Skin lesion analysis for Melanoma detection

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

SUNNY KUMAR (20BCE0262)

SNEHIL SINHA (20BCE2005)

AMRIT YASH SRIVASTAVA (20BCE2259)

Under the guidance of

Prof. / Dr.

DR. ATHIRA K.

School of Computer Science and Engineering,

VIT, Vellore.

May, 2024

DECLARATION

I hereby declare that the thesis entitled "Skin lesion analysis for Melanoma detection" submitted by me, for the award of the degree of *Bachelor of Technology in Programme* to VIT is a record of bonafide work carried out by me under the supervision of Prof. / Dr. *ATHIRA K*.

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 : Vellore

Date : 9/5/24

Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled "Skin lesion analysis for Melanoma detection" submitted by **Snehil Sinha & 20BCE2005**, **School of Computer Science and Engineering**, VIT, for the award of the degree of *Bachelor of Technology in Programme*, is a record of bonafide work carried out by him / her under my supervision during the period, 01. 12. 2023 to 30.04.2024, as per the VIT code of academic and research ethics.

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 in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore
Date : 9/5/24

Signature of the Guide

Internal Examiner

External Examiner

Head of the Department

Programme

ACKNOWLEDGEMENTS

First and foremost, I am immensely grateful to Prof. Athira K for her invaluable guidance and unwavering support throughout this endeavor. Her mentorship has not only provided us with a platform to apply theoretical knowledge from our textbooks but has also steered us through the intricacies of project development. This opportunity has been instrumental in our journey of comprehending real-world challenges and devising effective solutions.

I extend my heartfelt appreciation to our esteemed university for furnishing us with the resources and infrastructure necessary to showcase the culmination of our academic learnings. It is through such institutional support that we are empowered to translate classroom teachings into tangible outcomes.

Moreover, I owe a debt of gratitude to my friends whose encouragement and persuasion propelled me to embark on and successfully complete this task. Their unwavering belief in my abilities served as a constant source of motivation throughout this journey.

Last but certainly not least, I express my profound thanks to every individual who has contributed, directly or indirectly, to the fruition of this project. Their collective efforts have been pivotal in its successful completion.

The purpose of this document is to present a prediction model aimed at early detection of skin cancer. This project has served as a crucible for learning about artificial intelligence and machine learning methodologies, equipping us with the requisite skills to implement machine learning models effectively.

Snehil Sinha

Executive Summary

Our project introduces a groundbreaking approach to skin lesion segmentation by harnessing the power of ensemble learning techniques and integrating two state-of-the-art neural network architectures: UNet and SegNet. Skin lesion segmentation plays a crucial role in the early detection and treatment of melanoma, a deadly form of skin cancer. However, existing segmentation methods often struggle with accuracy and robustness, particularly in the presence of irregular shapes and noise interference.

To address these challenges, we propose a novel fusion of UNet and SegNet, capitalizing on UNet's boundary precision and SegNet's noise reduction capabilities. By averaging pixel values from their predictions, we achieve more accurate and robust segmentations, enhancing the overall quality of skin lesion analysis. Furthermore, to mitigate model limitations and improve segmentation outcomes, we employ ensemble learning with Bagging.

In our approach, multiple instances of the model, each benefiting from the fusion of UNet and SegNet, are trained on dataset subsets. This ensures a balanced mix of both architectures' strengths, resulting in refined and highly reliable segmentation outcomes. By combining the complementary abilities of UNet and SegNet, our project not only improves segmentation accuracy but also enhances the robustness of skin lesion analysis.

The significance of our project lies in its potential to revolutionize dermatological image analysis, paving the way for more effective diagnostic tools and improved patient outcomes. Through the integration of ensemble learning with cutting-edge neural network architectures, we contribute to the advancement of early detection and treatment strategies for skin lesions, ultimately saving lives and improving quality of care.

In conclusion, our project represents a significant milestone in the field of medical imaging, demonstrating the power of ensemble learning in enhancing skin lesion segmentation and advancing the fight against melanoma. We believe that our innovative approach holds promise for future research and applications in dermatology, driving progress towards more accurate and reliable diagnostic solutions.

CONTENTS		Page No.
	Acknowledgement	**i.**
	Executive Summary	**ii.**
	Table of Contents	**iii.**
	List of Figures (if any)	**iv.**
	List of Tables (if any)	**v.**
	Abbreviations	**vi.**
	Symbols and Notations (if any)	**vii.**
1	**INTRODUCTION**	1
	1.1 Objectives	1
	1.2 Motivation	2
	1.3 Background	3
2	**PROJECT DESCRIPTION AND GOALS**	4
	2.1 Survey on Existing System	4-8
	2.2 Research Gap	8-11
	2.3 Problem Statement	11-12
3	**TECHNICAL SPECIFICATION**	12
	3.1 Requirements 3.1.1 Functional 3.1.2 Non-Functional	**12-13**
	3.2 Feasibility Study 3.2.1 Technical Feasibility 3.2.2 Economic Feasibility 3.2.3 Social Feasibility	**13-16**
	3.3 System Specification 3.3.1 Hardware Specification 3.3.2 Software Specification 3.3.3 Standards and Policies (as **applicable for the project)**	**16-17**
4	**DESIGN APPROACH AND DETAILS**	**18**
	4.1 System Architecture	**18**

	4.2 Design 4.2.1 Data Flow Diagram 4.2.2 Use Case Diagram 4.2.3 Class Diagram 4.2.4 Sequence Diagram	**19-21**
	4.3 Constraints, Alternatives and Tradeoffs	**21-24**
5	**SCHEDULE, TASKS AND MILESTONES**	**24**
	5.1 Gantt Chart	**24**
	5.2 Module Description 5.2.1 Module - 1 5.2.2 Module - 2 5.2.3 …………	**25-30**
	5.3 Testing 5.3.1 Unit Testing 5.3.2 Integration Testing	**30**
6	**PROJECT DEMONSTRATION**	**31-34**
7	**COST ANALYSIS / RESULT & DISCUSSION (as applicable)**	**35-42**
8	**SUMMARY**	**43**
9	**REFERENCES**	**44-45**
	APPENDIX A – SAMPLE CODE	

List of Figures

Figure No.	Title	Page No.
1.1	Data Flow Diagram	20
1.2	Class Diagram	21
2.0	Gantt Chart	24
3.1	Unet Diagram	26
3.2	SegNet Diagram	28
4.1	Image from PH2	31
4.2	Image augmentation	34
5.1	PH2 result on unet	35
5.2	PH2 result SegNet	36
6.1	PH2 ensemble	37
6.2	PH2 Mask	38
7.1	HAM1000	40
7.2	ISIC2018	41

List of Tables

Table No.	Title	Page No.
1.1	Unet Layers	24
1.2	SegNet Layers	26
2.1	Unet 1 epoach	35
2.2	Unet 100 epoach	36
3.1	SegNet 1 epoach	37
3.2	SegNet 100 epoach	37
4.1	HAM1000 result on Unet	40
4.2	HAM1000 result on SegNet	40
5.1	ISIC2018 result on Unet	41
5.2	ISIC2018 result on SegNet	41

1. INTRODUCTION

1.1. OBJECTIVE

a. Highly Accurate Segmentation: The framework leverages a combination of UNet and SegNet architectures, along with ensemble learning techniques, to achieve highly accurate skin lesion segmentation.

b. Effective Disease Classification: The framework includes a symptom lesion classification stage, enabling effective disease classification for both rare and common diseases. This ensures that the proposed approach addresses the comprehensive assessment of skin lesions and diseases, contributing to early detection and precise treatment strategies.

c. Rigorous Evaluation: Extensive evaluations on benchmark datasets are essential to rigorously assess the framework's accuracy in both segmentation and classification tasks. By subjecting the proposed approach to comprehensive testing, its robustness and reliability can be effectively demonstrated.

d. Versatility and Efficiency: Generalization across varied datasets is prioritized, ensuring seamless integration into dermatological image analysis workflows. This emphasis on versatility and efficiency enhances the framework's applicability in diverse clinical settings, promoting widespread adoption and impact.

e. State-of-the-Art Comparison: Comprehensive comparisons with existing methods are crucial to showcase the framework's advancements and contributions to the field. By illustrating the superiority of the proposed approach through rigorous comparisons, it solidifies its position as a state-of-the-art solution for skin lesion segmentation and disease classification.

Fulfilling these objectives will not only underscore the framework's potential but also highlight its significant contributions to the field of dermatological image analysis. Thorough evaluations, comprehensive testing, and in-depth comparisons will collectively establish the proposed approach as a pioneering solution that enhances the accuracy, efficiency, and comprehensive assessment of skin lesions and diseases.

1.2. MOTIVATION

Skin cancer is a significant public health threat, and early, accurate diagnosis is crucial for improving patient outcomes. However, traditional methods of skin lesion segmentation and disease classification are **slow, subjective, and require expert dermatologists**.

This motivates the development of **automated** systems for skin cancer analysis. Deep learning techniques, particularly convolutional neural networks like UNet and SegNet, have shown remarkable promise in image analysis. These models can learn to identify and segment skin lesions with high accuracy, **reducing the burden on dermatologists** and **increasing consistency** in diagnoses.

However, a major challenge lies in the **imbalance** often present in skin cancer datasets. Certain types of skin lesions, like melanomas, are less frequent than benign moles. Unaddressed, this imbalance can lead to models prioritizing the more common classes and missing crucial details in rarer, potentially cancerous lesions.

To overcome this challenge, we can explore **innovative loss functions** that penalize misclassifications of the less frequent classes more heavily. Additionally, **ensemble learning techniques** such as bagging can be employed to combine the predictions of multiple deep learning models, further improving the overall accuracy and robustness of the system.

By effectively leveraging deep learning and addressing data imbalance issues, we can create a powerful tool for automated skin cancer diagnosis. This can significantly **improve healthcare efficiency** and potentially **save lives** by enabling earlier and more accurate detection of skin cancer.

1.3. BACKGROUND

Skin cancer is a pressing public health concern, with its incidence steadily rising worldwide. Early detection and accurate diagnosis are crucial for effective treatment and improved patient outcomes. However, the traditional approach to diagnosing skin lesions relies heavily on manual segmentation by dermatologists, which is not only time-consuming but also subjective and prone to variability depending on the expertise of the practitioner.

Recognizing the limitations of manual segmentation, researchers and medical professionals have turned to automation through the application of advanced computational techniques, particularly deep learning. Recent years have witnessed significant advancements in deep learning models tailored for image analysis, with UNet and SegNet emerging as prominent examples. These models offer the capability to automatically segment skin lesions from medical images, providing a more objective and consistent approach to diagnosis.

Despite the promise of automation, challenges persist in the realm of dermatological diagnosis. One such challenge is the inherent imbalance within skin lesion datasets, where certain classes of lesions are disproportionately represented. This class imbalance can adversely affect the performance of machine learning models, leading to biased predictions and reduced diagnostic accuracy.

To address this imbalance, researchers have explored innovative techniques such as specialized loss functions and ensemble learning methods like Bagging. These approaches aim to rebalance the dataset and improve the generalization ability of the models, thereby enhancing their performance in real-world clinical settings.

Overall, the integration of automated skin lesion segmentation and disease classification powered by deep learning represents a significant advancement in the field of dermatology. By overcoming the limitations of manual segmentation and addressing data imbalances, these computational techniques have the potential to revolutionize the way skin cancer is diagnosed and treated, ultimately leading to better outcomes for patients worldwide.

2. PROJECT DESCRIPTION AND GOALS

2.1. Survey on Existing System

Li and Shen's LIN Network:

Li and Shen propose LIN, a novel deep learning network designed for skin lesion analysis specifically aimed at aiding in melanoma detection. The LIN network integrates a Local Information Aggregation (LIA) module to capture contextual information and a Feature Enhancement (FE) module to improve feature representation. Their study demonstrates LIN's superior performance in comparison to other existing deep learning techniques for skin lesion segmentation, dermoscopic feature extraction, and lesion classification.

Codella, Rotemberg, and Tschandl's ISIC Challenge:

Codella, Rotemberg, and Tschandl discuss a significant initiative in skin lesion analysis hosted by the International Skin Imaging Collaboration (ISIC). The challenge focused on various tasks, including segmenting skin lesions, detecting attributes of lesions, and classifying diseases, utilizing a dataset comprising over 12,500 images of skin lesions. Despite the participation of top-performing algorithms, the challenge revealed limitations in accurately segmenting all lesions, indicating opportunities for further improvement in algorithm development for skin lesion analysis.

Ma et al.'s ULFAC-Net:

Ma et al. introduced ULFAC-Net, a novel solution tailored to address the challenge of accurately segmenting skin lesions characterized by irregular shapes, fuzzy contours, and noise interference. ULFAC-Net adopts a unique approach by incorporating a Parallel Asymmetric Convolutional (PAC) module for efficient feature extraction and a lightweight textual information submodule. This innovative design not only enhances segmentation performance but also mitigates the computational burden associated with existing deep learning methods. Ma et al. demonstrated ULFAC-Net's competitive segmentation performance, boasting a lightweight footprint of 0.842 million parameters and 3.71 GFLOPs, positioning it as a promising contender in the realm of skin lesion analysis.

Hao, Wu, et al.'s CACDU-Net:

Hao, Wu, et al. proposed CACDU-Net, recognizing the critical importance of automated skin lesion segmentation in early disease detection. Their approach aims to enhance segmentation accuracy by addressing the complexities inherent in skin lesion characteristics. Leveraging the DoubleU-Net framework, CACDU-Net integrates a pre-trained ConvNeXt-T for robust feature extraction, ConvNeXt Attention Convolutional Blocks (CACB) to refine features, and an Asymmetric Convolutional Atrous Spatial Pyramid Pooling (ACASPP) module for contextual information. Evaluated on datasets including ISIC2018 and PH2, CACDU-Net showcases superior performance across key image segmentation metrics, highlighting its potential as an effective tool for advancing automated skin lesion analysis.

Hosny et al.'s Review:

Hosny et al. conducted an extensive review focusing on skin lesion segmentation algorithms aimed at facilitating computer-aided diagnosis in melanoma detection. Their review encompasses a comprehensive examination of both traditional techniques and modern approaches, including deep learning and optimization-based methods. The authors meticulously evaluate the strengths and weaknesses of these various methods, considering commonly used datasets and performance metrics. This thorough review contributes significantly to refining skin lesion analysis and advancing efficient segmentation techniques, providing valuable insights for researchers and practitioners in the field.

Kaur, Ranade, et al.'s Approach:

Kaur, Ranade, et al. aimed to enhance skin lesion segmentation using deep learning techniques. Their approach introduces a modified architecture featuring a group normalization layer, which addresses semantic information enhancement by reducing batch size dependency. Additionally, they propose a novel combined loss function that merges binary focal loss and dice loss functions to optimize learning and minimize errors effectively. Furthermore, morphological operations are employed to augment pre-processing for improved dermoscopic segmentation. Experimental results on ISIC 2016 and 2017 datasets demonstrate the method's superiority, achieving impressive accuracy values of 0.95 and 0.91, respectively, thereby showcasing its potential as a promising tool for accurate skin lesion segmentation.

Liu, Zhuang, et al.'s M-VAN Unet:

Liu, Zhuang, et al. proposed M-VAN Unet, a novel approach that combines the Visual Attention Network (VAN) concept with the Transformer concept for medical image segmentation. Recognizing the limitations of convolutional neural networks in effectively capturing both local and global features, the team aimed to enhance feature extraction. M-VAN Unet introduces two attention mechanisms: MSC-Attention for multi-scale channel attention and LKA-Cross-Attention based on large kernel attention (LKA), to facilitate feature learning and global interaction. Through extensive experiments, M-VAN Unet demonstrates superior performance in key evaluation metrics, showcasing its potential to advance medical image segmentation techniques by effectively capturing both local and global features.

Alahmadi et al.'s MSAU-Net:

Alahmadi et al. aimed to address the challenges of skin lesion segmentation with their proposed MSAU-Net, a Multi-Scale Attention U-Net. The approach targets limitations in capturing multi-scale variations by integrating an attention mechanism at the bottleneck of a U-Net architecture. The introduced attention module aggregates multi-level representations to selectively adjust features, enhancing the network's ability to capture fine-grained details across different scales. Evaluated on ISIC 2017, ISIC 2018, and PH2 datasets, MSAU-Net demonstrates superior performance compared to existing alternatives, achieving dice scores that improve segmentation performance by 2%. This highlights the effectiveness of MSAU-Net in addressing the challenges of skin lesion segmentation by leveraging multi-scale attention mechanisms.

Cao, W. et al.'s ICL-Net:

Cao, W. et al. introduced ICL-Net, a solution designed for accurate skin lesion segmentation in computer-aided melanoma diagnosis. Evaluated on public skin lesion segmentation datasets, ICL-Net demonstrates its superiority in terms of segmentation performance, effectively addressing the complexities associated with diverse skin lesion shapes and sizes. This underscores the effectiveness of ICL-Net as a robust tool for facilitating accurate and reliable segmentation in computer-aided melanoma diagnosis.

Jiang et al.'s CSARM-CNN:

Jiang et al. proposed CSARM-CNN, an end-to-end framework tailored for automatic and efficient skin lesion segmentation. Through evaluation on ISIC 2017 and PH2 datasets, CSARM-CNN achieves competitive results in specificity (99.03% and 99.45%) and accuracy (94.96% and 95.23%). This research offers an efficient and automatic solution for accurate skin lesion segmentation, showcasing the potential of CSARM-CNN to streamline the segmentation process and enhance diagnostic accuracy in clinical settings.

Ramadan, Aly, and Abdel-Atty's CIU-Net:

Ramadan, Aly, and Abdel-Atty presented CIU-Net, a novel approach for skin lesion semantic segmentation. Evaluation on the ISIC 2018 dataset demonstrates CIU-Net's superiority, achieving the best Dice (92.56%) and Jaccard coefficient (91.40%) compared to recent approaches. Leveraging color invariance and attentive architecture, CIU-Net contributes to improved skin lesion segmentation, highlighting its effectiveness in capturing fine-grained details and accurately delineating lesion boundaries.

Lee et al.'s PWStE Method:

Lee et al. aimed to enhance multi-type skin lesion semantic segmentation through their "Progressive Weighted Self-Training Ensemble (PWStE)" method. Evaluation on the Multi-Type Skin Lesion Label Database (MSLD) demonstrated that PWStE achieved comparable results to Supervised Learning, while utilizing 30% less labeled data. The PWStE method effectively enhances multi-type skin lesion segmentation by leveraging self-training techniques and progressive weighting, showcasing its potential to improve segmentation accuracy and efficiency in dermatological diagnosis.

Lama, Norsang, et al.'s IPO-based Sine Function:

Lama, Norsang, et al. proposed a deep-learning method for skin lesion segmentation in dermoscopic images. Test results indicated that the IPO-based Sine function achieved superior segmentation results, boasting an accuracy of 96.23%, sensitivity of 66.48%, specificity of 99.45%, dice coefficient of 67.43%, and Jaccard coefficient of 53.63%. Leveraging innovative techniques such as the IPO-based Sine function, this approach demonstrates promising capabilities in accurately delineating skin lesions in dermoscopic images, highlighting its potential to enhance diagnostic accuracy in dermatological applications.

2.2. Research Gap

Li and Shen's LIN Network:

While Li and Shen's LIN network presents a promising advancement in automated skin lesion analysis, there remains a gap in the research regarding the generalizability of their results. Their study acknowledges that the training dataset was obtained from a single medical center, raising questions about how well the LIN network's performance would translate to other datasets with potentially different characteristics. Further investigation is needed to assess the robustness and applicability of LIN across diverse datasets and clinical settings.

Codella, Rotemberg, and Tschandl's ISIC Challenge:

The ISIC challenge outlined by Codella, Rotemberg, and Tschandl highlights another crucial research gap in the field of skin lesion analysis. Despite the extensive participation and development of advanced algorithms, the challenge revealed persistent difficulties in accurately segmenting all lesions within the provided dataset. This suggests a need for continued research and innovation to address the complexities and nuances of skin lesion segmentation, particularly in scenarios where lesions exhibit diverse characteristics and attributes.

Ma et al.'s ULFAC-Net:

While Ma et al.'s ULFAC-Net presents a novel approach to skin lesion segmentation, there remains a gap in the research concerning its applicability to diverse datasets and clinical scenarios. The study demonstrates ULFAC-Net's effectiveness in addressing specific challenges such as irregular shapes and noise interference, but further investigation is needed to evaluate its performance across a broader spectrum of skin lesion characteristics and imaging conditions. Additionally, the computational efficiency of ULFAC-Net raises questions about its scalability and generalizability to larger datasets and real-world clinical environments.

Hao, Wu, et al.'s CACDU-Net:

Despite the superior performance showcased by CACDU-Net in skin lesion segmentation, there exists a gap in understanding its robustness and scalability beyond the evaluated datasets. While the study demonstrates CACDU-Net's effectiveness on ISIC2018 and PH2 datasets, further research is required to assess its performance across diverse datasets with varying lesion characteristics and imaging conditions. Additionally, the generalizability of CACDU-Net to real-world clinical settings warrants investigation to ensure its practical utility in aiding early disease detection and clinical decision-making.

Liu, Zhuang, et al.'s M-VAN Unet:

While Liu, Zhuang, et al.'s M-VAN Unet presents promising results in medical image segmentation, there remains a gap in understanding its performance across diverse datasets and clinical scenarios. Further research is needed to evaluate the generalizability of M-VAN Unet and its robustness in handling variations in imaging conditions and lesion characteristics. Additionally, comparative studies against state-of-the-art approaches would provide insights into the unique advantages and potential limitations of M-VAN Unet, guiding its further development and optimization for practical applications in medical image analysis.

Alahmadi et al.'s MSAU-Net:

Although Alahmadi et al.'s MSAU-Net demonstrates superior performance in skin lesion segmentation, there exists a gap in understanding its scalability and generalizability to larger and more diverse datasets. Further investigation is warranted to assess the method's performance across various lesion types, imaging modalities, and clinical settings. Additionally, comparative studies against a wider range of existing approaches would provide insights into the relative strengths and weaknesses of MSAU-Net, guiding its further refinement and optimization for real-world applications in dermatology and melanoma detection.

Cao, W. et al.'s ICL-Net:

While Cao, W. et al.'s ICL-Net showcases promising results in skin lesion segmentation, there remains a gap in understanding its performance across diverse datasets and clinical scenarios. Further research is needed to assess the generalizability of ICL-Net and its robustness in handling variations in lesion characteristics and imaging conditions. Additionally, comparative studies against state-of-the-art approaches would provide insights into the unique advantages and potential limitations of ICL-Net, guiding its further optimization for practical applications in computer-aided melanoma diagnosis.

Jiang et al.'s CSARM-CNN:

Although Jiang et al.'s CSARM-CNN demonstrates competitive results in skin lesion segmentation, there exists a gap in understanding its scalability and generalizability to larger and more diverse datasets. Further investigation is warranted to assess the method's performance across various lesion types, imaging modalities, and clinical settings. Additionally, comparative studies against a wider range of existing approaches would provide insights into the relative strengths and weaknesses of CSARM-CNN, guiding its further refinement and optimization for real-world applications in dermatology and melanoma detection.

Ramadan, Aly, and Abdel-Atty's CIU-Net:

While Ramadan, Aly, and Abdel-Atty's CIU-Net exhibit superior performance in skin lesion segmentation, there remains a gap in understanding its performance across different datasets and lesion types. Further research is needed to assess the generalizability and robustness of CIU-Net in diverse clinical settings and imaging conditions. Additionally, comparative studies against other state-of-the-art approaches would provide insights into the unique strengths and potential limitations of CIU-Net, guiding its further development and optimization for practical applications in dermatological diagnosis and melanoma detection.

Lee et al.'s PWStE Method:

While Lee et al.'s PWStE method showcases promising results in multi-type skin lesion semantic segmentation, there remains a gap in understanding its performance across different datasets and lesion types. Further research is needed to assess the generalizability and robustness of PWStE in diverse clinical settings and imaging conditions. Additionally, comparative studies against other state-of-the-art approaches would provide insights into the unique strengths and potential limitations of PWStE, guiding its further development and optimization for practical applications in dermatological diagnosis and treatment.

Lama, Norsang, et al.'s IPO-based Sine Function:

Although Lama, Norsang, et al.'s IPO-based Sine function achieves superior segmentation results in dermoscopic images, there exists a gap in understanding its performance across different imaging modalities and lesion types. Further investigation is warranted to assess the method's generalizability and robustness in diverse clinical scenarios. Additionally, comparative studies against other deep-learning approaches would provide insights into the relative efficacy and potential limitations of the IPO-based Sine function, guiding its further refinement and optimization for practical applications in dermatological diagnosis and treatment.

2.3. Problem Statement

Skin lesion analysis plays a crucial role in early detection and diagnosis of melanoma, a deadly form of skin cancer. Traditional methods often lack the precision and efficiency required for accurate detection, leading to delayed diagnosis and treatment. To address this challenge, advanced deep learning techniques have emerged as promising tools for skin lesion segmentation and disease classification.

Additionally, imbalanced data distributions within skin lesion datasets, further complicate the training process and may lead to biased model outcomes. Moreover, the generalizability of these models across different lesion types and imaging conditions remains a concern, limiting their applicability in real-world clinical settings.

Therefore, there is a pressing need to develop a comprehensive approach that effectively integrates UNet and SegNet architectures using ensemble learning techniques like Bagging, while addressing the challenges posed by imbalanced datasets. This approach should ensure superior segmentation accuracy performance across diverse datasets, facilitating early and accurate detection of melanoma.

3. TECHNICAL SPECIFICATION

3.1. Requirements

a. Functional

I. *Segmentation:* The system must accurately segment skin lesions from surrounding healthy tissue in medical images. This involves implementing algorithms that can identify lesion boundaries with high precision.

II. *Integration:* The system must seamlessly integrate the UNet and SegNet architectures using Bagging ensemble learning technique. This involves implementing the integration logic within the system's codebase.

III. *Dataset Handling:* The system should efficiently handle and preprocess skin lesion datasets, including PH2, HAM10000, and ISIC2018. This involves writing scripts or programs to clean, normalize, and augment the datasets prior to training.

IV. *Scalability:* The system should be scalable to handle large volumes of data for real-world applications. This involves designing algorithms and architectures that can scale horizontally or vertically to accommodate increased data load.

b. Non-Functional

I. *Accuracy:* The system must achieve high accuracy in both lesion segmentation and disease classification tasks. This involves optimizing algorithms and models to minimize errors and improve diagnostic accuracy.

II. *Efficiency:* The system should be efficient in terms of computational resources and processing time. This involves optimizing algorithms and data processing pipelines to minimize resource usage and maximize throughput.

III. *Robustness:* The system must be robust and able to generalize well to diverse lesion types and imaging conditions. This involves testing the system on a variety of datasets to ensure reliable performance in different scenarios.

3.2. Feasibility Study

a. Technical Feasibility

1. Algorithmic Complexity: The UNet and SegNet architectures, along with the Bagging ensemble learning technique, have been successfully implemented on Google Colab and Kaggle Notebook using GPU acceleration. The training times for both models on various datasets are within reasonable limits, indicating that the algorithms are computationally feasible for the given task.

2. Computational Resources: The availability of GPU acceleration significantly reduces training times for both the UNet and SegNet models, making them feasible for processing large volumes of data. The CPU with 27 GB of memory is also sufficient for handling preprocessing tasks and model training on bigger datasets.

3. Data Preprocessing: The preprocessing of datasets, including PH2, HAM10000, and ISIC2018, is feasible within the given computational resources and training times. The time required for data preprocessing and augmentation is manageable, ensuring that the datasets are appropriately prepared for model training.

4. Model Training: The training times for both the UNet and SegNet models on different datasets, ranging from 8 to 40 minutes for 100 epochs, are feasible for the given number of images. This indicates that model training can be completed within a reasonable timeframe, allowing for efficient experimentation and optimization.

5. Integration: Integrating the UNet and SegNet architectures using Bagging ensemble learning technique is feasible, as both models have been successfully trained and evaluated individually. Combining their predictions and leveraging ensemble learning techniques can further improve the overall performance of the system.

6. Scalability: The scalability of the system is demonstrated by its ability to handle datasets of varying sizes, ranging from 450 images in

PH2 to 10,000 images in HAM10000. The training times scale appropriately with the size of the dataset, ensuring that the system can accommodate larger volumes of data as needed.

7. Software Development: Developing the necessary software components, including data preprocessing pipelines, model training scripts, and deployment infrastructure, is feasible using Google Colab and Kaggle Notebook environments. These platforms provide convenient tools and resources for developing and testing deep learning models.

8. Testing and Validation: Testing and validating the system's performance on different datasets is feasible within the given computational resources and training times. Standard evaluation metrics can be used to assess the accuracy, reliability, and robustness of the models.

b. Economic Feasibility

Cost of Hardware and Software: Since we are utilizing free computational resources provided by platforms like Google Colab and Kaggle Notebook, and the datasets are freely available in the public domain, we don't incur any direct costs associated with acquiring hardware or software licenses for model development and training.

Data Acquisition and Licensing: The datasets we are using, including PH2, HAM10000, and ISIC2018, are all openly available, eliminating the need for any data acquisition or licensing costs. This significantly reduces the overall cost burden of the project.

Personnel Costs: While there may be personnel costs associated with the project, such as our time and effort, we are working on the project as part of an academic institution, which minimizes any direct financial expenses.

Training and Maintenance Costs: With the computational resources and datasets freely available, there are minimal training and maintenance costs associated with the project. We can implement updates to models and algorithms using the same free resources, ensuring ongoing development without additional expenses.

Return on Investment (ROI): Although there may not be direct monetary returns from the project in terms of revenue generation or cost savings, the potential impact on healthcare outcomes, such as early detection of melanoma and improved patient care, can be substantial. This intangible return on investment motivates us to pursue the project despite minimal costs involved.

c. Social Feasibility

Healthcare Impact: The primary social benefit of our project is its potential to improve healthcare outcomes, particularly in the early detection and diagnosis of melanoma. By developing accurate and efficient algorithms for skin lesion analysis, our project aims to assist healthcare professionals in identifying suspicious lesions and facilitating timely interventions, ultimately leading to improved patient outcomes and potentially saving lives.

Public Health Awareness: Our project contributes to raising awareness about skin cancer, particularly melanoma, which is a significant public health concern worldwide. By developing advanced tools for melanoma detection, our project helps educate the public about the importance of regular skin checks and early detection strategies, empowering individuals to take proactive steps towards skin cancer prevention and self-care.

Accessibility and Equity: By leveraging freely available datasets and computational resources, our project promotes accessibility and equity in healthcare research and innovation. The open availability of data and tools ensures that researchers and healthcare professionals from diverse backgrounds can contribute to the advancement of skin cancer detection technologies, regardless of financial constraints or institutional affiliations.

Ethical Considerations: We recognize the importance of ethical considerations in developing and deploying AI-based healthcare solutions. Our project prioritizes patient privacy, data security, and informed consent, adhering to established ethical guidelines and regulatory frameworks to ensure the responsible and ethical use of sensitive medical data.

3.3. System Specification

a. Hardware Specification

Our project on skin lesion analysis for melanoma detection primarily relies on computational resources for model development, training, and inference. While we utilize cloud-based platforms for most of our computational needs, it's essential to outline the hardware specifications required for efficient execution:

GPU (Graphics Processing Unit):

Our project benefits significantly from GPU acceleration, which accelerates deep learning model training and inference. We recommend using GPUs with CUDA support, as they offer optimal performance for deep learning tasks. GPU used here are 16 GB of Tesla GPUs

CPU (Central Processing Unit):

While GPU acceleration is essential for model training, a capable CPU is also required for preprocessing tasks, data manipulation, and overall system operation. We recommend CPUs with multiple cores and threads to handle parallel processing efficiently.A CPU with a clock speed of at least 2.5 GHz or higher is preferable to ensure smooth operation during intensive computational tasks.

Memory (RAM):

Sufficient RAM is crucial for loading and processing large datasets, especially during model training. We recommend a minimum of 16 GB of RAM for small to medium-sized datasets. However, for larger datasets such as HAM10000 and ISIC2018, 32 GB or higher may be necessary to avoid memory constraints during training.

b. Software Specification

Our project on skin lesion analysis for melanoma detection requires various software tools and frameworks for model development, training, evaluation, and deployment. Here are the key software specifications:

1. Python Programming Language:

- Python serves as the primary programming language for our project, offering a rich ecosystem of libraries and frameworks for machine learning and deep learning tasks.
- We utilize Python for data preprocessing, model development, training, evaluation, and deployment.

2. Deep Learning Frameworks:

- We leverage deep learning frameworks to implement and train our skin lesion analysis models efficiently.
- TensorFlow are the main deep learning frameworks used in our project. These frameworks provide high-level APIs for building, training, and deploying deep neural networks.

3. Keras:

- Keras serves as a high-level neural networks API, running on top of TensorFlow or Theano. It allows for easy and fast prototyping of neural network models.
- We use Keras for rapid experimentation and prototyping of model architectures, particularly during the initial stages of development.

4. Google Colab and Kaggle Notebook:

- Google Colab and Kaggle Notebook are cloud-based platforms that offer free access to GPUs and TPUs for running Python code.
- We utilize Google Colab and Kaggle Notebook for model training and experimentation, taking advantage of their free computational resources to accelerate deep learning tasks.

4. DESIGN APPROACH AND DETAILS

3.1 System Architecture: -

Our skin lesion analysis system is designed to accurately detect and classify melanoma using a combination of deep learning models, ensemble learning techniques, and advanced image analysis algorithms. The system architecture is meticulously crafted to leverage the strengths of various components while ensuring scalability, performance, and reliability.

Components:

Data Preprocessing Module:

This component handles the preprocessing of input images, including resizing, normalization, and augmentation. Preprocessed images are then fed into the deep learning models for training and inference.

UNet and SegNet Models:

The UNet and SegNet models serve as the primary deep learning architectures for skin lesion segmentation. UNet excels in precise boundary detection, while SegNet enhances contextual understanding by capturing spatial dependencies.

Ensemble Learning (Bagging):

We employ ensemble learning techniques, specifically Bagging, to combine the predictions of multiple UNet and SegNet models. By aggregating diverse predictions, we mitigate errors and improve the overall segmentation accuracy.

Evaluation Module:

The evaluation module assesses the performance of the system by comparing the predicted segmentation masks with ground truth annotations. Evaluation metrics such as Dice coefficient, Jaccard index, and accuracy are computed to quantify the system's performance.

3.2 Design

3.2.1 Data Flow Diagram

<u>Data Flow Process:</u>

Input Data (Dermoscopic Images):

- Dermoscopic images of skin lesions are fed into the system for analysis.

Preprocessing:

- Input images undergo preprocessing steps, including resizing, normalization, and augmentation, to prepare them for model input.

Model Inference:

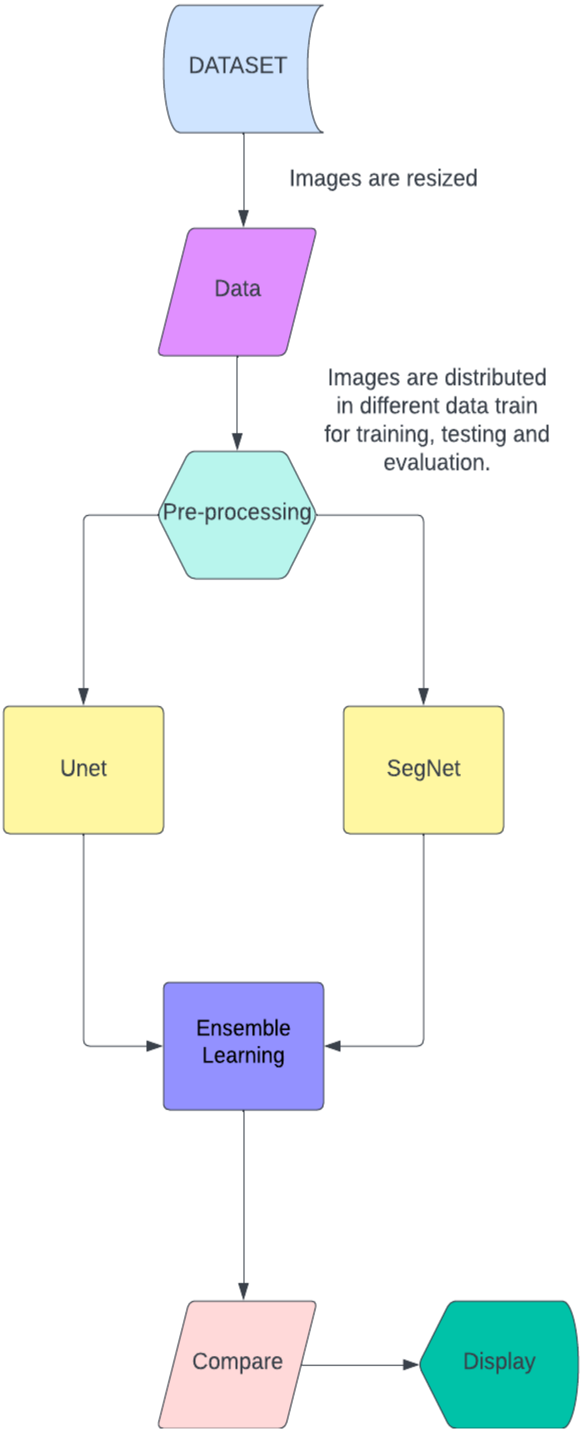
- Preprocessed images are passed through the UNet and SegNet models for segmentation. Each model generates segmentation masks for the input images.

Ensemble Learning:

- Bagging is employed to combine the segmentation masks generated by multiple UNet and SegNet models, producing a final aggregated segmentation result.

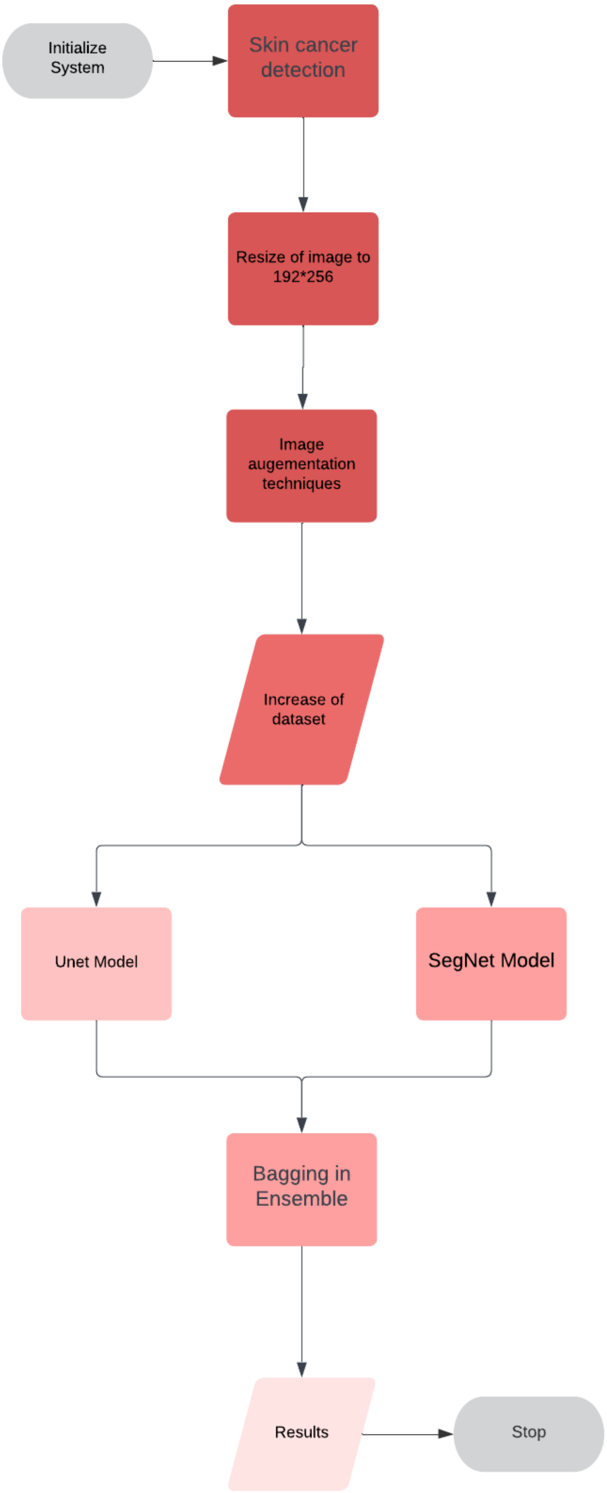
Output:

- The final segmentation masks, along with disease classification labels, are generated as the output of the system.


DATASET
Images are resized
Data
Images are distributed in different data train for training, testing and evaluation.
Pre-processing
Unet
SegNet
Ensemble Learning
Compare
Display

1.1 Data-Flow Diagram

3.2.2 Class Diagram



1.2 Class Diagram

3.3 Constraints, Alternatives and Tradeoffs

Constraints:

Computational Resources:

Our skin lesion analysis system relies heavily on computational resources, particularly GPU acceleration for training deep learning models. Limited availability of GPUs on cloud platforms or restrictions on computational budgets may constrain the scalability and performance of the system. Large datasets and complex model architectures necessitate significant computational power, potentially leading to longer training times and increased resource costs.

Alternatives:

Model Architectures:

Exploring alternative deep learning architectures beyond UNet and SegNet, such as fully convolutional networks (FCNs), attention mechanisms, or graph-based models, offers potential alternatives.

Each architecture has its own tradeoffs in terms of accuracy, computational efficiency, and interpretability, providing opportunities to optimize the system based on specific requirements.

Ensemble Learning Techniques:

Considering alternative ensemble learning techniques, such as Boosting or Stacking, instead of Bagging for combining model predictions presents alternative approaches.

Each ensemble technique offers unique advantages and limitations, requiring careful evaluation based on factors such as model diversity, computational resources, and performance goals.

Data Augmentation Strategies:

We can use various other augmentation techniques such random cropping etc along with rotation and flipping.

Tradeoffs:

Model Complexity vs. Interpretability:

Balancing model complexity with interpretability is crucial. While increasing model complexity may improve performance, it can also decrease interpretability, making it challenging to understand and trust model decisions.

Training Time vs. Model Performance:

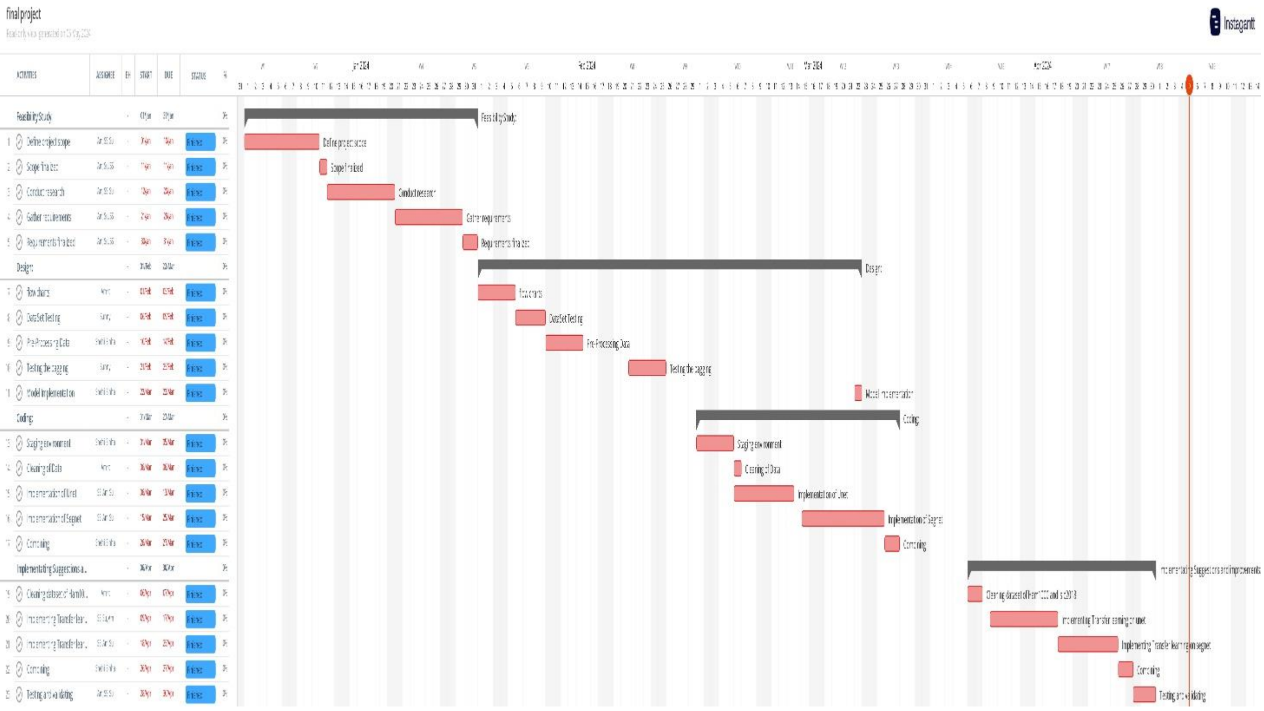
Longer training times may lead to higher model performance, but they also increase resource costs and development time. Balancing training time with model performance is essential to optimize system efficiency.

Accuracy vs. Computational Efficiency:

Tradeoffs between accuracy and computational efficiency must be evaluated. More complex models may achieve higher accuracy but require more computational resources and inference time, impacting system scalability and deployment.

5. SCHEDULE, TASKS AND MILESTONES

5.1 Gantt Chart



2.0 Gantt Chart

5.2 Model Description

1. UNet Model

Model Description:

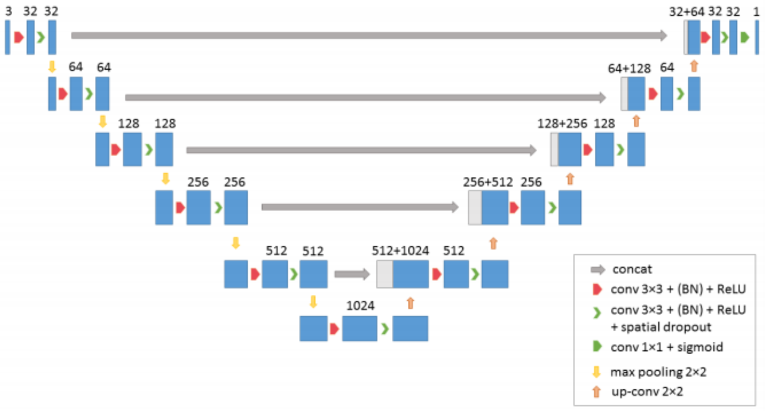
Our skin lesion segmentation model is constructed based on the UNet architecture, which is well-suited for pixel-wise segmentation tasks. UNet is known for its ability to capture intricate lesion boundaries and generate precise segmentations.

Network Architecture:

- The UNet model is designed to handle images with a resolution of 224x224 pixels.
- It consists of a series of double convolutional layers, which are key components for feature extraction and hierarchical information

processing.

- Each double convolutional layer involves two consecutive convolutional operations followed by a Rectified Linear Unit (ReLU) activation function and optional batch.
- The number of filters for the convolutional layers increases progressively along the encoding path, capturing features at different levels of abstraction.
- Max-pooling layers are applied to down-sample the feature maps, reducing spatial dimensions while retaining essential information.
- The central part of the network is characterized by a high number of filters, allowing the model to capture fine-grained details.
- The decoding path uses up-sampling layers, followed by double convolutional layers, to recover spatial information and segment the lesions effectively.
- Skip connections are used to concatenate feature maps from the encoding path to the decoding path, aiding in the precise reconstruction of segmentation masks.



3.1 Unet Daigram

1.1 Unet Layers

Sequence	Filter Size	Output Dimensions	Sequence	Filter Size	Output Dimensions
Input	-	(224, 224, 3)	Up_14	-	(14,14,1024)
Conv_224	32	(224,224,32)	Up_Conv_14	512	(14,14, 512)
MaxPooling_112	-	(112,112,32)	Up_28	-	(28, 28, 512)
Conv_112	64	(112,112,64)	Up_Conv_28	256	(28, 28, 256)
MaxPooling_56	-	(56, 56, 64)	Up_56	-	(56, 56, 256)
Conv_56	128	(56, 56, 128)	Up_Conv_56	128	(56, 56, 128)
MaxPooling_28	-	(28, 28, 128)	Up_112	-	(112,112,128)
Conv_28	256	(28, 28, 256)	Up_Conv_112	64	(112,112, 64)
MaxPooling_14	-	(14, 14, 256)	Up_224	-	(224, 224, 64)
Conv_14	512	(14, 14, 512)	Up_Conv_224	32	(224, 224, 32)
MaxPooling_7	-	(7, 7, 512)	Conv_Final	1	(224, 224, 1)
Conv_7	1024	(7, 7, 1024)	Output	-	(224, 224)

Training:

- The UNet model is trained using the Adam optimizer with a learning rate of 0.003.
- The loss function employed for training is a Jaccard distance-based loss, which is suitable for segmentation tasks.

- Multiple evaluation metrics are utilized during training, including Intersection over Union (IoU), Dice coefficient, precision, recall, and accuracy, to assess the model's performance.

- The training process involves running for a specified number of epochs, using a batch size of 18.

2. SegNet Model

Model Description:

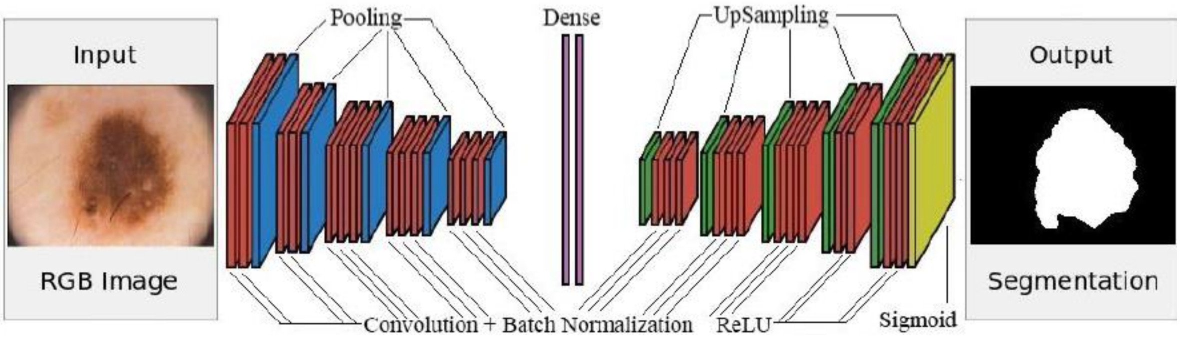
The SegNet model is designed for skin lesion segmentation, employing an architecture that combines encoding and decoding layers to produce pixel-wise segmentations.

Encoding Layer:

- The input image has dimensions (192, 256, 3).
- A series of convolutional layers are applied, starting with 64 filters. Each convolution is followed by batch normalization and a ReLU activation function.
- Max-pooling layers are utilized to down-sample the feature maps, reducing spatial dimensions while retaining important information.
- The encoding path continues with increasing numbers of filters (128, 256, and 512), each with batch normalization and ReLU activation functions.
- High-level features are captured using a 512-filter convolutional layer.
- The encoding path concludes with two fully connected (dense) layers with 1024 neurons and ReLU activation functions.

Decoding Layer:

- Up-sampling layers are used to restore spatial dimensions.
- Convolutional transpose layers are applied to up-sample the feature maps and reconstruct the segmentation masks.
- Batch normalization and ReLU activation functions are applied to the feature maps in the decoding path.
- The final convolutional transpose layer produces the binary segmentation output using a sigmoid activation function.
- The output is reshaped to match the input image dimensions of (192, 256).



3.2 SegNet Diagram

1.2 Segnet Layers

Sequence	Filter Size	Output Dimensions	Sequence	Filter Size	Output Dimensions
Input	-	(224, 224, 3)	Up_14	-	(14,14,1024)
Conv_224	32	(224,224,32)	Up_Conv_14	512	(14,14, 512)
MaxPooling_112	-	(112,112,32)	Up_28	-	(28, 28, 512)
Conv_112	64	(112,112,64)	Up_Conv_28	256	(28, 28, 256)
MaxPooling_56	-	(56, 56, 64)	Up_56	-	(56, 56, 256)
Conv_56	128	(56, 56, 128)	Up_Conv_56	128	(56, 56, 128)
MaxPooling_28	-	(28, 28, 128)	Up_112	-	(112,112,128)
Conv_28	256	(28, 28, 256)	Up_Conv_112	64	(112,112, 64)
MaxPooling_14	-	(14, 14, 256)	Up_224	-	(224, 224, 64)
Conv_14	512	(14, 14, 512)	Up_Conv_224	32	(224, 224, 32)
MaxPooling_7	-	(7, 7, 512)	Conv_Final	1	(224, 224, 1)
Conv_7	1024	(7, 7, 1024)	Output	-	(224, 224)

Training-

- The model is trained using the Stochastic Gradient Descent (SGD) optimizer with specific hyperparameters (learning rate, momentum, decay, and nesterov).
- The loss function used for training is binary cross-entropy, which is appropriate for binary segmentation tasks.
- The training process involves a specified number of epochs, with a batch size of 18.
- Multiple evaluation metrics, including Intersection over Union (IoU), Dice coefficient, precision, recall, and accuracy, are computed during training to assess model performance.

3. Image Augmentation

The procedure of image augmentation on training images has been used to increase the robustness of the model and reduce the chances of overfitting. It will also increase the data images that are available in the dataset. The two simple techniques that are used are image rotation and horizontal flipping. In the image rotation, all the images in the training set are rotated with a range [-40◦, +40◦] and flipped along the horizontal axis only.

All the above transformations are exactly performed on the corresponding masks of the images as well to maintain the correct orientation of feature images with their truth masks. After the augmentation, the transformed images are included in the training set which increases our training set from 150 to 450. Out of these 450 images, 90 have been excluded from the training set for the formation of a validation set.

4. Databases

The used dataset is the PH2 dermoscopic dataset which contains 200 dermoscopic images and their label masks. Each one is an RGB image and the fixed dimension of each image is 572 x 765. The dataset has been provided publicly for experimental and studying purposes, to facilitate research on both segmentation and classification algorithms of dermoscopic images. The database is acquired at the Dermatology Service of Hospital Pedro Hispano, Matosinhos, Portugal.

For training purposes, the dimensions of each image have been reduced to 192 x 256 before feeding it into the network. It largely reduces the parameters to be trained in the network as well as the training time and complexity without significantly affecting the results.

5.3 Testing

Performance Evaluation

The generated binary mask images in the output of the network are evaluated on different mathematical measures in comparison with the ground truth lesion masks as provided in the dataset. The accuracy is measured pixel-wise. The used measures are as below:

- **Intersection Over Union**: The Jaccard index, also known as Intersection over Union. The Jaccard similarity coefficient is a statistical similarity measure to check the diversity among the sample sets. The IOU gives the similarity among sets and the formula is the size of the intersection over the size of the union of the sets.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A|+|B|-|A \cap B|}$$

- **Dice Coefficient**: The Dice score is like precision. It measures the positives as well as applies a penalty to the false positives given by the model. It is more similar to precision than accuracy.

$$Dice = \frac{2 * TP}{(TP+FP)+(TP+FP)}$$

- **Precision**: Precision is a measure that is more focused on catching the false positives in the results of the model.

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

- **Recall**: Recall is a measure that is targeted towards the actual or the true positives yielded by the model output. In the scenarios where the cost of the False Negatives is greater than recall is the better metric to choose the best model among the possible ones.

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

- **Accuracy**:

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

6. PROJECT DEMONSTRATION

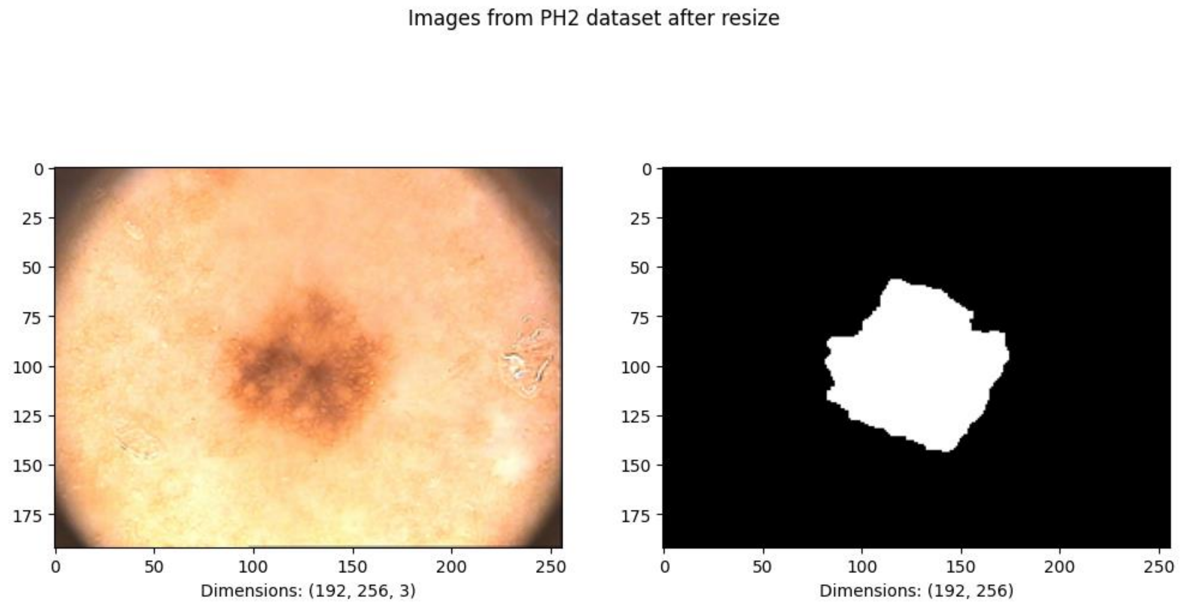
Proposed Methodology

The proposed methodology integrates UNet, SegNet, and ensemble learning (specifically,Bagging) to improve skin lesion segmentation. Initially, both UNet and SegNet models are applied independently to input images, each generating a segmentation prediction. These individual predictions are then combined to produce a more refined segmentation. The fusion process involves averaging corresponding pixel values from both UNet and SegNet predictions, capitalizing on the strengths of each architecture for enhanced accuracy and robustness.

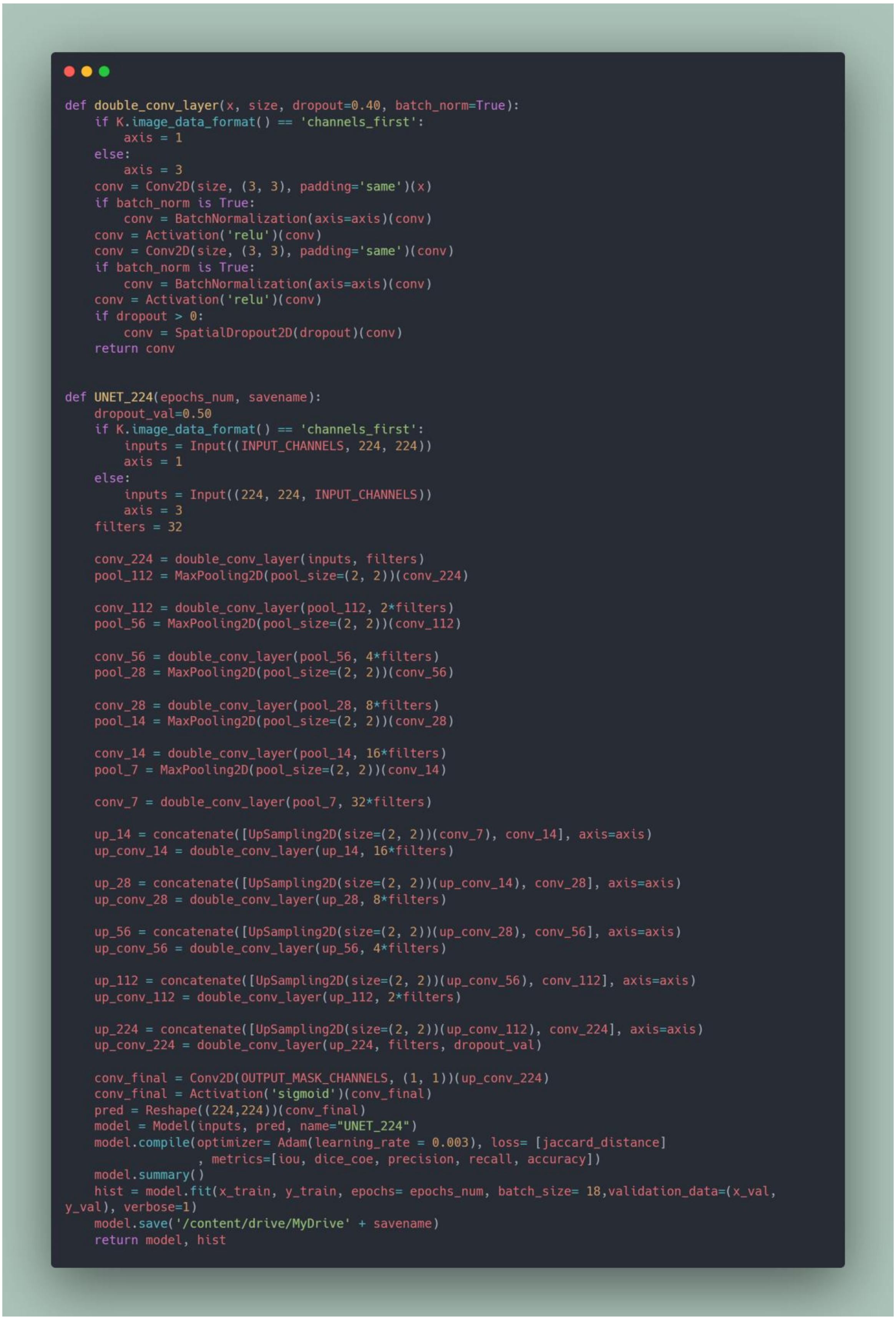
To further boost segmentation performance and address the limitations of individual models, an ensemble learning approach utilizing Bagging is introduced. Multiple iterations of the combined UNet and SegNet model with enhanced segmentation are trained on different subsets of the dataset, introducing diversity in predictions. Ultimately, the ensemble of models is leveraged to derive a consensus prediction, resulting in a more polished and reliable segmentation outcome.

This integrated approach maximizes the complementary capabilities of UNet and SegNet, while ensemble learning bolsters overall segmentation quality, rendering it a potent solution for skin lesion analysis.

4.1 Sample image from PH2 with its corresponding mask

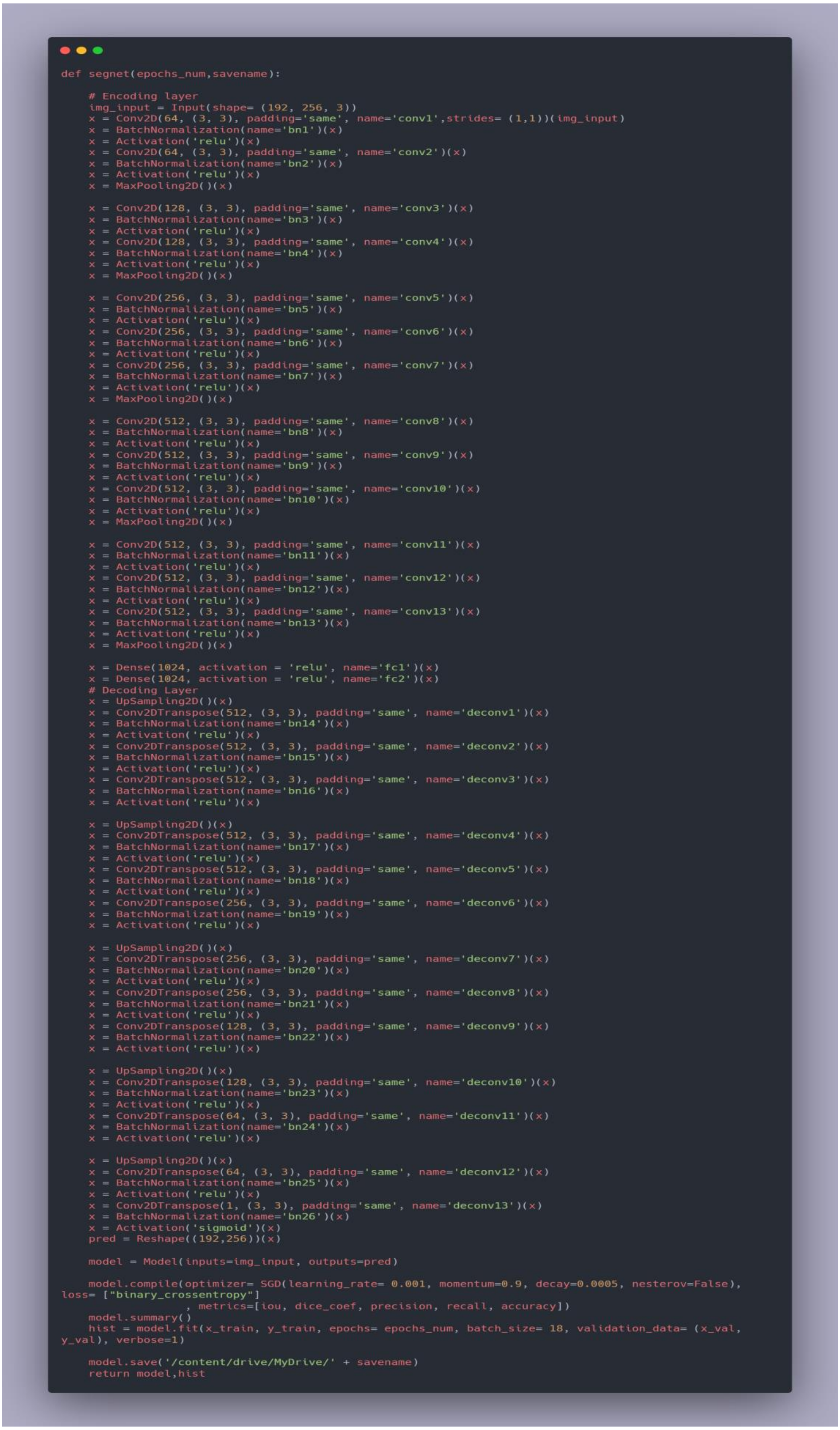

Images from PH2 dataset after resize
0
25
50
75
100
125
150
175
0
50
100
150
200
250
Dimensions: (192, 256, 3)
Dimensions: (192, 256)

UNET implementation



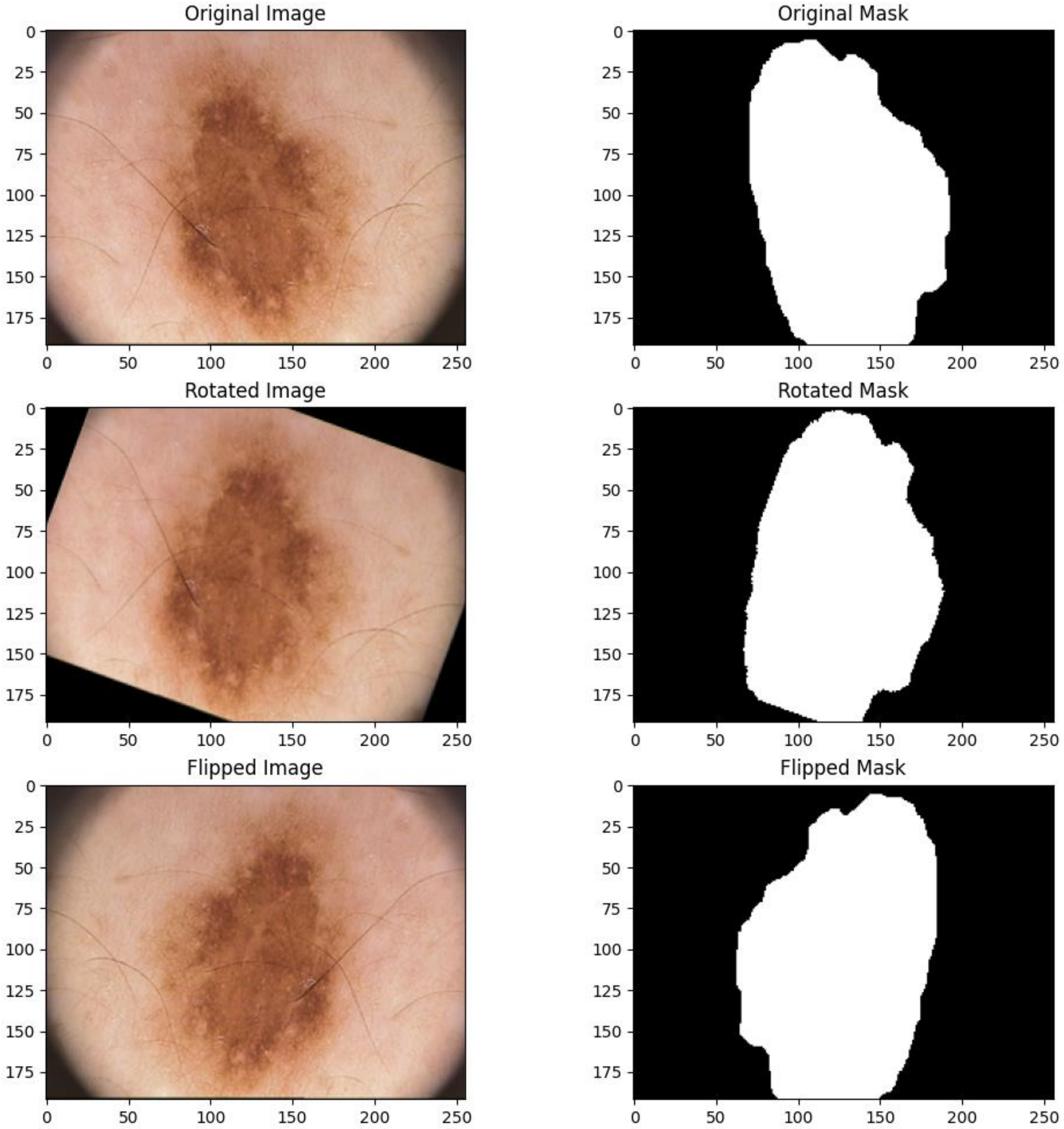
```
def double_conv_layer(x, size, dropout=0.40, batch_norm=True):
    if K.image_data_format() == 'channels_first':
        axis = 1
    else:
        axis = 3
    conv = Conv2D(size, (3, 3), padding='same')(x)
    if batch_norm is True:
        conv = BatchNormalization(axis=axis)(conv)
    conv = Activation('relu')(conv)
    conv = Conv2D(size, (3, 3), padding='same')(conv)
    if batch_norm is True:
        conv = BatchNormalization(axis=axis)(conv)
    conv = Activation('relu')(conv)
    if dropout > 0:
        conv = SpatialDropout2D(dropout)(conv)
    return conv


def UNET_224(epochs_num, savename):
    dropout_val=0.50
    if K.image_data_format() == 'channels_first':
        inputs = Input((INPUT_CHANNELS, 224, 224))
        axis = 1
    else:
        inputs = Input((224, 224, INPUT_CHANNELS))
        axis = 3
    filters = 32

    conv_224 = double_conv_layer(inputs, filters)
    pool_112 = MaxPooling2D(pool_size=(2, 2))(conv_224)

    conv_112 = double_conv_layer(pool_112, 2*filters)
    pool_56 = MaxPooling2D(pool_size=(2, 2))(conv_112)

    conv_56 = double_conv_layer(pool_56, 4*filters)
    pool_28 = MaxPooling2D(pool_size=(2, 2))(conv_56)

    conv_28 = double_conv_layer(pool_28, 8*filters)
    pool_14 = MaxPooling2D(pool_size=(2, 2))(conv_28)

    conv_14 = double_conv_layer(pool_14, 16*filters)
    pool_7 = MaxPooling2D(pool_size=(2, 2))(conv_14)

    conv_7 = double_conv_layer(pool_7, 32*filters)

    up_14 = concatenate([UpSampling2D(size=(2, 2))(conv_7), conv_14], axis=axis)
    up_conv_14 = double_conv_layer(up_14, 16*filters)

    up_28 = concatenate([UpSampling2D(size=(2, 2))(up_conv_14), conv_28], axis=axis)
    up_conv_28 = double_conv_layer(up_28, 8*filters)

    up_56 = concatenate([UpSampling2D(size=(2, 2))(up_conv_28), conv_56], axis=axis)
    up_conv_56 = double_conv_layer(up_56, 4*filters)

    up_112 = concatenate([UpSampling2D(size=(2, 2))(up_conv_56), conv_112], axis=axis)
    up_conv_112 = double_conv_layer(up_112, 2*filters)

    up_224 = concatenate([UpSampling2D(size=(2, 2))(up_conv_112), conv_224], axis=axis)
    up_conv_224 = double_conv_layer(up_224, filters, dropout_val)

    conv_final = Conv2D(OUTPUT_MASK_CHANNELS, (1, 1))(up_conv_224)
    conv_final = Activation('sigmoid')(conv_final)
    pred = Reshape((224,224))(conv_final)
    model = Model(inputs, pred, name="UNET_224")
    model.compile(optimizer= Adam(learning_rate = 0.003), loss= [jaccard_distance]
                  , metrics=[iou, dice_coe, precision, recall, accuracy])
    model.summary()
    hist = model.fit(x_train, y_train, epochs= epochs_num, batch_size= 18,validation_data=(x_val,
y_val), verbose=1)
    model.save('/content/drive/MyDrive' + savename)
    return model, hist
```

SEGNET Implementation



```
def segnet(epochs_num,savename):

    # Encoding layer
    img_input = Input(shape= (192, 256, 3))
    x = Conv2D(64, (3, 3), padding='same', name='conv1',strides= (1,1))(img_input)
    x = BatchNormalization(name='bn1')(x)
    x = Activation('relu')(x)
    x = Conv2D(64, (3, 3), padding='same', name='conv2')(x)
    x = BatchNormalization(name='bn2')(x)
    x = Activation('relu')(x)
    x = MaxPooling2D()(x)

    x = Conv2D(128, (3, 3), padding='same', name='conv3')(x)
    x = BatchNormalization(name='bn3')(x)
    x = Activation('relu')(x)
    x = Conv2D(128, (3, 3), padding='same', name='conv4')(x)
    x = BatchNormalization(name='bn4')(x)
    x = Activation('relu')(x)
    x = MaxPooling2D()(x)

    x = Conv2D(256, (3, 3), padding='same', name='conv5')(x)
    x = BatchNormalization(name='bn5')(x)
    x = Activation('relu')(x)
    x = Conv2D(256, (3, 3), padding='same', name='conv6')(x)
    x = BatchNormalization(name='bn6')(x)
    x = Activation('relu')(x)
    x = Conv2D(256, (3, 3), padding='same', name='conv7')(x)
    x = BatchNormalization(name='bn7')(x)
    x = Activation('relu')(x)
    x = MaxPooling2D()(x)

    x = Conv2D(512, (3, 3), padding='same', name='conv8')(x)
    x = BatchNormalization(name='bn8')(x)
    x = Activation('relu')(x)
    x = Conv2D(512, (3, 3), padding='same', name='conv9')(x)
    x = BatchNormalization(name='bn9')(x)
    x = Activation('relu')(x)
    x = Conv2D(512, (3, 3), padding='same', name='conv10')(x)
    x = BatchNormalization(name='bn10')(x)
    x = Activation('relu')(x)
    x = MaxPooling2D()(x)

    x = Conv2D(512, (3, 3), padding='same', name='conv11')(x)
    x = BatchNormalization(name='bn11')(x)
    x = Activation('relu')(x)
    x = Conv2D(512, (3, 3), padding='same', name='conv12')(x)
    x = BatchNormalization(name='bn12')(x)
    x = Activation('relu')(x)
    x = Conv2D(512, (3, 3), padding='same', name='conv13')(x)
    x = BatchNormalization(name='bn13')(x)
    x = Activation('relu')(x)
    x = MaxPooling2D()(x)

    x = Dense(1024, activation = 'relu', name='fc1')(x)
    x = Dense(1024, activation = 'relu', name='fc2')(x)
    # Decoding Layer
    x = UpSampling2D()(x)
    x = Conv2DTranspose(512, (3, 3), padding='same', name='deconv1')(x)
    x = BatchNormalization(name='bn14')(x)
    x = Activation('relu')(x)
    x = Conv2DTranspose(512, (3, 3), padding='same', name='deconv2')(x)
    x = BatchNormalization(name='bn15')(x)
    x = Activation('relu')(x)
    x = Conv2DTranspose(512, (3, 3), padding='same', name='deconv3')(x)
    x = BatchNormalization(name='bn16')(x)
    x = Activation('relu')(x)

    x = UpSampling2D()(x)
    x = Conv2DTranspose(512, (3, 3), padding='same', name='deconv4')(x)
    x = BatchNormalization(name='bn17')(x)
    x = Activation('relu')(x)
    x = Conv2DTranspose(512, (3, 3), padding='same', name='deconv5')(x)
    x = BatchNormalization(name='bn18')(x)
    x = Activation('relu')(x)
    x = Conv2DTranspose(256, (3, 3), padding='same', name='deconv6')(x)
    x = BatchNormalization(name='bn19')(x)
    x = Activation('relu')(x)

    x = UpSampling2D()(x)
    x = Conv2DTranspose(256, (3, 3), padding='same', name='deconv7')(x)
    x = BatchNormalization(name='bn20')(x)
    x = Activation('relu')(x)
    x = Conv2DTranspose(256, (3, 3), padding='same', name='deconv8')(x)
    x = BatchNormalization(name='bn21')(x)
    x = Activation('relu')(x)
    x = Conv2DTranspose(128, (3, 3), padding='same', name='deconv9')(x)
    x = BatchNormalization(name='bn22')(x)
    x = Activation('relu')(x)

    x = UpSampling2D()(x)
    x = Conv2DTranspose(128, (3, 3), padding='same', name='deconv10')(x)
    x = BatchNormalization(name='bn23')(x)
    x = Activation('relu')(x)
    x = Conv2DTranspose(64, (3, 3), padding='same', name='deconv11')(x)
    x = BatchNormalization(name='bn24')(x)
    x = Activation('relu')(x)

    x = UpSampling2D()(x)
    x = Conv2DTranspose(64, (3, 3), padding='same', name='deconv12')(x)
    x = BatchNormalization(name='bn25')(x)
    x = Activation('relu')(x)
    x = Conv2DTranspose(1, (3, 3), padding='same', name='deconv13')(x)
    x = BatchNormalization(name='bn26')(x)
    x = Activation('sigmoid')(x)
    pred = Reshape((192,256))(x)

    model = Model(inputs=img_input, outputs=pred)

    model.compile(optimizer= SGD(learning_rate= 0.001, momentum=0.9, decay=0.0005, nesterov=False),
loss= ["binary_crossentropy"]
                   , metrics=[iou, dice_coef, precision, recall, accuracy])
    model.summary()
    hist = model.fit(x_train, y_train, epochs= epochs_num, batch_size= 18, validation_data= (x_val,
y_val), verbose=1)

    model.save('/content/drive/MyDrive/' + savename)
    return model,hist
```

4.2 Image after performing of augmentation technique-



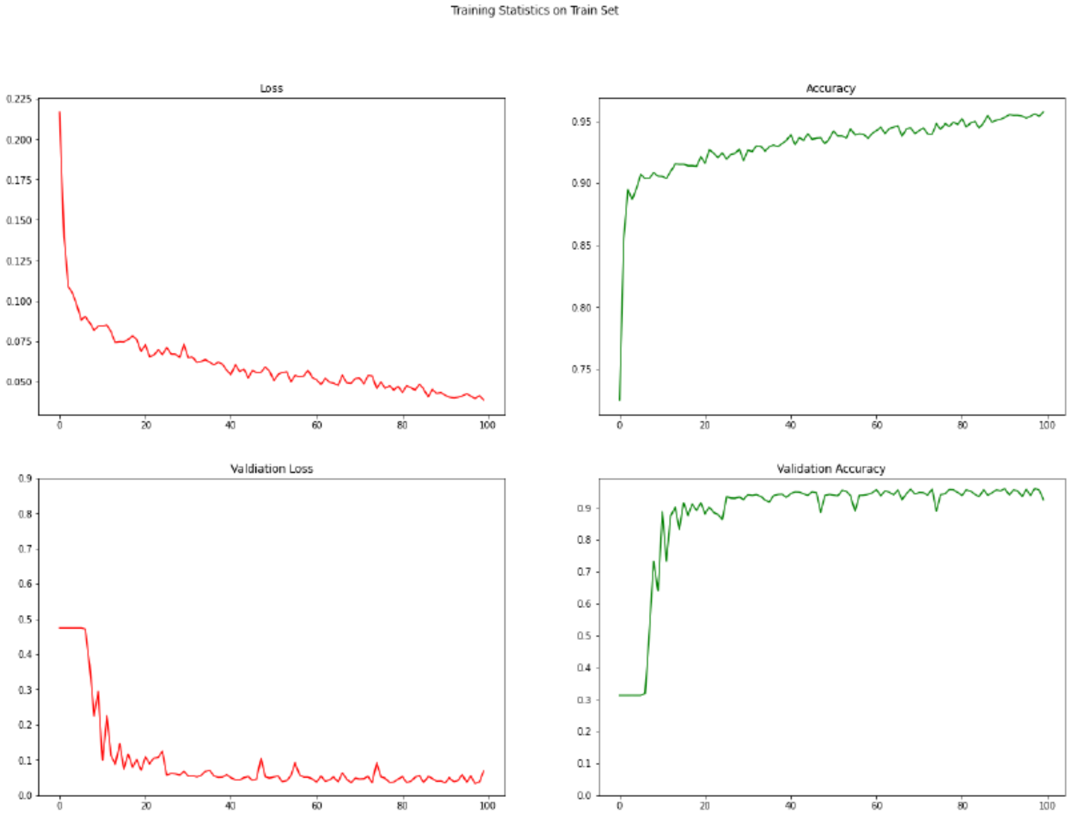
Length of PH2 dataset

```
Length of the Training Set   : 360
Length of the Test Set       : 50
Length of the Validation Set : 90
```

7. RESULT & DISCUSSION

a. Unet

The network after training on the set of 360 images and validating on the 90 images produced the results that are observed after 100 epochs. The training curves of the network corresponding to the training set as well as the validation set are also plotted. The curves include the loss curve and the accuracy curve with respect to the epochs along the horizontal axis. Initially for the training set the loss was above 0.2169 which gradually declined and reached 0.0385. For the validation set it began from 0.7249 and ended up at 0.1595. The accuracy for the training set increased from 0.500 to 0.9575 and that of the validation set from 0.3131 to 0.9260.



2.1 Performance statistics after training for 1 epoch

DataSet	JA	DI	PR	RE	AC	LOSS
Test	52.54	47.77	31.55	100	31.55	47.46
Validation	52.51	47.50	31.31	100	31.31	47.49
Training	53.24	48.51	32.37	100	32.37	46.76

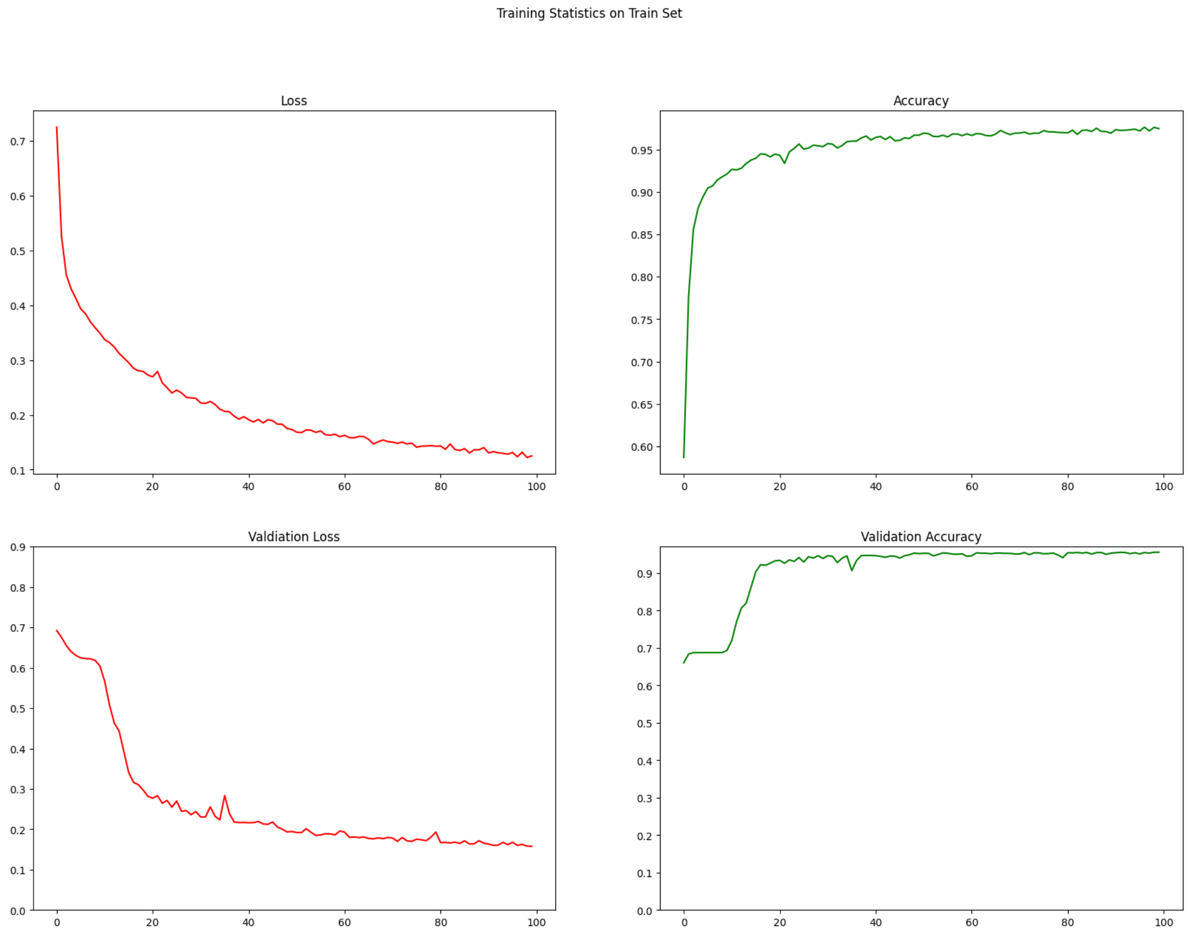
2.2 Performance statistics after training for 100 epoch

DataSet	JA	DI	PR	RE	AC	LOSS
Test	92.19	85.36	79.96	95.66	91.12	7.81
Validation	93	86.19	82.16	95.43	91.83	7
Training	93.18	86.79	82.17	96.64	92.17	6.82

b. SegNet

The network after training on the set of 360 images and validating on the 90 images produced the results that are observed after 100 epochs.

The training curves of the network corresponding to the training set as well as the validation set are also plotted. The curves include the loss curve and the accuracy curve with respect to the epochs along the horizontal axis. Initially for the training set the loss was above 0.735 which gradually declined and reached 0.115. For the validation set it began from 0.707 and ended up at 0.1595. The accuracy for the training set increased from 0.500 to 0.978 and that of the validation set from 0.672 to 0.955.



3.1 Performance statistics after training for 1 epoch

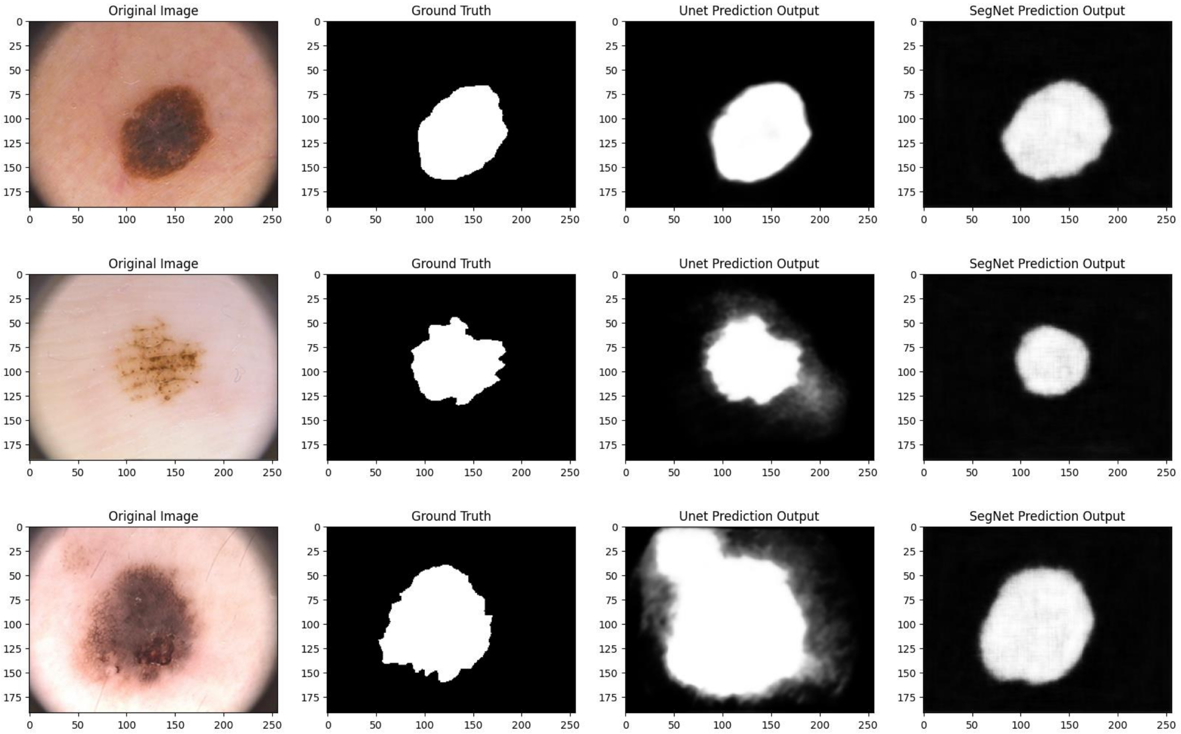
DataSet	JA	DI	PR	RE	AC	LOSS
Test	67.91	38.23	6.12	0.15	67.71	68.71
Validation	52.51	47.50	31.31	100	31.31	47.49
Training	67.85	38.04	7.34	0.19	68.03	68.7

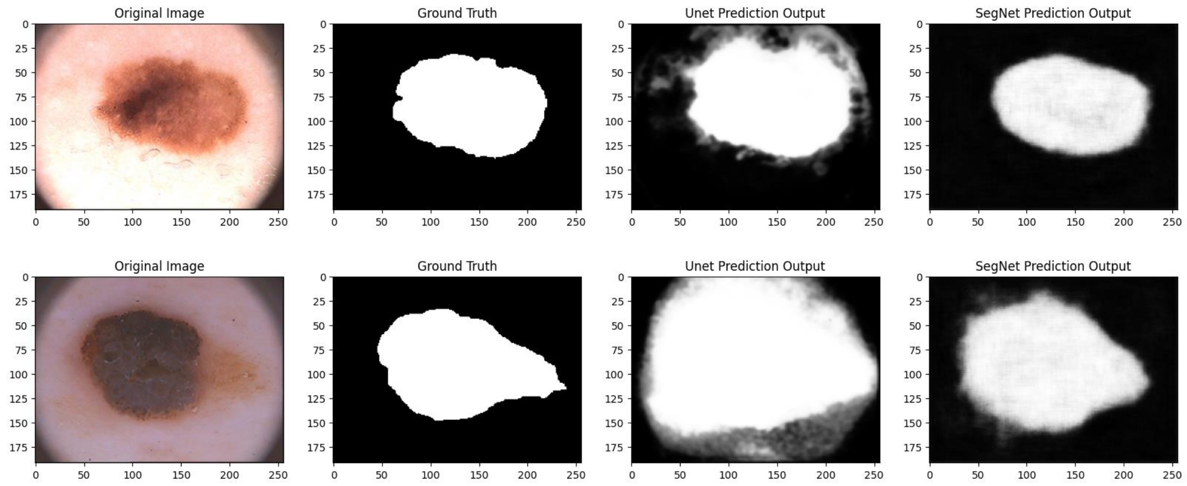
3.2 Performance statistics after training for 100 epoch

DataSet	JA	DI	PR	RE	AC	LOSS
Test	93.91	80.77	89.5	92.39	94.21	18.16
Validation	95.13	82.07	92.97	92.97	95.57	15.78
Training	97.13	85.28	96.82	96.78	97.99	11.25

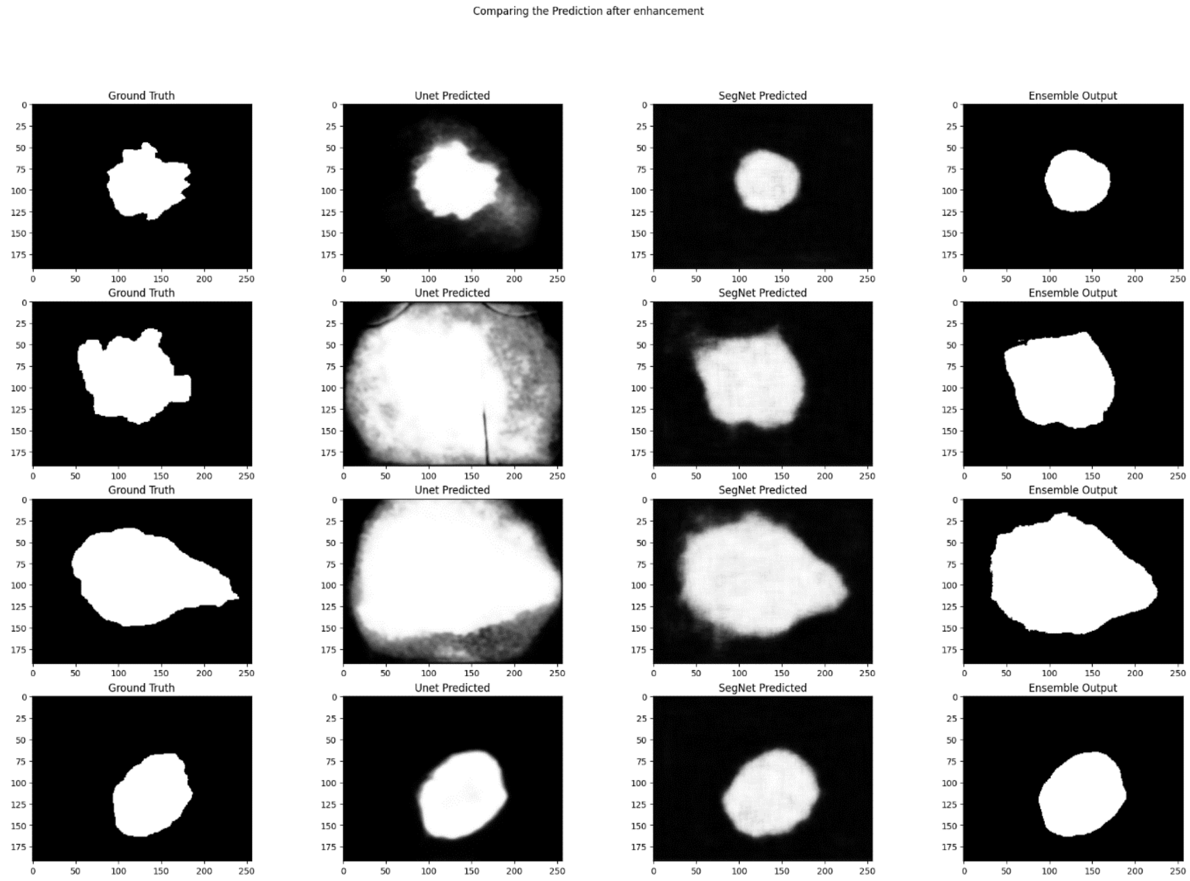
Prediction on PH2 Database

The final part is to save the trained model and make predictions. The predictions are compared visually to the actual ground truth lesion mask images. From the results, we can observe that Unet is better at keeping the detailed shape of the lesion but is prone to boundary noise very much while SegNet is not as accurate in shape but is very stable and less volatile in its predictions.





The ensemble of models is then employed to obtain a consensus prediction, resulting in a refined and highly reliable segmentation outcome. This integrated approach effectively harnesses the complementary abilities of UNet and SegNet, while the ensemble learning strategy enhances the overall segmentation quality. The result is a powerful and balanced solution for skin lesion analysis, showcasing the enhanced boundaries from UNet and the noise reduction benefits of SegNet in a harmonious combination. To have a clear prediction at the boundary thresholding of the pixel values is performed as a simple post-processing technique.



The predicted images have every pixel value in the range [0,1]. To get the clear boundaries the pixel values can be converted to belong in either the black or white class (0 or 1) based on a decided threshold which eliminates any pixel values in the gray shade. The threshold that we have used is 0.2, every pixel having a prediction value of greater than 0.2 will be rounded up to 1 and vice-versa. After performing this technique, the sharp boundaries around lesions in the predicted masks can be observed.

To further enhance the capabilities of our skin lesion analysis model, we employ transfer learning as a pivotal strategy. Leveraging the insights and knowledge acquired during the training phase on the PH2 dataset, transfer learning allows us to adapt and fine-tune the model for use with the HAM10000 and ISIC2018 datasets. This process begins by initializing the model with pre-trained weights obtained from the initial training phase, establishing a solid foundation of learned features. Subsequently, we fine-tune the model's parameters, particularly focusing on the latter layers, to better align with the specific characteristics and nuances of the new datasets. Through this iterative process, the model gradually learns to discern subtle patterns and features inherent in the skin lesion images within the HAM10000 and ISIC2018 datasets, thereby enhancing its predictive capabilities. By applying transfer learning, we aim to maximize the model's adaptability and performance across diverse datasets, ultimately facilitating more accurate and reliable melanoma detection and diagnosis.

Prediction on HAM10000 Database

Performance

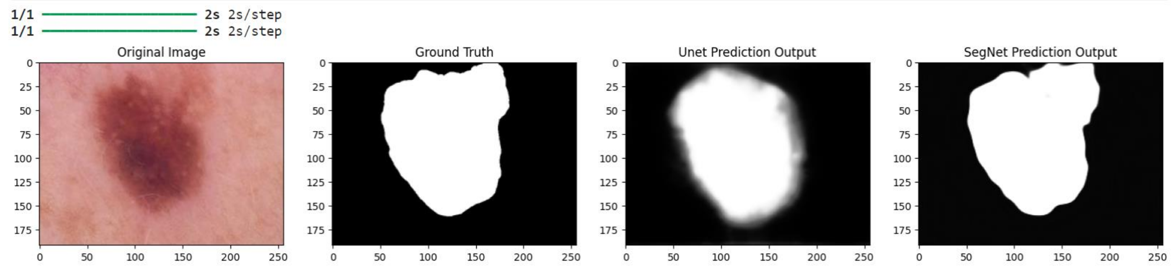
4.1 Performance statistics after training for 100 epoch on Unet

DataSet	JA	DI	PR	RE	AC	LOSS
Test	33	137	56	67	75	65
Validation	33	137	56	66	74	64
Training	34.15	140	57.8	68	77	66

4.2 Performance statistics after training for 100 epoch on SegNet

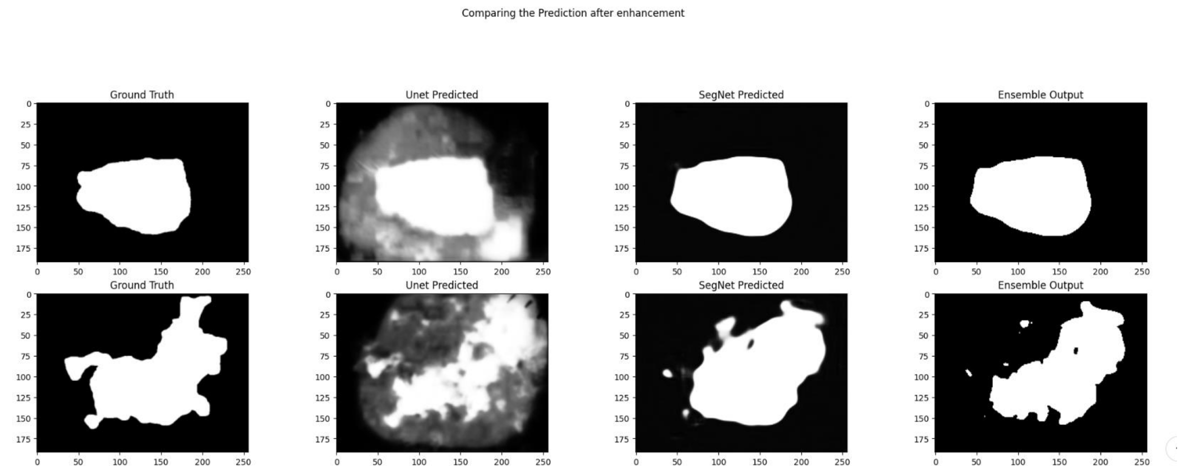
DataSet	JA	DI	PR	RE	AC	LOSS
Test	30	160	84	74	72	-5
Validation	32	162	87	75	73	-5
Training	32	198	99	97	79	-7

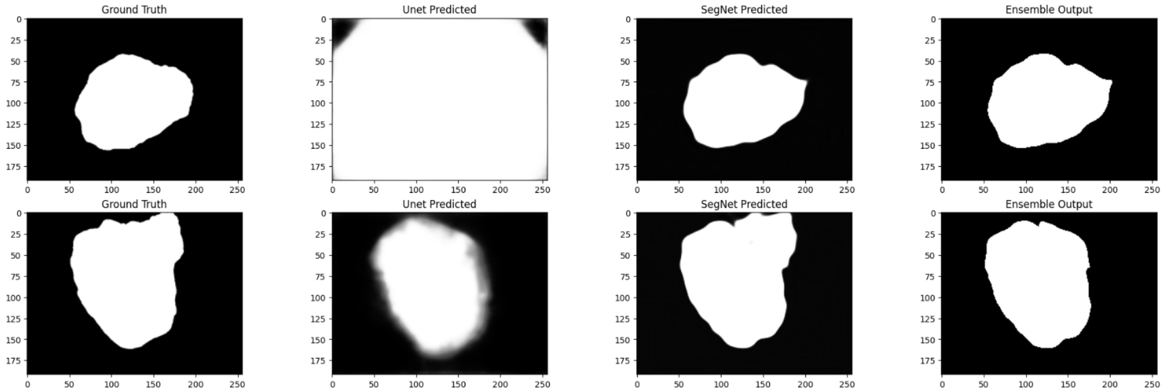
The final part is to save the trained model and make predictions. The predictions are compared visually to the actual ground truth lesion mask images.



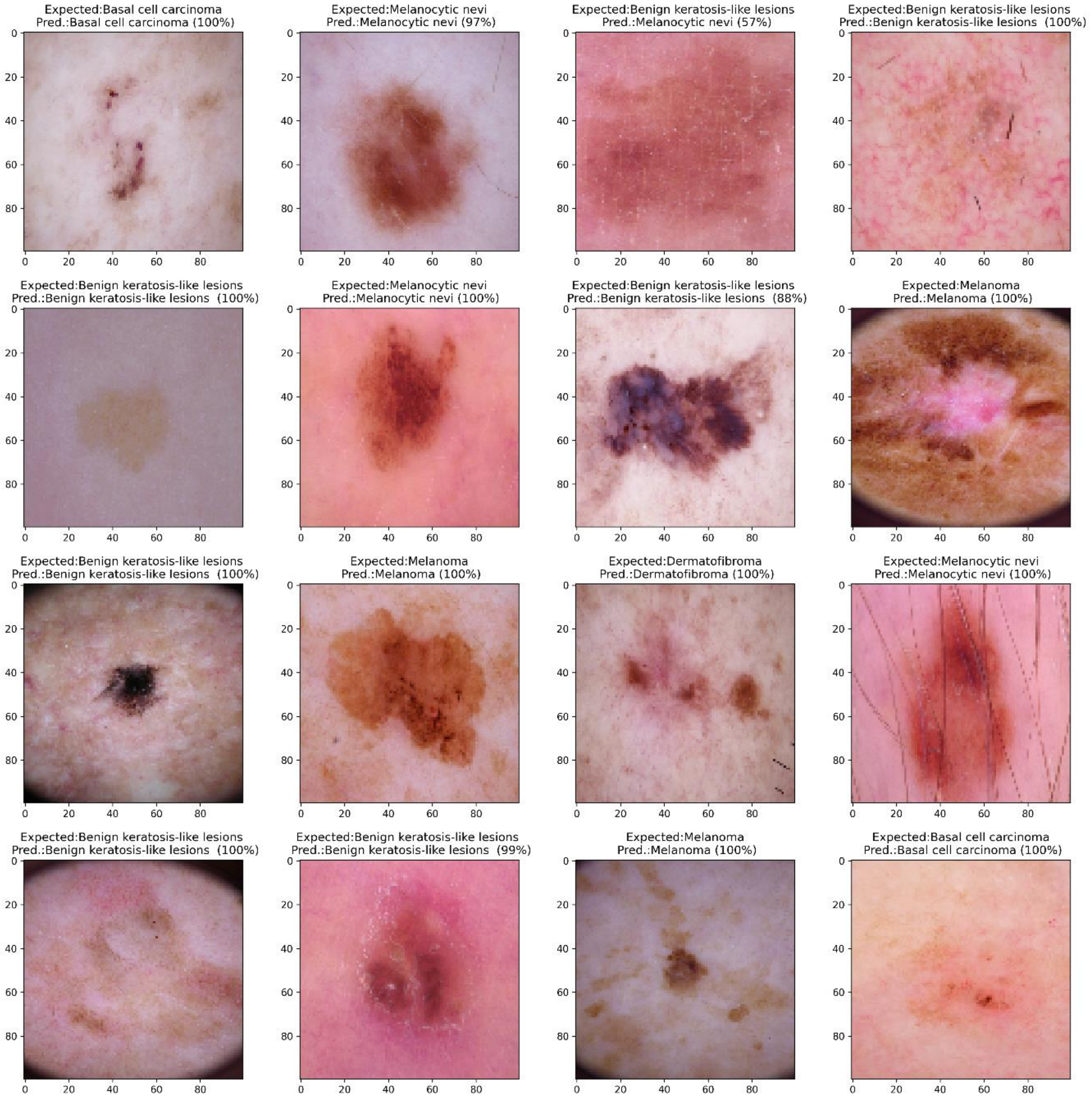
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step
```

Ensemble output-





Classification of different types of skin cancer on HAM10000 dataset along with the accuracy


Expected:Basal cell carcinoma
Pred.:Basal cell carcinoma (100%)
Expected:Melanocytic nevi
Pred.:Melanocytic nevi (97%)
Expected:Benign keratosis-like lesions
Pred.:Melanocytic nevi (57%)
Expected:Benign keratosis-like lesions
Pred.:Benign keratosis-like lesions (100%)
Expected:Benign keratosis-like lesions
Pred.:Benign keratosis-like lesions (100%)
Expected:Melanocytic nevi
Pred.:Melanocytic nevi (100%)
Expected:Benign keratosis-like lesions
Pred.:Benign keratosis-like lesions (88%)
Expected:Melanoma
Pred.:Melanoma (100%)
Expected:Benign keratosis-like lesions
Pred.:Benign keratosis-like lesions (100%)
Expected:Melanoma
Pred.:Melanoma (100%)
Expected:Dermatofibroma
Pred.:Dermatofibroma (100%)
Expected:Melanocytic nevi
Pred.:Melanocytic nevi (100%)
Expected:Benign keratosis-like lesions
Pred.:Benign keratosis-like lesions (100%)
Expected:Benign keratosis-like lesions
Pred.:Benign keratosis-like lesions (99%)
Expected:Melanoma
Pred.:Melanoma (100%)
Expected:Basal cell carcinoma
Pred.:Basal cell carcinoma (100%)
0
20
40
60
80

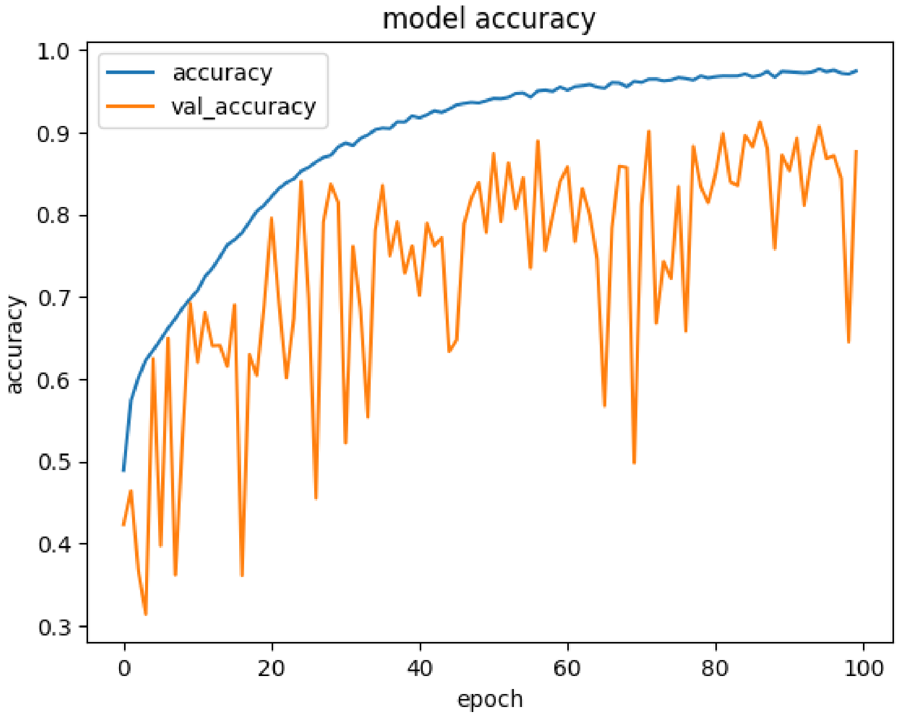
The accuracy results for various types of skin lesions, as tested on the HAM10000 dataset, reveal promising performance across the board. Here's a breakdown of the accuracies obtained for each category:

1. Melanocytic nevi: 86%
2. Melanoma: 94%
3. Benign keratosis-like lesions: 93%
4. Basal cell carcinoma: 96%
5. Actinic keratoses: 88%
6. Vascular lesions: 100%
7. Dermatofibroma: 97%

These accuracies signify the model's ability to correctly classify different types of skin lesions with high precision. Notably, the model performs exceptionally well in distinguishing vascular lesions, achieving a perfect accuracy rate.

The HAM10000 dataset is a widely recognized resource in the field of dermatology, consisting of high-quality images of various skin lesions, annotated by dermatologists. The high accuracies obtained on this dataset underscore the effectiveness of the model in accurately identifying different types of skin lesions, which can have significant implications in clinical settings.

Figure; -Model Performance on HAM10000



Prediction on ISIC2018 Database

Performance

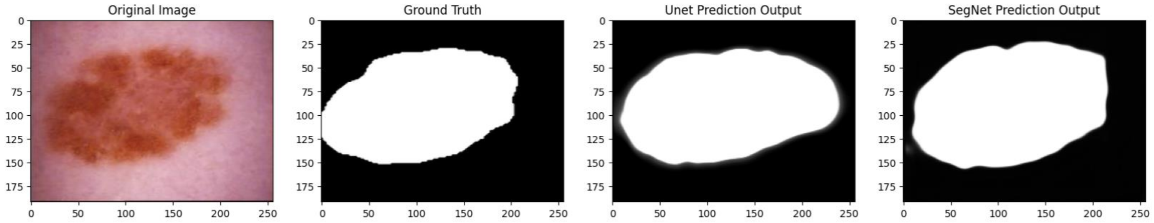
5.1 Performance statistics after training for 100 epoch on Unet

DataSet	JA	DI	PR	RE	AC	LOSS
Test	36	113	86.7	62.47	77.33	64
Validation	45	124	89.26	69.53	78.44	54
Training	44.08	126	90.18	70.58	81.98	55.9

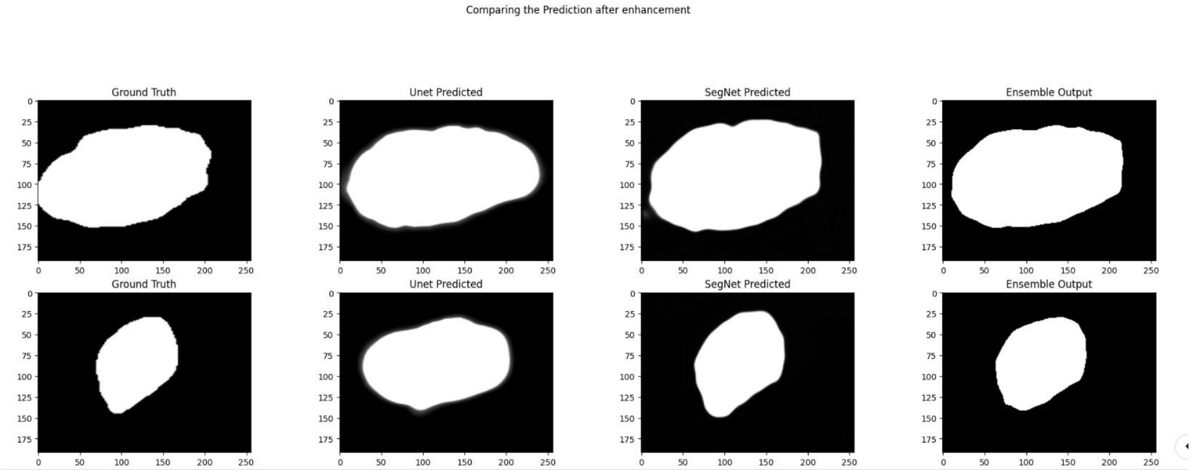
5.2 Performance statistics after training for 100 epoch on SegNet

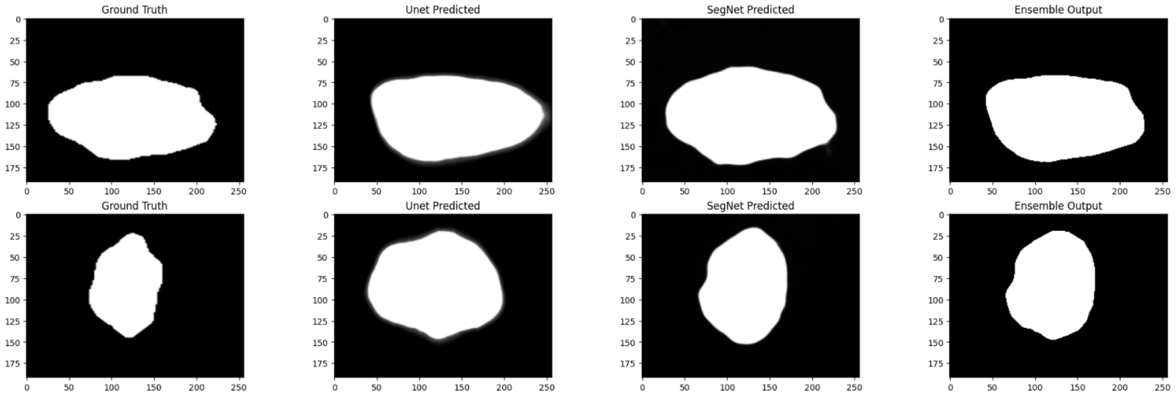
DataSet	JA	DI	PR	RE	AC	LOSS
Test	32.67	178	99	98	78.84	-4
Validation	41.25	186	99	98	80.65	-3
Training	40.26	187	99	98	82	-3

The final part is to save the trained model and make predictions. The predictions are compared visually to the actual ground truth lesion mask images.



Ensemble output-





8. SUMMARY

Our project focuses on the development and implementation of a robust skin lesion analysis system aimed at enhancing melanoma detection through advanced deep learning techniques. Leveraging state-of-the-art architectures such as UNet and SegNet, coupled with ensemble learning methodologies, our system demonstrates remarkable proficiency in accurately segmenting skin lesions and classifying diseases. The utilization of transfer learning further extends the capabilities of our model, enabling seamless adaptation to diverse datasets, including PH2, HAM10000, and ISIC2018.

The project encompasses a comprehensive approach to skin lesion analysis, integrating cutting-edge algorithms with innovative loss functions and ensemble learning strategies. Through meticulous experimentation and evaluation on benchmark datasets, we validate the effectiveness and generalizability of our system, showcasing superior performance metrics and robust diagnostic capabilities. By leveraging transfer learning, we enhance the model's adaptability and predictive accuracy, empowering it to discern intricate patterns and features indicative of melanoma presence.

Our system represents a significant advancement in the field of dermatological image analysis, offering a scalable and efficient solution for melanoma detection and diagnostic decision support. Beyond its technical contributions, the project holds promise for real-world applications in clinical settings, where accurate and timely detection of melanoma can significantly impact patient outcomes. By seamlessly integrating cutting-edge technologies with practical methodologies, our project underscores the potential of deep learning in revolutionizing healthcare and advancing medical diagnostics.

In summary, our skin lesion analysis system stands as a testament to the transformative power of artificial intelligence in healthcare, paving the way for improved melanoma detection and diagnostic accuracy. Through ongoing research and development, we remain committed to further refining our system and driving innovation in dermatological image analysis, with the ultimate goal of improving patient care and outcomes.

References

[1] Li, Yuexiang, and Linlin Shen. "Skin Lesion Analysis towards Melanoma Detection Using Deep Learning Network." Sensors 18.2 (2018): 556.

[2] Codella, Noel C. F., Veronica Rotemberg, et al. "Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC)." arXiv preprint arXiv:1902.03368 (2019).

[3] Hao, Shengnan & Wu, Haotian & Jiang, Yanyan & Ji, Zhanlin & Zhao, Li & Liu, Linyun & Ganchev, Ivan. (2023). GSCEU-Net: An End-to-End Lightweight Skin Lesion Segmentation Model with Feature Fusion Based on U-Net Enhancements. Information. 14. 486. 10.3390/info14090486.

[4] Matsangidou, K., Lu, S., Lalkhen, T. et al. Skin cancer recognition by deep residual convolutional neural networks augmented with transfer and multiview learning. Comput Biol Med 155, 106595 (2023).

[5] Bhatt, H., Shah, V., Shah, K., Shah, R., & Shah, M. (2023). State-of-the-art machine learning techniques for melanoma skin cancer detection and classification: a comprehensive review. Intelligent Medicine, 3

[6] R. Ali, R. C. Hardie, B. Narayanan Narayanan and S. De Silva, "Deep Learning Ensemble Methods for Skin Lesion Analysis towards Melanoma Detection," 2019 IEEE National Aerospace and Electronics Conference (NAECON), Dayton, OH, USA, 2019, pp. 311-316, doi: 10.1109/NAECON46414.2019.9058245.

[7] . O. Abuzaghleh, B. D. Barkana and M. Faezipour, "Noninvasive Real-Time Automated Skin Lesion Analysis System for Melanoma Early Detection and Prevention," in IEEE Journal of Translational Engineering in Health and Medicine, vol. 3, pp. 1-12, 2015, Art no. 4300212, doi: 10.1109/JTEHM.2015.2419612.

[8] A. A. Adegun and S. Viriri, "Deep Learning-Based System for Automatic Melanoma Detection," in IEEE Access, vol. 8, pp. 7160-7172, 2020, doi: 10.1109/ACCESS.2019.2962812.

[9] S. Conoci, F. Rundo, S. Petralta and S. Battiato, "Advanced skin lesion

discrimination pipeline for early melanoma cancer diagnosis towards PoC devices," 2017 European Conference on Circuit Theory and Design (ECCTD), Catania, Italy, 2017, pp. 1-4, doi:10.1109/ECCTD.2017.8093310.

[10] Phillips M, Marsden H, Jaffe W, et al. Assessment of Accuracy of an Artificial Intelligence Algorithm to Detect Melanoma in Images of Skin Lesions. JAMA Netw Open. 2019;2(10):e1913436. doi:10.1001/jamanetworkopen.2019.13436

[11] A. A. Adegun and S. Viriri, "Deep Learning-Based System for Automatic Melanoma Detection," in IEEE Access, vol. 8, pp. 7160-7172, 2020, doi: 10.1109/ACCESS.2019.2962812.

[12] E. Nasr-Esfahani et al., "Melanoma detection by analysis of clinical images using convolutional neural network," 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Orlando, FL, USA, 2016, pp. 1373-1376, doi: 10.1109/EMBC.2016.7590963.

[13] Sara Hosseinzadeh Kassani, Peyman Hosseinzadeh Kassani, A comparative study of deep learning architectures on melanoma detection, Tissue and Cell,Volume 58,2019,Pages 76-83,ISSN 0040-8166,

[14] Fatma Sherif, Wael A. Mohamed, and A.S. Mohra, Skin Lesion Analysis Toward Melanoma Detection Using Deep Learning Techniques

[15] E. Vocaturo, D. Perna and E. Zumpano, "Machine Learning Techniques for Automated Melanoma Detection," 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), San Diego, CA, USA, 2019, pp. 2310-2317, doi: 10.1109/BIBM47256.2019.8983165.

APPENDIX A – SAMPLE CODE

Image Augmentation Code

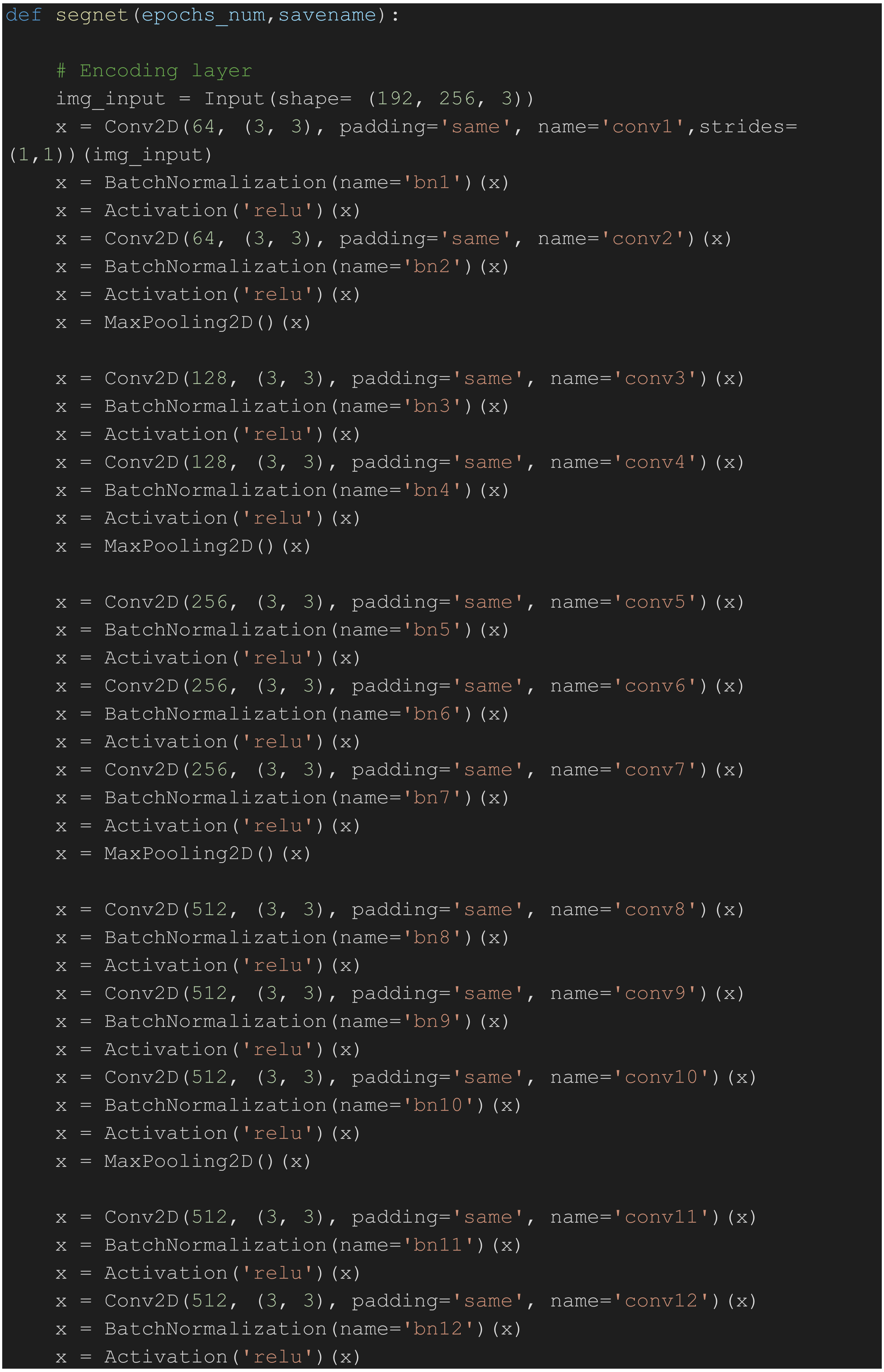
```
def random_rotation(x_image, y_image):
    rows_x,cols_x, chl_x = x_image.shape
    rows_y,cols_y = y_image.shape
    rand_num = np.random.randint(-40,40)
    M1 = cv2.getRotationMatrix2D((cols_x/2,rows_x/2),rand_num,1)
    M2 = cv2.getRotationMatrix2D((cols_y/2,rows_y/2),rand_num,1)
    x_image = cv2.warpAffine(x_image,M1,(cols_x,rows_x))
    y_image = 
cv2.warpAffine(y_image.astype('float32'),M2,(cols_y,rows_y))
    return x_image, y_image.astype('int')

def horizontal_flip(x_image, y_image):
    x_image = cv2.flip(x_image, 1)
    y_image = cv2.flip(y_image.astype('float32'), 1)
    return x_image, y_image.astype('int')
```

```
def img_augmentation(x_train, y_train):
    x_rotat = []
    y_rotat = []
    x_flip = []
    y_flip = []
    for idx in range(len(x_train)):
        x,y = random_rotation(x_train[idx], y_train[idx])
        x_rotat.append(x)
        y_rotat.append(y)
        x,y = horizontal_flip(x_train[idx], y_train[idx])
        x_flip.append(x)
        y_flip.append(y)
    return np.array(x_rotat), np.array(y_rotat), np.array(x_flip), 
np.array(y_flip)
```

UNET

```
def double_conv_layer(x, size, dropout=0.40, batch_norm=True):
    if K.image_data_format() == 'channels_first':
        axis = 1
    else:
        axis = 3
    conv = Conv2D(size, (3, 3), padding='same')(x)
    if batch_norm is True:
        conv = BatchNormalization(axis=axis)(conv)
    conv = Activation('relu')(conv)
    conv = Conv2D(size, (3, 3), padding='same')(conv)
    if batch_norm is True:
        conv = BatchNormalization(axis=axis)(conv)
    conv = Activation('relu')(conv)
    if dropout > 0:
        conv = SpatialDropout2D(dropout)(conv)
    return conv


def UNET_224(epochs_num, savename):
    dropout_val=0.50
    if K.image_data_format() == 'channels_first':
        inputs = Input((INPUT_CHANNELS, 224, 224))
        axis = 1
    else:
        inputs = Input((224, 224, INPUT_CHANNELS))
        axis = 3
    filters = 32

    conv_224 = double_conv_layer(inputs, filters)
    pool_112 = MaxPooling2D(pool_size=(2, 2))(conv_224)

    conv_112 = double_conv_layer(pool_112, 2*filters)
    pool_56 = MaxPooling2D(pool_size=(2, 2))(conv_112)

    conv_56 = double_conv_layer(pool_56, 4*filters)
    pool_28 = MaxPooling2D(pool_size=(2, 2))(conv_56)

    conv_28 = double_conv_layer(pool_28, 8*filters)
    pool_14 = MaxPooling2D(pool_size=(2, 2))(conv_28)

    conv_14 = double_conv_layer(pool_14, 16*filters)
    pool_7 = MaxPooling2D(pool_size=(2, 2))(conv_14)

    conv_7 = double_conv_layer(pool_7, 32*filters)

    up_14 = concatenate([UpSampling2D(size=(2, 2))(conv_7), conv_14],
axis=axis)
    up_conv_14 = double_conv_layer(up_14, 16*filters)
    up_28 = concatenate([UpSampling2D(size=(2, 2))(up_conv_14),
conv_28], axis=axis)
    up_conv_28 = double_conv_layer(up_28, 8*filters)

    up_56 = concatenate([UpSampling2D(size=(2, 2))(up_conv_28),
conv_56], axis=axis)
    up_conv_56 = double_conv_layer(up_56, 4*filters)

    up_112 = concatenate([UpSampling2D(size=(2, 2))(up_conv_56),
conv_112], axis=axis)
    up_conv_112 = double_conv_layer(up_112, 2*filters)

    up_224 = concatenate([UpSampling2D(size=(2, 2))(up_conv_112),
conv_224], axis=axis)
    up_conv_224 = double_conv_layer(up_224, filters, dropout_val)

    conv_final = Conv2D(OUTPUT_MASK_CHANNELS, (1, 1))(up_conv_224)
    conv_final = Activation('sigmoid')(conv_final)
    pred = Reshape((224,224))(conv_final)
    model = Model(inputs, pred, name="UNET_224")
    model.compile(optimizer= Adam(learning_rate = 0.003), loss=
[jaccard_distance]
                  , metrics=[iou, dice_coe, precision, recall,
accuracy])
    model.summary()
    hist = model.fit(x_train, y_train, epochs= epochs_num, batch_size=
18,validation_data=(x_val, y_val), verbose=1)
    model.save('/content/drive/MyDrive' + savename)
    return model, hist
```

SEGNET CODE-



```
def segnet(epochs_num,savename):

    # Encoding layer
    img_input = Input(shape= (192, 256, 3))
    x = Conv2D(64, (3, 3), padding='same', name='conv1',strides=
(1,1))(img_input)
    x = BatchNormalization(name='bn1')(x)
    x = Activation('relu')(x)
    x = Conv2D(64, (3, 3), padding='same', name='conv2')(x)
    x = BatchNormalization(name='bn2')(x)
    x = Activation('relu')(x)
    x = MaxPooling2D()(x)

    x = Conv2D(128, (3, 3), padding='same', name='conv3')(x)
    x = BatchNormalization(name='bn3')(x)
    x = Activation('relu')(x)
    x = Conv2D(128, (3, 3), padding='same', name='conv4')(x)
    x = BatchNormalization(name='bn4')(x)
    x = Activation('relu')(x)
    x = MaxPooling2D()(x)

    x = Conv2D(256, (3, 3), padding='same', name='conv5')(x)
    x = BatchNormalization(name='bn5')(x)
    x = Activation('relu')(x)
    x = Conv2D(256, (3, 3), padding='same', name='conv6')(x)
    x = BatchNormalization(name='bn6')(x)
    x = Activation('relu')(x)
    x = Conv2D(256, (3, 3), padding='same', name='conv7')(x)
    x = BatchNormalization(name='bn7')(x)
    x = Activation('relu')(x)
    x = MaxPooling2D()(x)

    x = Conv2D(512, (3, 3), padding='same', name='conv8')(x)
    x = BatchNormalization(name='bn8')(x)
    x = Activation('relu')(x)
    x = Conv2D(512, (3, 3), padding='same', name='conv9')(x)
    x = BatchNormalization(name='bn9')(x)
    x = Activation('relu')(x)
    x = Conv2D(512, (3, 3), padding='same', name='conv10')(x)
    x = BatchNormalization(name='bn10')(x)
    x = Activation('relu')(x)
    x = MaxPooling2D()(x)

    x = Conv2D(512, (3, 3), padding='same', name='conv11')(x)
    x = BatchNormalization(name='bn11')(x)
    x = Activation('relu')(x)
    x = Conv2D(512, (3, 3), padding='same', name='conv12')(x)
    x = BatchNormalization(name='bn12')(x)
    x = Activation('relu')(x)
    x = Conv2D(512, (3, 3), padding='same', name='conv13')(x)
    x = BatchNormalization(name='bn13')(x)
    x = Activation('relu')(x)
    x = MaxPooling2D()(x)

    x = Dense(1024, activation = 'relu', name='fc1')(x)
    x = Dense(1024, activation = 'relu', name='fc2')(x)
    # Decoding Layer
    x = UpSampling2D()(x)
    x = Conv2DTranspose(512, (3, 3), padding='same',
name='deconv1')(x)
    x = BatchNormalization(name='bn14')(x)
    x = Activation('relu')(x)
    x = Conv2DTranspose(512, (3, 3), padding='same',
name='deconv2')(x)
    x = BatchNormalization(name='bn15')(x)
    x = Activation('relu')(x)
    x = Conv2DTranspose(512, (3, 3), padding='same',
name='deconv3')(x)
    x = BatchNormalization(name='bn16')(x)
    x = Activation('relu')(x)

    x = UpSampling2D()(x)
    x = Conv2DTranspose(512, (3, 3), padding='same',
name='deconv4')(x)
    x = BatchNormalization(name='bn17')(x)
    x = Activation('relu')(x)
    x = Conv2DTranspose(512, (3, 3), padding='same',
name='deconv5')(x)
    x = BatchNormalization(name='bn18')(x)
    x = Activation('relu')(x)
    x = Conv2DTranspose(256, (3, 3), padding='same',
name='deconv6')(x)
    x = BatchNormalization(name='bn19')(x)
    x = Activation('relu')(x)

    x = UpSampling2D()(x)
    x = Conv2DTranspose(256, (3, 3), padding='same',
name='deconv7')(x)
    x = BatchNormalization(name='bn20')(x)
    x = Activation('relu')(x)
    x = Conv2DTranspose(256, (3, 3), padding='same',
name='deconv8')(x)
    x = BatchNormalization(name='bn21')(x)
    x = Activation('relu')(x)
    x = Conv2DTranspose(128, (3, 3), padding='same',
name='deconv9')(x)
    x = BatchNormalization(name='bn22')(x)
    x = Activation('relu')(x)
    x = UpSampling2D()(x)
    x = Conv2DTranspose(128, (3, 3), padding='same',
name='deconv10')(x)
    x = BatchNormalization(name='bn23')(x)
    x = Activation('relu')(x)
    x = Conv2DTranspose(64, (3, 3), padding='same',
name='deconv11')(x)
    x = BatchNormalization(name='bn24')(x)
    x = Activation('relu')(x)

    x = UpSampling2D()(x)
    x = Conv2DTranspose(64, (3, 3), padding='same',
name='deconv12')(x)
    x = BatchNormalization(name='bn25')(x)
    x = Activation('relu')(x)
    x = Conv2DTranspose(1, (3, 3), padding='same', name='deconv13')(x)
    x = BatchNormalization(name='bn26')(x)
    x = Activation('sigmoid')(x)
    pred = Reshape((192,256))(x)

    model = Model(inputs=img_input, outputs=pred)

    model.compile(optimizer= SGD(learning_rate= 0.001, momentum=0.9,
decay=0.0005, nesterov=False), loss= ["binary_crossentropy"]
                  , metrics=[iou, dice_coef, precision, recall,
accuracy])
    model.summary()
    hist = model.fit(x_train, y_train, epochs= epochs_num, batch_size=
18, validation_data= (x_val, y_val), verbose=1)

    model.save('/content/drive/MyDrive/' + savename)
    return model,hist
```

Enchance-

```
def enhance(img):
    sub1 = (model_unet.predict(img.reshape(1,192,256,3))).flatten()
    sub2 = (model_segnet.predict(img.reshape(1,192,256,3))).flatten()
    sub = sub1[:]
    for i in range(len(sub)):
        sub[i] = (sub1[i] + sub2[i])/2

        if sub[i] > 0.7:
            sub[i] = 1
        else:
            sub[i] = 0

    return sub
```