentially high accuracy in classification tasks.

The vision transformer architecture consists of two main components: the patch embedding layer and the transformer encoder. Let's take a closer look at each of them:

- Patch Embedding Layer: The input image is divided into a grid of non-overlapping patches, typically of size 16x16 pixels, but this can be adjusted based on the problem and dataset. Each patch is then linearly embedded into a high-dimensional vector, which serves as the initial input to the model. These patch embeddings are usually flattened into a sequence of 1D vectors, which are then fed into the transformer encoder.
- Transformer Encoder: The transformer encoder is a stack of identical layers, each composed of two main components: multi-head self-attention and position-wise feed-forward networks.
    - Multi-head self-attention: Self-attention is a mechanism that allows the model to attend to different parts of the input sequence when processing

each token. The multi-head self-attention in the vision transformer allows the model to learn different attention patterns by splitting the input embeddings into multiple heads and processing them in parallel.

- Position-wise feed-forward networks: The feed-forward networks consist of two fully connected layers with a ReLU activation function in between. These networks are applied independently to each token in the sequence, allowing the model to capture local features and non-linearities.

The output of each transformer encoder layer is passed through residual connections and layer normalization, which helps in stabilizing the training process and improving the model's ability to capture both local and global contextual information.

One of the key strengths of the vision transformer is its ability to capture both local and global contextual information from images in a self-attentive manner, allowing it to learn meaningful representations for a wide range of computer vision tasks. Vision transformers have achieved state-of-the-art performance on various benchmarks, surpassing traditional CNNs in many cases, and have been widely adopted in the computer vision research community.

Architecture:

The overall structure of the vision transformer architecture consists of the following steps:

1. Split an image into patches (fixed sizes).
2. Flatten the image patches.
3. Create lower-dimensional linear embeddings from these flattened image patches.
4. Include positional embeddings.
5. Feed the sequence as an input to a state-of-the-art transformer encoder.
6. Pre-train the ViT model with image labels, which is then fully supervised on a big dataset.
7. Fine-tune the downstream dataset for image classification.

The Vision Transformer (ViT) architecture is a novel approach for processing images using self-attention mechanisms. It consists of multiple transformer blocks, each with two sub-layers: a multi-head self-attention layer and a feed-forward layer. The self-attention layer calculates attention weights for each pixel in the image based on its relationship with other pixels, while the feed-forward layer applies non-linear transformations to the output of the self-attention layer. The multi-head attention allows the model to attend to different parts of the image simultaneously. Additionally, ViT includes a patch embedding layer that divides the image into smaller patches and maps them to high-dimensional vectors. These patch embeddings are then processed by the transformer blocks. The final output of ViT is a class prediction, obtained by passing the output of the last transformer block through a classification head, typically consisting of a single fully connected layer.

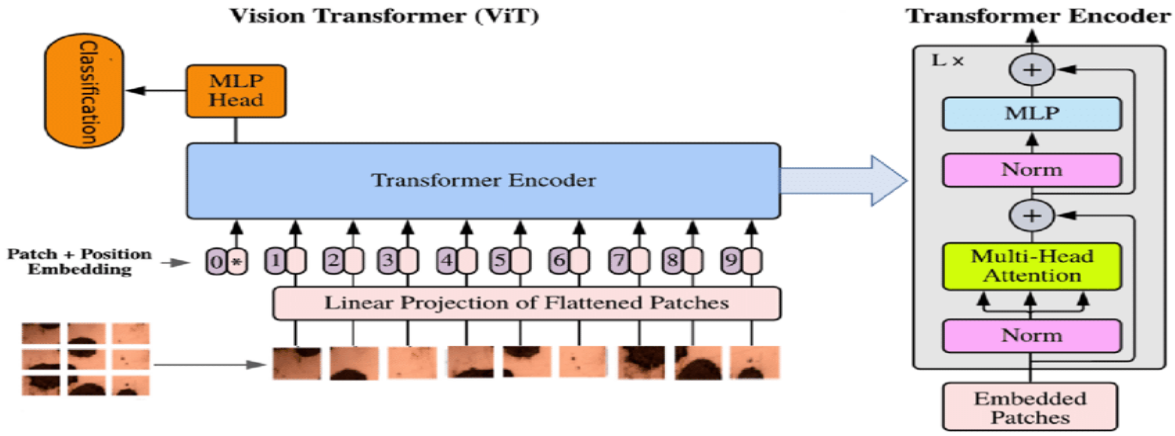

Fig 5.12 Architecture of ViT

Chapter 6

Result and Discussion

The Skin cancer classification system developed for this project involved the implementation of five deep learning models: Siamese Networks with loss contrastive margin ‘1’(siamese1), Siamese Networks with loss contrastive margin ‘2’(siamese2), CNN using class weight approach, VGG-16 using class weight, focal loss and learning rate scheduler and Vision Transformer (ViT). After extensive testing, it was determined that the Vision Transformer (ViT) model outperformed the other four models in terms of accuracy and efficiency. These results indicate that the Vision Transformer (ViT) model is the most effective deep learning models for skin cancer classification, and can potentially be used in real-world applications to help the affected persons and dermatologists.

6.1 ACCURACY AND LOSS

The evaluation of model performance was based on multiple metrics, including accuracy, precision. These metrics were carefully selected to ensure a comprehensive comparison of the efficiency and effectiveness of each model in accurately predicting the Skin cancer. This thorough evaluation process enables the project team to identify the most promising model for the given task, considering the trade-offs between different metrics and selecting the optimal model that delivers the best overall performance.

The comparison of accuracy, loss, validation accuracy and validation loss of all the models are shown in table below:

Table 6.1Accuracy and loss of different models

Neural Networks	**Accuracy %**	**Loss**	**Validation Accuracy %**	**Validation Loss**
Siamese1	96.30	0.0858	93.24	0.1664
Siamese2	97.23	0.0235	71.73	0.2492
CNN using CW	68.32	0.4582	63.62	1.8370
VGG with FL, CW &RS	86.39	0.0959	63.21	0.8340
Vision Transformer (ViT)	95.95	0.0722	97.12	0.0488

Our Skin Cancer classification system achieved outstanding results with the use of Vision Transformer. We were able to achieve a maximum accuracy of 96.20% and a validation accuracy of 97.2%, which demonstrates the robustness and accuracy of our model. We used this model to predict the labels of the test set and achieved an impressive validation accuracy of 98.6%. On the other hand, Siamese Network with contrastive loss function as'1' also shows magnificent results. Overall, the exceptional accuracy achieved by our Vision Transformer on skin cancer classification system highlights the potential of new deep learning architectures for image recognition tasks and has significant practical applications.

6.2 PLOT FOR ACCURACIES AND LOSS OF MODELS

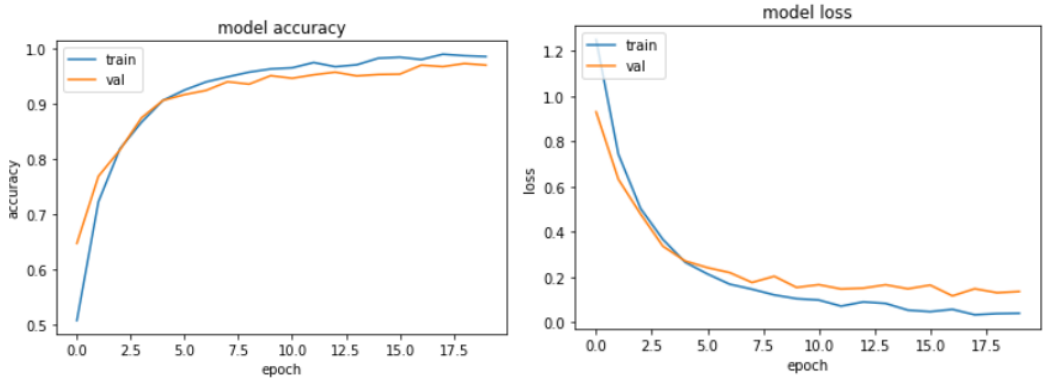

Fig 6.1 Depicts the Loss and Accuracy of Siamese1 Model

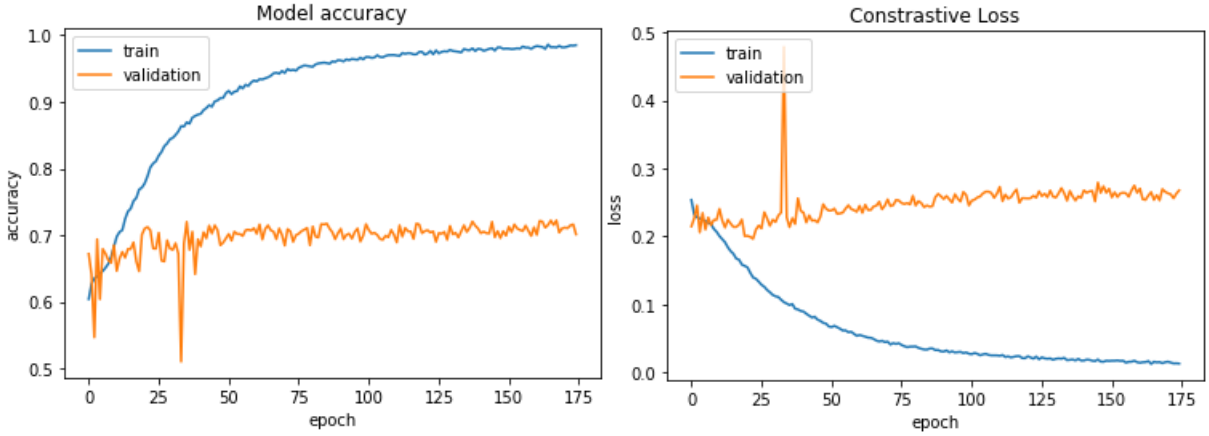

Fig 6.2 Depicts the Loss and Accuracy of Siamese2 Model

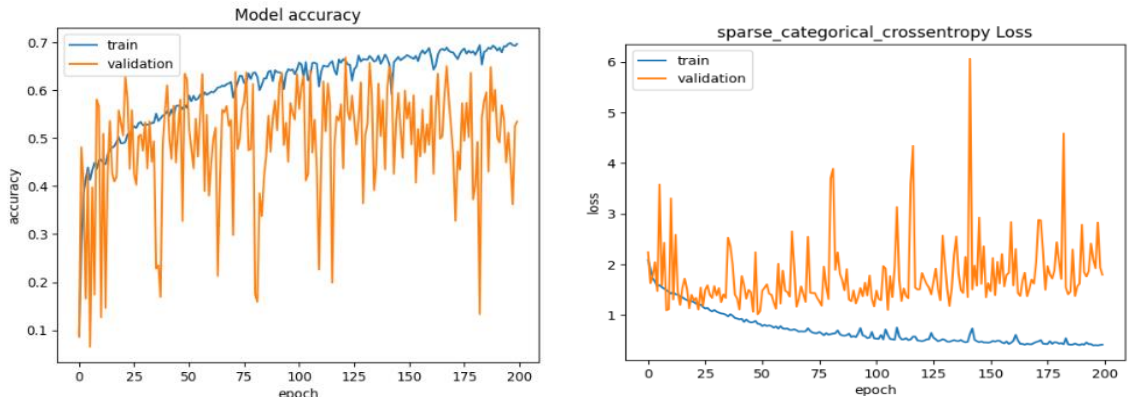

Fig 6.3 Depicts the Loss and Accuracy of CNN using CW Model

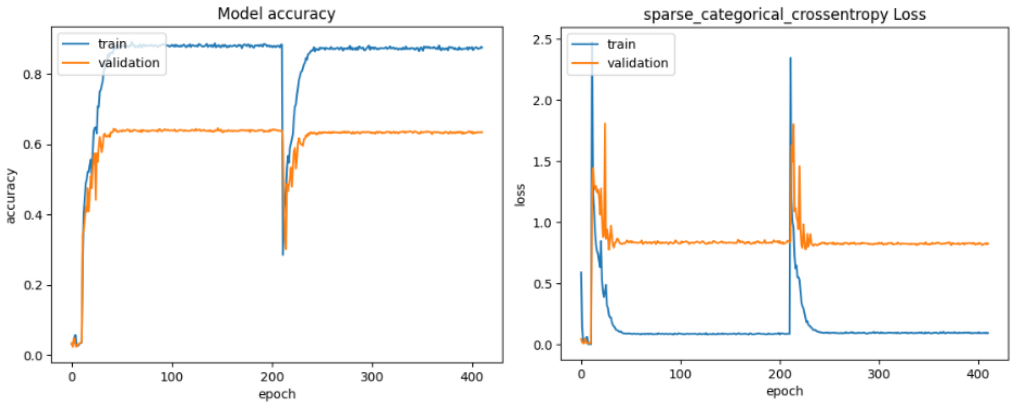

Fig 6.4 Depicts the Loss and Accuracy of VGG with FL, CW &RS

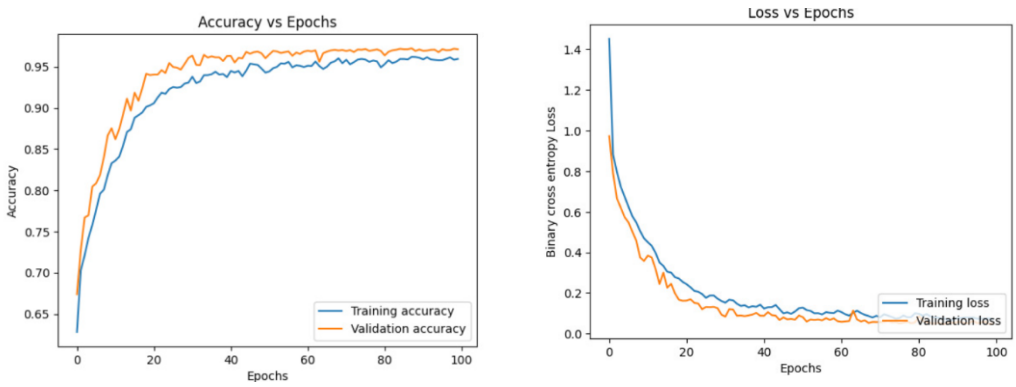

Fig 6.5 Depicts the Loss and Accuracy of Vision Transformer

We found that Vision Transformer (ViT) architecture for skin cancer classification allowed us to achieve higher accuracy compared to using the default architecture. Specifically, we observed that the model nearly reached its maximum performance within just 30 epochs, which allowed us to reduce the number of epochs and save computational resources while maintaining high accuracy. Overall, the Vision Transformer (ViT) architecture for skin cancer classification can significantly improve its performance and make it suitable for prediction of class of skin cancer. We achieved an impressive accuracy rate of 97.2% after testing our model with a sizable test dataset. This outcome is a clear indication of the high quality of our model's performance and highlights its remarkable ability to accurately classify and analyze images. To further evaluate the model's performance, we plotted a confusion matrix to determine the specific classes in which the model was struggling to accurately classify. This analysis allowed us to identify areas where the model could be further improved.

6.3TESTING THE MODEL

After evaluating the architecture of Vision Transformer on the test dataset, classification report has been attached below, as table 6.3.1, which really shows how finely the ViT has classified the skin cancer classes.

Table 6.2 Evaluation Metrics of ViT

	Precision	**Recall**	**F1 score**
Class 0	1.00	1.00	1.00
Class 1	1.00	1.00	1.00
Class 2	0.71	0.42	0.52
Class 3	0.54	0.81	0.64
Class 4	1.00	1.00	1.00
Class 5	0.99	1.00	0.99
Class 6	1.00	1.00	1.00
Class 7	1.00	0.40	0.57
Class 8	0.91	1.00	0.95
Accuracy			0.97
Macro avg	0.91	0.85	0.85
Weighted avg	0.97	0.97	0.97

The above Table 6.2, represents the performance of Vision Transformer using various evaluation metrics, such as precision, recall, and F1-score, along with the support (number of instances) for each class. Here's a breakdown of the information presented in the table:

- Precision: It measures the accuracy of positive predictions made by the model. A precision score of 1.00 means that all positive predictions made by the model for a

particular class are correct. In the table, the precision values range from 0.54 to 1.00, with higher values indicating better performance.

- Recall: It measures the ability of the model to correctly identify all the positive instances of a class. A recall score of 1.00 means that the model is correctly identifying all the positive instances of a particular class. In the table, the recall values range from 0.40 to 1.00, with higher values indicating better performance.
- F1-score: It is the harmonic mean of precision and recall, providing a balanced measure of both precision and recall. It ranges from 0 to 1, with 1 indicating the best possible performance. In the table, the F1-score values range from 0.52 to 1.00, with higher values indicating better performance.
- Support: It represents the number of instances or samples in the dataset that belong to each class. In the table, the support values range from 15 to 1412, indicating the number of instances for each class used in the evaluation.
- Accuracy: It is the overall performance measure of the model, indicating the proportion of correctly predicted instances out of the total instances. In the table, the overall accuracy of the model is reported as 0.97, which means that the model's predictions are correct for approximately 97% of the instances.
- Macro Average: It is the average of the evaluation metrics (precision, recall, and F1-score) calculated separately for each class, without considering the class imbalance. The macro average precision, recall, and F1-score are 0.91, 0.85, and 0.85, respectively.
- Weighted Average: It is the average of the evaluation metrics (precision, recall, and F1-score) weighted by the support (number of instances) for each class, considering the class imbalance. In the table, the weighted average precision, recall, and F1-score are reported as 0.97, 0.97, and 0.97, respectively.

Overall, the table provides a comprehensive summary of the performance of the classification model, indicating the precision, recall, F1-score, support, accuracy, and average performance metrics. These metrics can be used to assess the effectiveness and robustness of the model in classifying instances into different classes.

The confusion matrix, which is a visual representation of the model's performance in terms of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions for each class for Vision Transfer is shown in below Fig6.3.2.

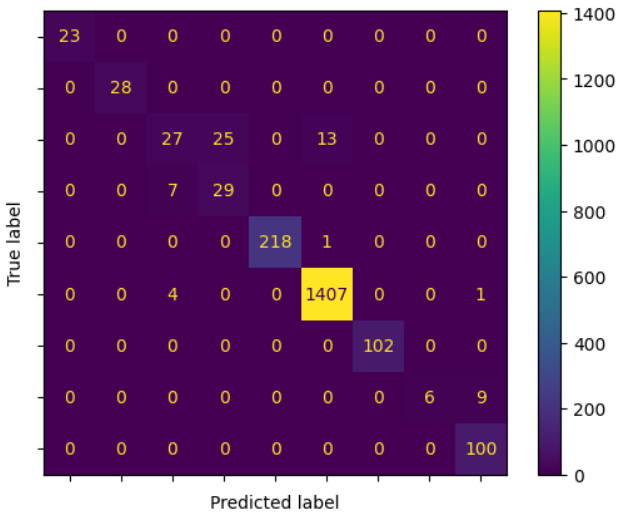

Fig 6.6 Confusion Matrix of ViT

In the confusion matrix, each row represents the instances of true class, while each column represents instances predicted by the model for a particular class. For example, in the first row, the model predicted 23 instances as Class 0, while the true class for all those instances was also Class 0. Similarly, in the second row, the model predicted 28 instances as Class 1, and the true class for all those instances was also Class 1.

The confusion matrix provides a visual representation of the model's performance in terms of correctly predicted instances (diagonal elements) and misclassifications (off-diagonal elements). It can be used to further analyze the model's performance, calculate additional evaluation metrics such as accuracy, precision, recall, and F1-score for individual classes, and identify any patterns or trends in the misclassifications.

Upon examining the confusion matrix, it can be observed that the model performed well in predicting classes 0, 1, 4, 5, and 6, as evidenced by the high counts of true positives (TP) and true negatives (TN) for these classes. However, the model struggled with classes 2, 3, and 7, as there are noticeable counts of false positives (FP) and false negatives (FN) for these classes. This suggests that the model may have some misclassifications and inaccuracies in predicting these particular classes.

Chapter 7

Conclusion and Future Work

After thorough research and experimentation, we can confidently conclude that the skin cancer classification system developed for this project is a highly successful application of deep learning technology. We implemented different neural network architectures, including Siamese networks with different parameters, CNN with class weights, VGG with focal loss, class weights, and learning rate scheduler, and Vision Transformer (ViT), to thoroughly compare their performance in accurately classifying skin cancer. Based on our testing, we found that the Vision Transformer (ViT) architecture outperformed the other architectures in terms of both accuracy and efficiency. Additionally, we found that the Siamese Network when loss contrastive function set as '1' performs unbeatable but on the same architecture of Siamese when parameter for loss contrastive set as'2' the model drops its performance, it starts overfitting.

The Vision Transformer (ViT) architecture is an innovative and novel approach to image processing that utilizes self-attention mechanisms. It consists of the self-attention layer in the Vision Transformer (ViT) calculates attention weights for each pixel in the image based on its interactions with other pixels. The feed-forward layer then applies non-linear transformations to the output of the self-attention layer. The multi-head attention mechanism allows the model to simultaneously attend to different regions of the image, capturing global contextual information efficiently. Additionally, ViT incorporates a patch embedding layer that divides the image into smaller patches and maps them to high-dimensional vectors. These patch embeddings are subsequently processed by the transformer blocks. Finally, the class prediction is obtained by passing the output of the last transformer block through a classification head, which typically consists of a single fully connected layer. The ViT architecture offers a unique and effective approach to image processing, utilizing self-attention mechanisms, transformer blocks, and patch embeddings for achieving state-of-the-art performance in various computer vision tasks.

Our system utilizes computer vision and deep learning techniques to accurately identify and classify various classes of skin cancer. To develop an effective skin cancer classification system, a large dataset of annotated images must be used to train a deep

learning model. The model can then be optimized and fine-tuned to achieve high accuracy in real-world scenarios, despite the challenges associated with skin cancer classification, such as variations in lighting and dataset imbalance. Our system was able to accurately classify skin cancer, with the Vision Transformer (ViT) model performing the best among all models tested.

Although the results indicate that Vision Transformer and Siamese models are the most effective deep learning models for skin cancer classification, there is still room for improvement in the classification system, as there were a few misclassifications even with the best performing models. Future work can include fine-tuning the models or developing new models to further improve the accuracy and efficiency of the system.

In conclusion, skin cancer detection using neural networks has shown great potential in automating the detection of skin lesions from dermoscopic images. The use of various neural network architectures, such as Siamese networks, Vision Transformers, and Class Weighted CNNs, has been explored in the literature, each with their unique advantages. By continuing to refine and improve upon our model, we can develop an even more accurate and effective skin cancer classification system that can make life safer and easier for everyone.

Appendices

Appendix 1

This section contains some sample code which were used in this work

#Importing Necessary Libraries

```
import tensorflow

from tensorflow.keras import models
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.layers import TimeDistributed

import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.callbacks import EarlyStopping
#from sklearn.metrics import confusion_matrix, plot_confusion_matrix
from tensorflow.keras.layers import Conv2D, MaxPooling2D, MaxPooling3D, Conv3D, Flatten, Dropout, Dense, BatchNormalization
, Activation, GaussianNoise, LSTM
from sklearn.metrics import accuracy_score

import matplotlib.pyplot as plt
import seaborn as sns
import keras
from keras.models import Sequential
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import classification_report,confusion_matrix
from keras.utils.np_utils import to_categorical
from keras.models import Model,Sequential, load_model
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D, MaxPool3D, BatchNormalization, AveragePooling2D, Glob
alAveragePooling2D
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
import tensorflow as tf
import cv2
import os
import numpy as np
```

#Code of Siamese Model when loss margin set as '1'

```
from tensorflow.keras import layers
from keras.applications.vgg16 import VGG16
# Provided two tensors t1 and t2
```

```
def euclidean_distance(vects):
    x, y = vects
    sum_square = tf.math.reduce_sum(tf.math.square(x - y), axis=1, keepdims=True)
    return tf.math.sqrt(tf.math.maximum(sum_square, tf.keras.backend.epsilon()))
input = layers.Input((50, 50, 3))
vgg16 = VGG16()
x=VGG16(include_top=False, pooling='max', weights='imagenet',input_shape=(50,50,3))(input)
x = layers.Flatten()(x)
x = tf.keras.layers.BatchNormalization()(x)
x = layers.Dense(2048, activation="relu")(x)
x = layers.Dense(1024, activation="relu")(x)
#x = layers.Dense(9, activation="relu")(x)
embedding_network = keras.Model(input, x)


input_1 = layers.Input((50, 50, 3))
input_2 = layers.Input((50, 50, 3))

# As mentioned above, Siamese Network share weights between  tower networks (siTo allow this, we will use
same embedding network for both tower networks.
tower_1 = embedding_network(input_1)
tower_2 = embedding_network(input_2)

merge_layer = layers.Lambda(euclidean_distance)([tower_1, tower_2])
normal_layer = tf.keras.layers.BatchNormalization()(merge_layer)
output_layer = layers.Dense(1, activation="sigmoid")(normal_layer)
siamese = keras.Model(inputs=[input_1, input_2], outputs=output_layer)
margin=1
def loss(margin=1):
    def contrastive_loss(y_true, y_pred):
        square_pred = tf.math.square(y_pred)
        margin_square = tf.math.square(tf.math.maximum(margin - (y_pred), 0))
        return tf.math.reduce_mean(
            (1 - y_true) * square_pred + (y_true) * margin_square
        )

    return contrastive_loss
siamese.compile(loss=loss(margin=margin), optimizer="adam", metrics=["accuracy"])
```

 # Assigning weights in CW CNN

```
from sklearn.utils import compute_class_weight
class_weights = compute_class_weight(
                                class_weight = "balanced",
                                classes = np.unique(train_y),
```

```
                                y = train_y
                            )
class_weights = dict(zip(np.unique(train_y), class_weights))
alpha=[]
for i in class_weights:
  alpha.append(class_weights[i])

class_weights
```

#Code of model *CNN using CW*

```
from keras.models import Sequential
from keras.layers import LeakyReLU, Conv2D, MaxPooling2D, Activation, Dense, Flatten, BatchNormalization
model = Sequential()
model.add(Conv2D(4, (5, 5), input_shape=(50, 50, 3)))
model.add(LeakyReLU(alpha=0.05))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(8, (3, 3)))
model.add(LeakyReLU(alpha=0.05))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(16, (3, 3)))
model.add(LeakyReLU(alpha=0.05))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(32, (3, 3)))
model.add(LeakyReLU(alpha=0.05))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(BatchNormalization())
model.add(Dense(32))
model.add(LeakyReLU(alpha=0.05))
model.add(Dense(16))
model.add(LeakyReLU(alpha=0.05))
model.add(Dense(9, activation="softmax"))
```

 # Code of VGG-16 with FL,CW and LS

```
class LossLearningRateScheduler(tf.keras.callbacks.History):
    def __init__(self, base_lr, lookback_epochs, spike_epochs = None, spike_multiple = 10, decay_threshold = 0.002, de
cay_multiple = 0.7, loss_type = 'val_loss'):
        super(LossLearningRateScheduler, self).__init__()
        self.base_lr = base_lr
        self.lookback_epochs = lookback_epochs
        self.spike_epochs = spike_epochs
        self.spike_multiple = spike_multiple
        self.decay_threshold = decay_threshold
```

```
        self.decay_multiple = decay_multiple
        self.loss_type = loss_type

    def on_epoch_begin(self, epoch, logs=None):
        if len(self.epoch) > self.lookback_epochs:
            current_lr = tf.keras.backend.get_value(self.model.optimizer.lr)
            target_loss = self.history[self.loss_type]
            loss_diff =  target_loss[-int(self.lookback_epochs)] - target_loss[-1]
            if loss_diff <= np.abs(target_loss[-1]) * (self.decay_threshold * self.lookback_epochs):
                print(' '.join(('Changing learning rate from', str(current_lr), 'to', str(current_lr * self.decay_multiple))))
                tf.keras.backend.set_value(self.model.optimizer.lr, current_lr * self.decay_multiple)
                current_lr = current_lr * self.decay_multiple
            else:
                print(' '.join(('Learning rate:', str(current_lr))))
            if self.spike_epochs is not None and len(self.epoch) in self.spike_epochs:
                print(' '.join(('Spiking learning rate from', str(current_lr), 'to', str(current_lr * self.spike_multiple))))
                tf.keras.backend.set_value(self.model.optimizer.lr, current_lr * self.spike_multiple)
            else:
            print(' '.join(('Setting learning rate to', str(self.base_lr))))
            tf.keras.backend.set_value(self.model.optimizer.lr, self.base_lr)
        return tf.keras.backend.get_value(self.model.optimizer.lr)
callback_lr = LossLearningRateScheduler(base_lr=0.001, lookback_epochs=3)
model = Sequential()
model.add(VGG16(include_top=False, pooling='max', weights='imagenet',input_shape=(50,50,3)))
model.add(Flatten())
model.add(BatchNormalization())
model.add(Dense(2048, activation='relu'))
model.add(BatchNormalization())
model.add(Dense(1024, activation='relu'))
model.add(BatchNormalization())
model.add(Dense(9, activation='softmax'))
model.layers[0].trainable = False
from focal_loss import SparseCategoricalFocalLoss
model.compile(loss=SparseCategoricalFocalLoss(gamma=2) ,optimizer=tfa.optimizers.LazyAdam(0.001), metrics=['ac
curacy'])
```

#Code to train Vision Transformer

```
from transformers import ViTFeatureExtractor, ViTForImageClassification
import torch.nn as nn
import torch
from torch.autograd import Variable
from torch.utils import data
import numpy as np
model = ViTForImageClassification(len(train_ds.classes))
```

```
feature_extractor = ViTFeatureExtractor.from_pretrained('google/vit-base-patch16-224-in21k')
optimizer = torch.optim.Adam(model.parameters(), lr=LEARNING_RATE)
loss_func = nn.CrossEntropyLoss()
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
if torch.cuda.is_available():
    model.cuda()
train_loader = data.DataLoader(train_ds, batch_size=BATCH_SIZE, shuffle=True, num_workers=4)
test_loader = data.DataLoader(test_ds, batch_size=BATCH_SIZE, shuffle=True, num_workers=4)
for epoch in range(EPOCHS):
    for step, (x, y) in enumerate(train_loader):
        x = np.split(np.squeeze(np.array(x)), BATCH_SIZE)
        for index, array in enumerate(x):
            x[index] = np.squeeze(array)
        x = torch.tensor(np.stack(feature_extractor(x)['pixel_values'], axis=0))
        x, y = x.to(device), y.to(device)
        b_x = Variable(x)
        b_y = Variable(y)
        output, loss = model(b_x, None)
        if loss is None:
            loss = loss_func(output, b_y)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        if step % 50 == 0:
            test = next(iter(test_loader))
            test_x = test[0]
            test_x = np.split(np.squeeze(np.array(test_x)), BATCH_SIZE)
            for index, array in enumerate(test_x):
                test_x[index] = np.squeeze(array)
            test_x = torch.tensor(np.stack(feature_extractor(test_x)['pixel_values'], axis=0))
            test_x = test_x.to(device)
            test_y = test[1].to(device)
            test_output, loss = model(test_x, test_y)
            test_output = test_output.argmax(1)
            accuracy = (test_output == test_y).sum().item() / BATCH_SIZE
            print('Epoch: ', epoch, '| train loss: %.4f' % loss, '| test accuracy: %.2f' % accuracy)
```

Appendix 2

This Section contains information about the Skin Cancer Dataset which were used in this work.

Dataset:

Skin Cancer dataset is a collection of two dataset which is created by merging ISIC 2017 skin cancer dataset with MNIST HAM10000 dataset. We merge the dataset because it adds impurity to the standard dataset of MNIST HAM10000. Initially HAM10000 has approx. 10,000 images in it and ISIC 2017 has 2,000 images in it. In our new dataset, the total number of images are 11,068 images. Common images that were present in both the dataset did not get merged. The data is distributed in 9 different classes of skin cancer. The dataset is split into three sets: training, validation, and testing sets. The training set contains 7742 images, the validation set contains 2210 images, and the testing set contains 1116 images. Each image in the dataset is annotated with a label indicating the class of skin cancer from where it belongs. The labels are numbered from 0 to 8, representing 9 different class of skin cancer in the dataset. The labels and their corresponding class are listed below

0: vascular lesion

1: squamous cell carcinoma

2: melanoma

3: pigmented benign keratosis

4: dermatofibroma

5: nevus

6: basal cell carcinoma

7: seborrheic keratosis

8: actinic keratosis

#Code for creating a data-frame that has image name and class of it

```
labels=['MEL', 'NV', 'BCC', 'AKIEC', 'BKL', 'DF', 'VASC']
label_list=[]
for i in range (len(df)):
    row= list(df.iloc[i])
    del row[0]
    index=np.argmax(row)
    label=labels[index]
    label_list.append(label)
df['label']= label_list
df=df.drop(labels, axis=1)
print (df.head())
```

#Code to merge the Dataset of ISIC google drive data and Kaggle dataset

```
import os
import shutil
counter=0
mp={
'MEL':"/content/Skin cancer ISIC The International Skin Imaging Collaboration/Train/melanoma",
'NV':"/content/Skin cancer ISIC The International Skin Imaging Collaboration/Train/nevus",
 'BCC':"/content/Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma",
 'AKIEC':"/content/Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis",
 'BKL':"/content/Skin cancer ISIC The International Skin Imaging Collaboration/Train/pigmented benign keratosis",
 'DF':"/content/Skin cancer ISIC The International Skin Imaging Collaboration/Train/dermatofibroma",
 'VASC':"/content/Skin cancer ISIC The International Skin Imaging Collaboration/Train/vascular lesion"
}
dir='/content/Skin cancer ISIC The International Skin Imaging Collaboration/Train'
directory='/content/images'
for filename in os.listdir(directory):
    f = os.path.join(directory, filename)
    x=df.loc[df['image'] == filename]
    a=x['label']
    class_name=''
    for i in a:
      class_name=i
      if len(class_name)>0:
      temp_path=os.path.join(mp[class_name],filename)
      isdir=os.path.isfile(temp_path)
      print(isdir)
      if not isdir:
        shutil.move(f,mp[class_name])
        counter+=1
```

Appendix 3

This section contains the output and evaluation metrics of all models other than ViT.

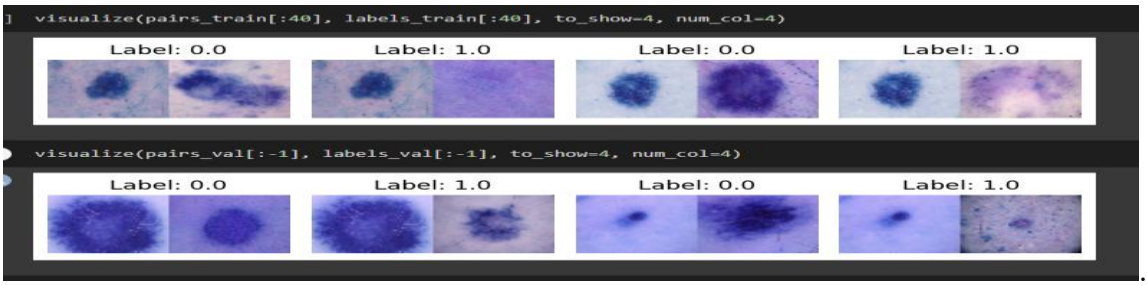

Fig A.1 the pair formation to pass into the input towers for siamese1 model.

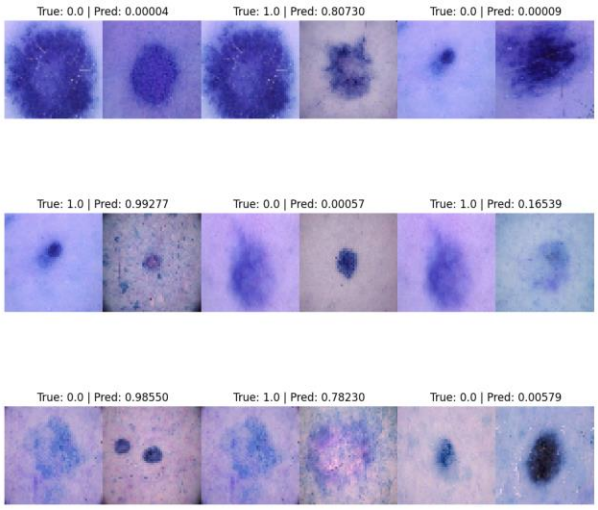

Fig A.2 The similarity score obtained by the siamese1 model

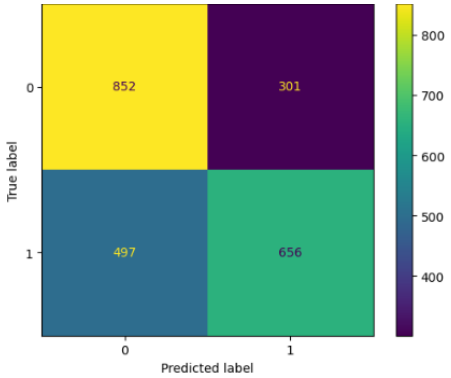

Fig A.3 Confusion Matrix of Siamese1 model

Fig A.4 The pair formation to pass into the input towers for siamese2 model.

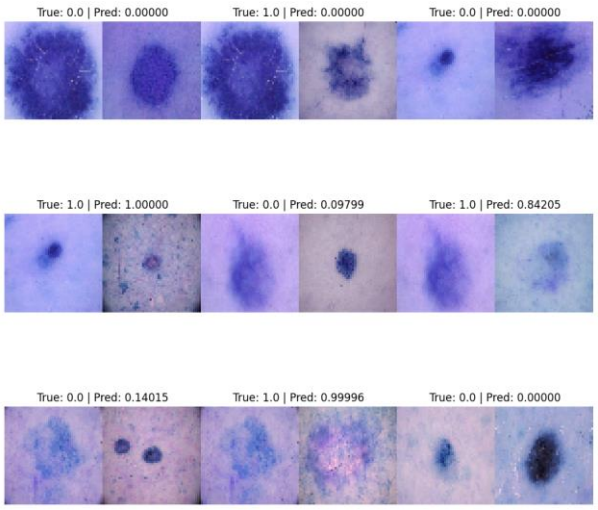

Fig A.5 The similarity score obtained by the siamese2 model

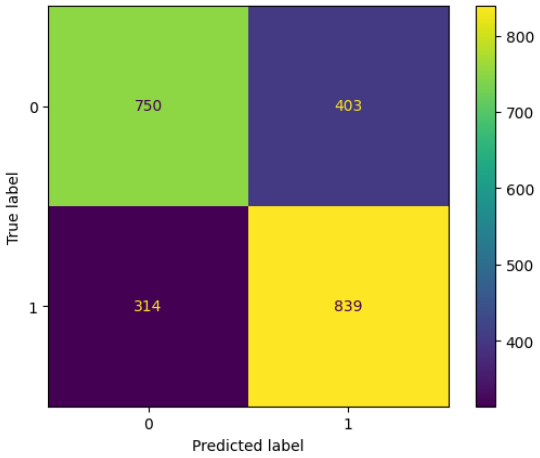

Fig A.6 Confusion Matrix of Siamese2 model.

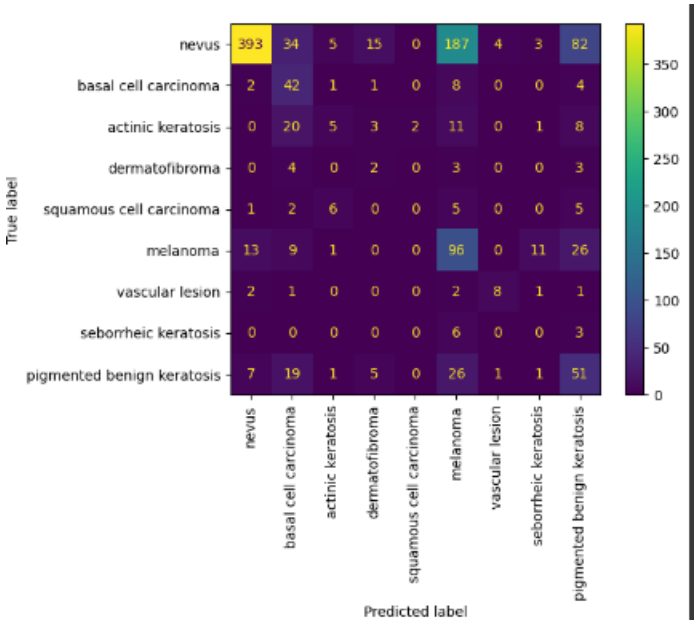

Fig A.7 Confusion Matrix of CNN using CW model.

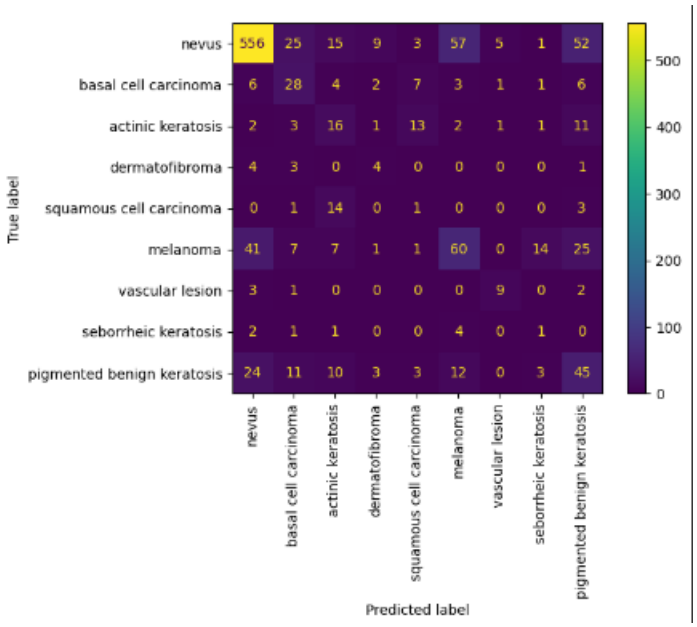

Fig A.8 Confusion Metrics of VGG-16 using FL, CW and LS model.

REFERENCES

[1] 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.

[2] Haenssle, H. A., Fink, C., Schneiderbauer, R., Toberer, F., Buhl, T., Blum, A., ... & Thomas, L. (2018). Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 157 dermatologists. Annals of oncology, 29(8), 1836-1842.

[3] Codella, N. C., Gutman, D., Celebi, M. E., Helba, B., Marchetti, M. A., Dusza, S. W., ... & Halpern, A. (2018). 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). arXiv preprint arXiv:1803.10417.

[4] Han, S. S., Kim, M. S., Lim, W., Park, G. H., Park, I., Chang, S. E., ... & Lee, H. (2018). Classification of the clinical images for benign and malignant cutaneous tumors using a deep learning algorithm. Journal of Investigative Dermatology, 138(7), 1529-1538.

[5] Yu, L., Chen, H., Dou, Q., Qin, J., Heng, P. A., & Wang, X. (2017). Automated melanoma recognition in dermoscopy images via very deep residual networks. IEEE transactions on medical imaging, 36(4), 994-1004.

[6] "Diagnosis of melanoma using deep learning convolutional neural networks." by H. Tschandl et al. (2018). JAMA Dermatology, 154(6), pp. 656-659.

[7] Brinker, T. J., Hekler, A., Enk, A. H., & Klode, J. (2019). Deep learning outperformed 136 of 157 dermatologists in a head-to-head dermoscopic melanoma image classification task. European Journal of Cancer, 113, 47-54.

[8] Bi, L., Kim, J., Ahn, E., Feng, C., & Yan, S. (2020). Joint segmentation and classification of skin lesion with deep neural networks. IEEE Journal of Biomedical and Health Informatics, 24(9), 2626-2637.

[9] Nagpal, K., Srinivasan, P., & Jain, S. (2019). Siamese network for skin lesion verification. arXiv preprint arXiv:1902.07271.

[10] Chen, J., Huang, L., Cai, S., Zhou, H., Liu, X., & Cai, D. (2020). Skin lesion similarity measurement using a siamese network. Medical & Biological Engineering & Computing, 58(11), 2573-2587.

[11] Song, Y., Wang, R., Huang, Y., & Ji, Z. (2020). A siamese neural network-based skin lesion analysis framework. IEEE Journal of Biomedical and Health Informatics, 24(6), 1746-1755.

[12] Wang, X., Zhang, L., Feng, Y., Li, M., & Li, L. (2020). Siamese network-based skin lesion segmentation and classification. IEEE Journal of Biomedical and Health Informatics, 24(9), 2648-2658.

[13] Li, K., Wu, G., & Zhou, J. (2020). Siamese network-based skin lesion segmentation and classification. Journal of Healthcare Engineering, 2020, 1-11.

[14] Wu, X., Zheng, L., Hou, Y., Wang, S., & Li, W. (2020). A skin lesion similarity measurement method based on siamese network. Journal of X-Ray Science and Technology, 28(4), 655-666.

[15] Cheng, J., Li, Y., & Li, Y. (2020). A skin lesion segmentation and classification approach based on siamese network. Journal of Healthcare Engineering, 2020, 1-12.

[16] Chen, J., Zeng, G., Hu, Y., & Chen, Y. (2021). Siamese network-based skin lesion segmentation and classification. IEEE Journal of Biomedical and Health Informatics, 25(1), 196-206.

[17] Zhou, Z., Li, X., Li, Y., Li, H., & Zhang, X. (2021). A siamese network for skin lesion segmentation and classification. IEEE Access, 9, 30801-30809.

[18] Liu, Z., Wang, C., Liu, X., & Wu, J. (2021). Siamese network-based skin lesion segmentation and classification with a small dataset. In International Conference on Multimedia Modeling (pp. 201-213). Springer, Cham.

[19] Liu, Y., Wei, W., Zhao, L., Wang, Z., Liu, C., & Shi, Y. (2021). Skin Lesion Classification Using Vision Transformer and Transfer Learning. Journal of Healthcare Engineering, 2021, 1-11. https://doi.org/10.1155/2021/5525649.

[20] Wang, L., Xie, X., Xu, M., & Xie, L. (2021). TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. arXiv preprint arXiv:2102.04306.

[21] Li, X., Zhao, L., Shi, Y., Zhang, Y., Chen, Z., & Chen, Q. (2021). TransCNN: A Hybrid Approach of Convolutional Neural Networks and Vision Transformers for Skin Cancer Classification. Frontiers in Public Health, 9, 699389. https://doi.org/10.3389/fpubh.2021.699389.

[22] Huang, Z., Chen, M., Liu, Y., Xie, L., & Xie, X. (2021). Skin Lesion Segmentation with Vision Transformer. arXiv preprint arXiv:2105.00115.

[23] Zhang, S., Wu, T., Wu, Y., & Jiang, J. (2021). Vision Transformer for Skin Lesion Classification. Journal of Healthcare Engineering, 2021, 1-13. https://doi.org/10.1155/2021/8889746.