A Final Year Project Report

on

OPTIMIZING SKIN CANCER DETECTION USING THE MOBILENET DEEP LEARNING MODEL: A LIGHTWEIGHT AND EFFICIENT APPROACH

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, ANANTHAPURAMU

In Partial Fulfilment of the Requirements for the Award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING (DATA SCIENCE)

Submitted by

B. Praveen	**-**	**(21691A3269)**
C. Revanth	**-**	**(21691A3278)**
G. Sai Haritha	**-**	**(21691A3286)**
D. Siva Kumar Goud	**-**	**(21691A32A2)**

Under the Guidance of

Mr. T. Balaji , M.Tech.,(PhD).,
Assistant Professor

Department of Computer Science & Engineering (Data Science)

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

(UGC – AUTONOMOUS)

(Affiliated to JNTUA, Ananthapuramu)

(Accredited by NBA, Approved by AICTE, New Delhi)

AN ISO 21001 : 2018 Certified Institution

P. B. No: 14, Angallu, Madanapalle – 517325

2021-2025

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

(UGC-AUTONOMOUS INSTITUTION)
Affiliated to JNTUA, Ananthapuramu & Approved by AICTE, New Delhi
NAAC Accredited with A+ Grade, NIRF India Rankings 2024 - Band: 201-300 (Engg.)
NBA Accredited - B.Tech. (CIVIL, CSE, CST, ECE, EEE, MECH), MBA & MCA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (DATA SCIENCE)

BONAFIDE CERTIFICATE

This is to certify that the project work entitled **“OPTIMIZING SKIN CANCER DETECTION USING THE MOBILENET DEEP LEARNING MODEL: A LIGHTWEIGHT AND EFFICIENT APPROACH”** is a bonafide work carried out by

B. Praveen	**-**	**(21691A3269)**
C. Revanth	**-**	**(21691A3278)**
G. Sai Haritha	**-**	**(21691A3286)**
D. Siva Kumar Goud	**-**	**(21691A32A2)**

Submitted in partial fulfillment of the requirements for the award of the degree **Bachelor of Technology** in the stream of **Computer Science & Engineering (Data Science)** in **Madanapalle Institute of Technology and Science, Madanapalle,** affiliated to **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** during the academic year 2024-2025.

Guide

Mr. T. Balaji, M.Tech.,(Ph.D).,
Assistant Professor,
Department of CSE-DS

Head of the Department

Dr. S. Kusuma., Ph. D.,
Assistant Professor and
Head of the Department,
Department of CSE-DS

ACKNOWLEDGEMENT

We sincerely thank the **MANAGEMENT** of **Madanapalle Institute of Technology & Science** for providing excellent infrastructure and lab facilities that helped me to complete this project.

We sincerely thank **Dr. C. Yuvaraj, M.E., Ph.D., Principal** for guiding and providing facilities for the successful completion of our project at **Madanapalle Institute of Technology & Science,** Madanapalle.

We express our deep sense of gratitude to **Dr. S. Kusuma., Ph.D., Assistant Professor and Head of the Department of CSE - Data Science** for her continuous support in making necessary arrangements for the successful completion of the Project.

We express my deep sense of gratitude to **Mr. G. Raj Kumar, M.Tech., (Ph.D), Project Coordinator,** for his valuable guidance and encouragement that helped us to complete this project.

We express our deep gratitude to our guide **Mr. T. Balaji, M.Tech., (Ph.D)., Assistant Professor, Department of CSE - Data Science** for his guidance and encouragement that helped us to complete this project.

We also wish to place on record my gratefulness to other **Faculty of the CSE - Data Science Department** and also to our friends and our parents for their help and cooperation during our project work.

ABSTRACT

Skin cancer is one of the most common cancers globally, and early and correct detection is of key importance to enhance survival and proper treatment. Yet, diagnosis by dermatologists is time-consuming, subjective, and error-prone, particularly in the presence of a large number of cases or uncertain appearance of lesions. The visual similarity of benign and malignant lesions, variations in skin color, lighting, and imaging artifacts complicate the task of classification even more. These obstacles have prompted researchers to consider artificial intelligence (AI)-based methods, especially deep learning, to assist dermatologists with skin cancer diagnosis in a more efficient and reliable manner.

In this project, we introduce automated skin cancer classification using MobileNet, a light-weight and computationally less demanding Convolutional Neural Network (CNN) architecture. MobileNet is geared towards low-resource environments and can be easily used to integrate in mobile apps, edge devices, and point-of-care diagnostic instruments. Our methodology employs the HAM10000 dataset, which is a massive set of 10,000 high-resolution dermoscopic images covering numerous types of pigmented skin lesions. The dataset offers a stable basis for the training of deep learning models as it presents variant examples in distinct skin cancer types.The MobileNet model was trained and tested on this dataset, achieving a staggering training accuracy of 92.86% and a validation accuracy of 98.56%. These findings indicate not only the high accuracy of the model for classifying skin lesions but also its generalization ability and robustness. Furthermore, by virtue of having a small size and fewer parameters, MobileNet drastically minimizes both the cost of computation as well as the inference time for a much denser model like ResNet or Inception and is hence suitable for real-time medical scenarios.

Our results highlight the promise of MobileNet as an affordable, low-cost, yet scalable solution to skin cancer screening. This piece of work lends itself to driving the development of AI-empowered healthcare through the provision of precise diagnostic support to non-specialist healthcare providers and assisting dermatologists in rendering faster, more uniform decisions. In the long run, this incorporation into clinics can result in enhanced early detection, decreased burden on patients, and improved fight against skin cancer.

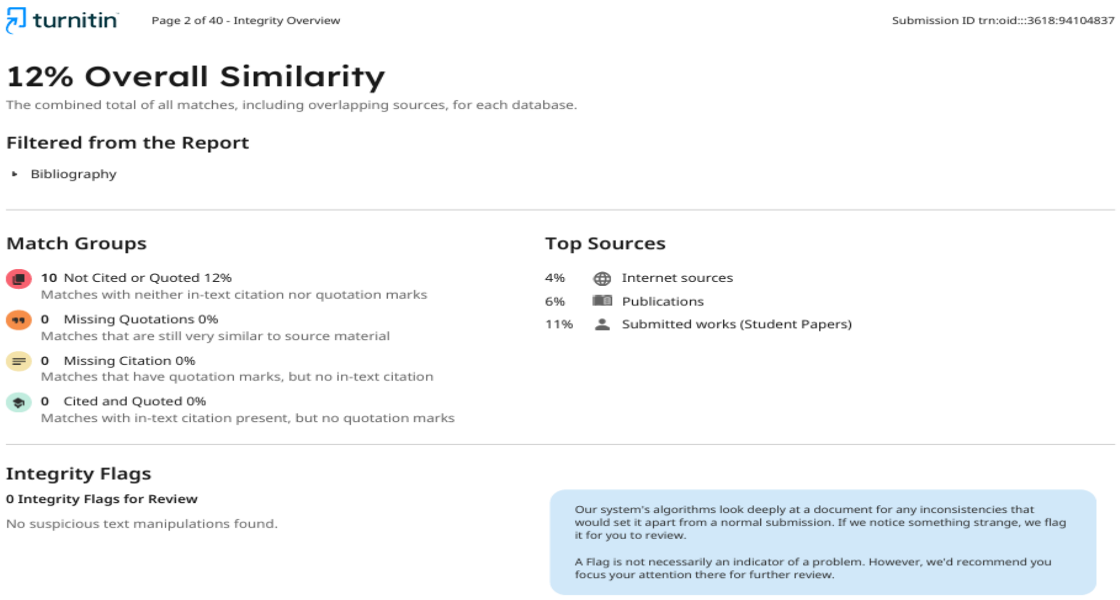
MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

(UGC-AUTONOMOUS INSTITUTION)
Affiliated to JNTUA, Ananthapuramu & Approved by AICTE, New Delhi
NAAC Accredited with A+ Grade, NIRF India Rankings 2024 - Band: 201-300 (Engg.)
NBA Accredited - B.Tech. (CIVIL, CSE, CST, ECE, EEE, MECH), MBA & MCA

RECOGNISED RESEARCH CENTER

Plagiarism Verification Certificate

This is to certify that the B. Tech Project report titled, **"OPTIMIZING SKIN CANCER DETECTION USING THE MOBILENET DEEP LEARNING MODEL: A LIGHTWEIGHT AND EFFICIENT APPROACH"** submitted by **B.Praveen – (21691A3269), C. Revanth – (21691A3278), G. Sai Haritha – (21691A3286), D. Siva Kumar Goud - (21691A32A2)** has been evaluated using **Anti- Plagiarism Software, Turnitin** and based on the analysis report generated by the software, the report similarity index is found to be 12 %.



turnitin Page 2 of 40 - Integrity Overview Submission ID trn:oid:::3618:94104837

12% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

- 10 Not Cited or Quoted 12%
  Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

4%	Internet sources
6%	Publications
11%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

GUIDE

Mr. T. Balaji, M. Tech.,(Ph.D).,
Assistant Professor,
Department of CSE - DS

DECLARATION

We hereby declare that the results embodied in this project **"OPTIMIZING SKIN CANCER DETECTION USING THE MOBILENET DEEP LEARNING MODEL: A LIGHTWEIGHT AND EFFICIENT APPROACH"** by us under the guidance of **, Assistant Professor, Dept. of CSE - Data Science** in partial fulfillment of the award of **Bachelor of Technology** in **Computer Science & Engineering (Data Science)** from **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** and we have not submitted the same to any other University/institute for the award of any other degree.

Date:

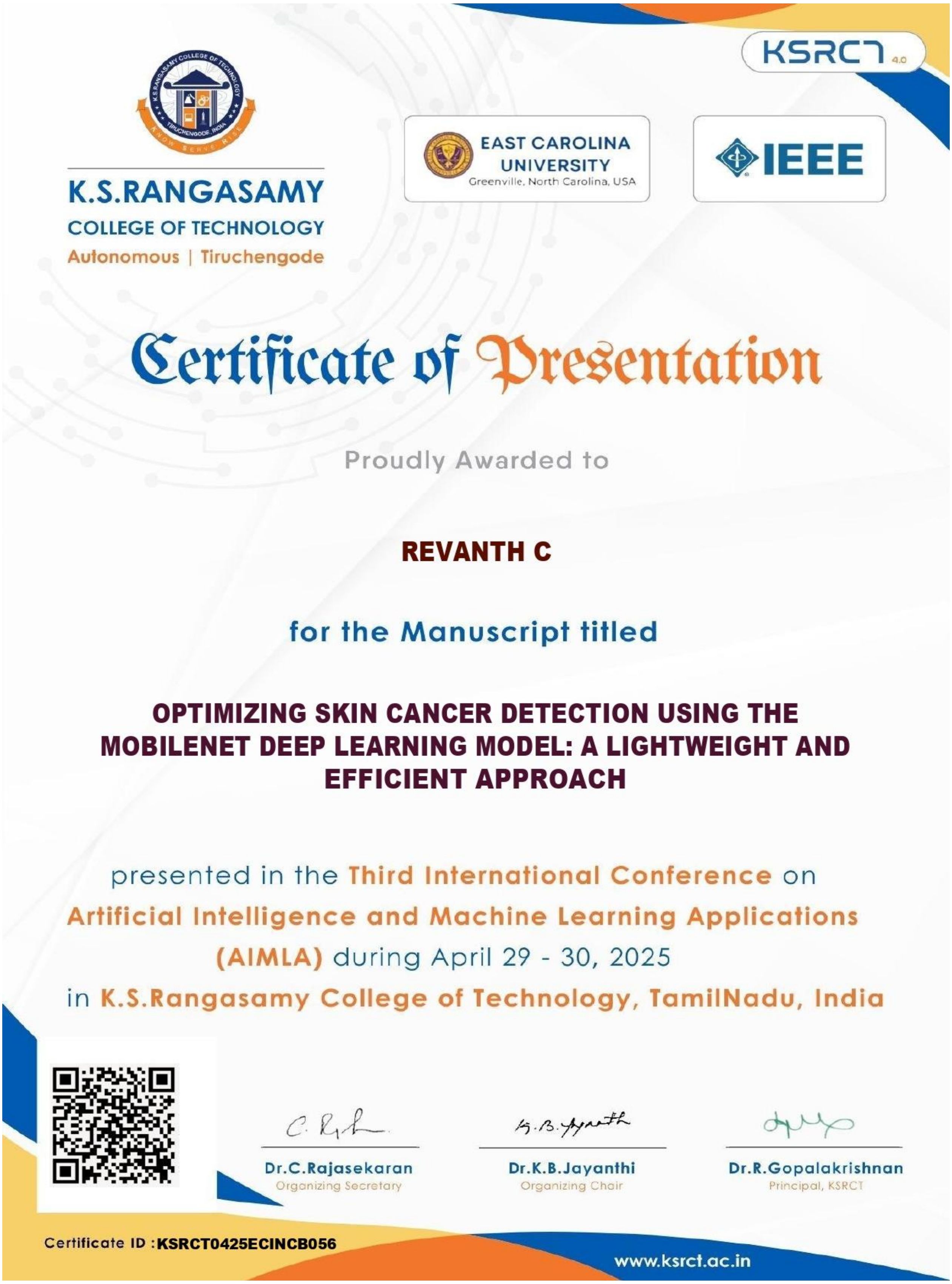
Place:

PROJECT ASSOCIATES

B. Praveen
C. Revanth
G. Sai Haritha
D. Siva Kumar Goud

I certify that the above statement made by the students is correct to the best of my knowledge.

Date: **Guide:**



KSRCT 4.0

K.S.RANGASAMY
COLLEGE OF TECHNOLOGY
Autonomous | Tiruchengode

EAST CAROLINA UNIVERSITY
Greenville, North Carolina, USA

IEEE

Certificate of Presentation

Proudly Awarded to

REVANTH C

for the Manuscript titled

OPTIMIZING SKIN CANCER DETECTION USING THE MOBILENET DEEP LEARNING MODEL: A LIGHTWEIGHT AND EFFICIENT APPROACH

presented in the **Third International Conference** on **Artificial Intelligence and Machine Learning Applications (AIMLA)** during April 29 - 30, 2025 in **K.S.Rangasamy College of Technology, TamilNadu, India**

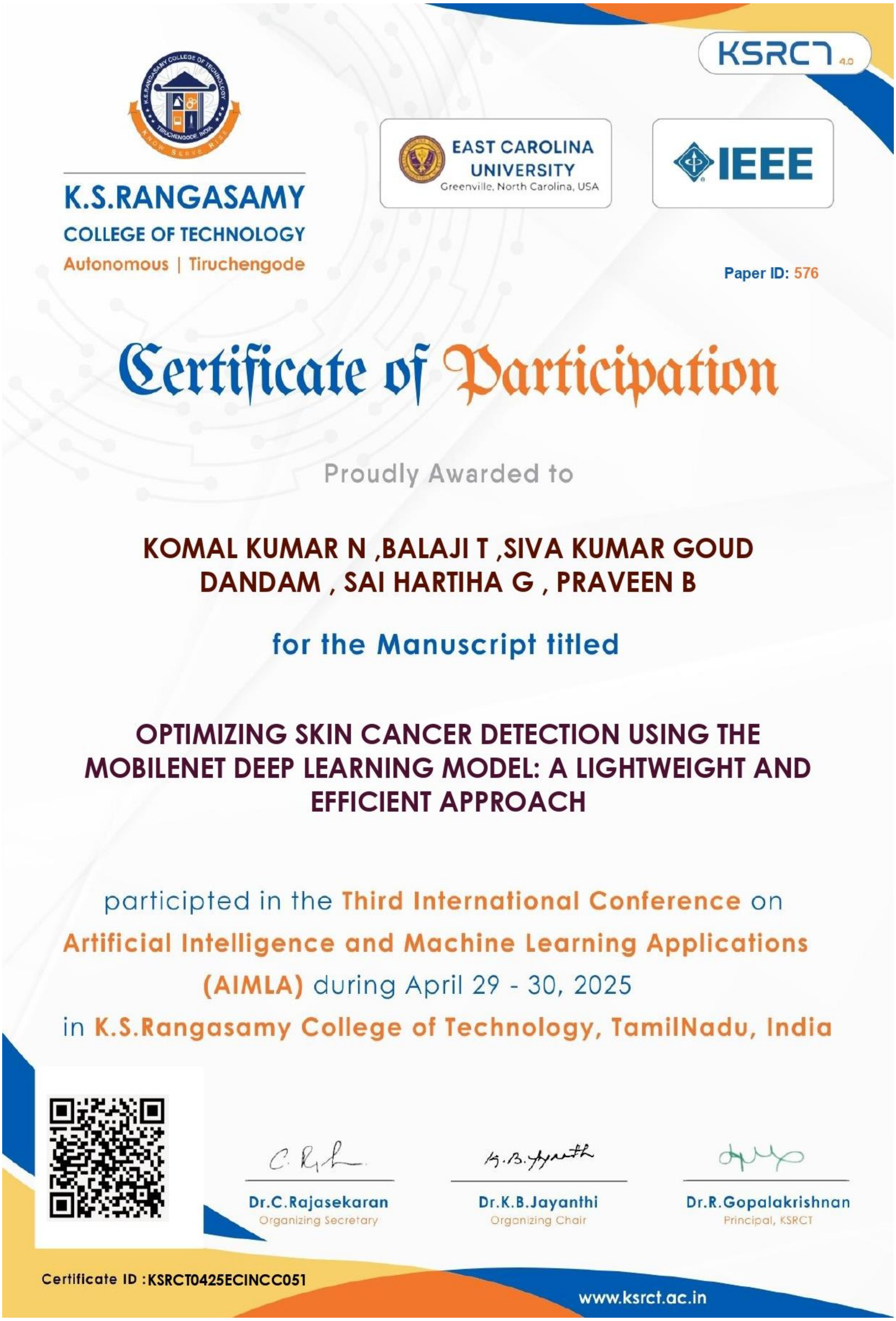
Dr.C.Rajasekaran
Organizing Secretary

Dr.K.B.Jayanthi
Organizing Chair

Dr.R.Gopalakrishnan
Principal, KSRCT

Certificate ID : **KSRCT0425ECINCB056**

www.ksrct.ac.in



K.S.RANGASAMY
COLLEGE OF TECHNOLOGY
Autonomous | Tiruchengode

EAST CAROLINA UNIVERSITY
Greenville, North Carolina, USA

IEEE

KSRCT 4.0

Paper ID: 576

Certificate of Participation

Proudly Awarded to

KOMAL KUMAR N ,BALAJI T ,SIVA KUMAR GOUD DANDAM , SAI HARTIHA G , PRAVEEN B

for the Manuscript titled

OPTIMIZING SKIN CANCER DETECTION USING THE MOBILENET DEEP LEARNING MODEL: A LIGHTWEIGHT AND EFFICIENT APPROACH

participted in the Third International Conference on Artificial Intelligence and Machine Learning Applications (AIMLA) during April 29 - 30, 2025 in K.S.Rangasamy College of Technology, TamilNadu, India

K.B.Jayanth

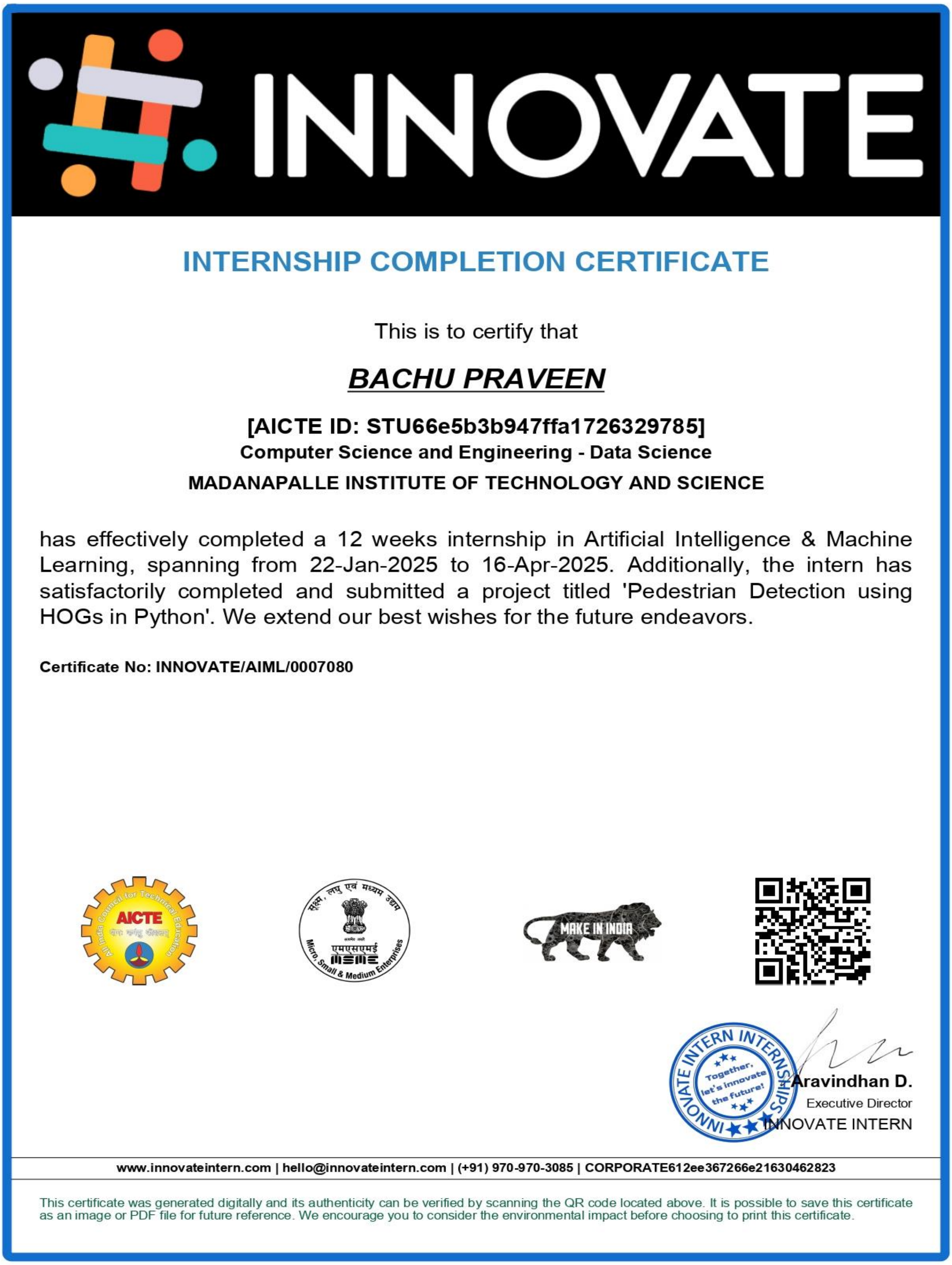
Dr.C.Rajasekaran
Organizing Secretary

Dr.K.B.Jayanthi
Organizing Chair

Dr.R.Gopalakrishnan
Principal, KSRCT

Certificate ID :KSRCT0425ECINCC051

www.ksrct.ac.in



INNOVATE

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that

BACHU PRAVEEN

[AICTE ID: STU66e5b3b947ffa1726329785]

Computer Science and Engineering - Data Science

MADANAPALLE INSTITUTE OF TECHNOLOGY AND SCIENCE

has effectively completed a 12 weeks internship in Artificial Intelligence & Machine Learning, spanning from 22-Jan-2025 to 16-Apr-2025. Additionally, the intern has satisfactorily completed and submitted a project titled 'Pedestrian Detection using HOGs in Python'. We extend our best wishes for the future endeavors.

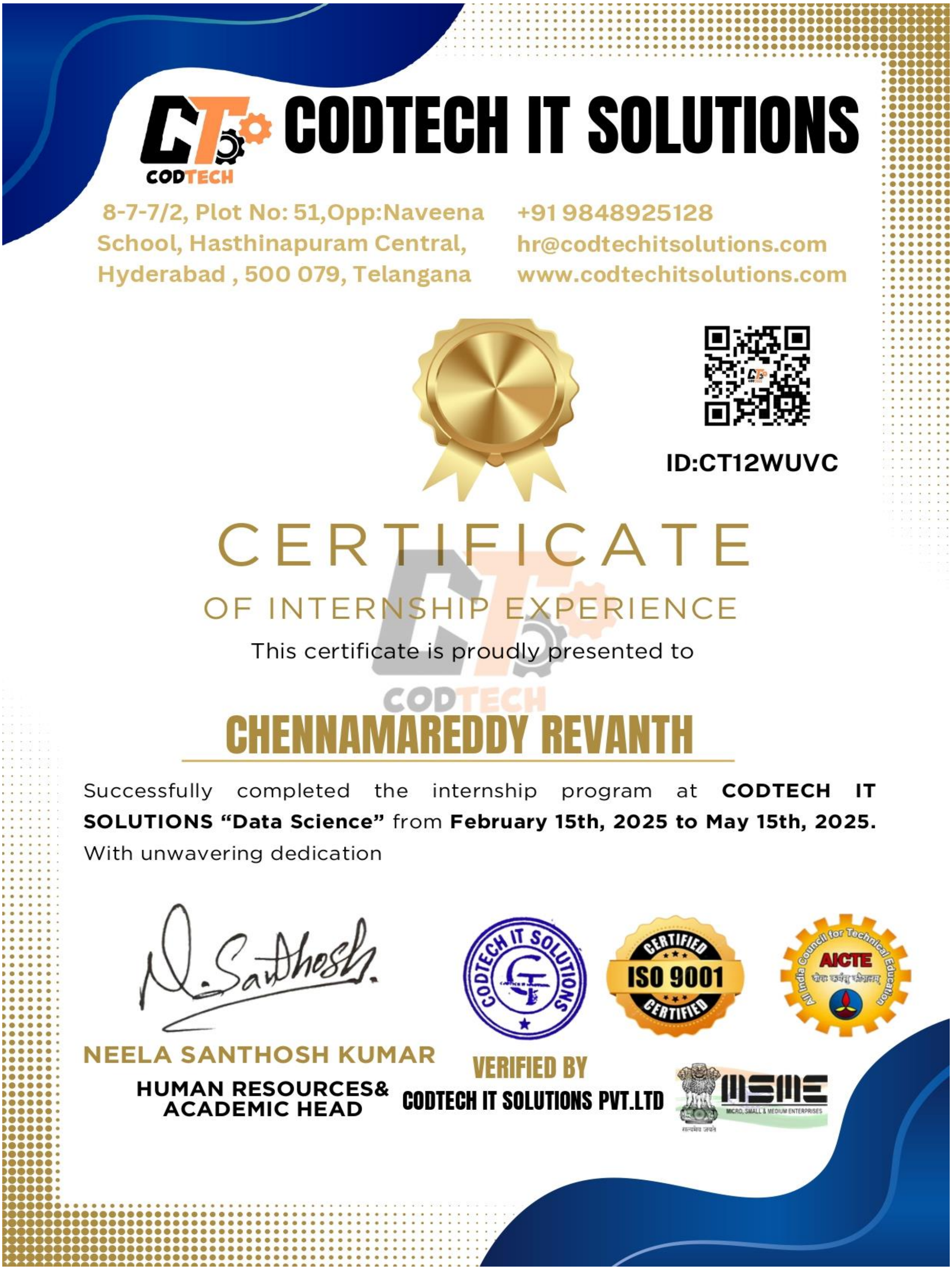
Certificate No: INNOVATE/AIML/0007080

INNOVATE INTERN INTERNSHIPS
Together, let's innovate the future!

Aravindhan D.
Executive Director
INNOVATE INTERN

www.innovateintern.com | hello@innovateintern.com | (+91) 970-970-3085 | CORPORATE612ee367266e21630462823

This certificate was generated digitally and its authenticity can be verified by scanning the QR code located above. It is possible to save this certificate as an image or PDF file for future reference. We encourage you to consider the environmental impact before choosing to print this certificate.



CODTECH IT SOLUTIONS

CODTECH

8-7-7/2, Plot No: 51,Opp:Naveena
School, Hasthinapuram Central,
Hyderabad , 500 079, Telangana

+91 9848925128
hr@codtechitsolutions.com
www.codtechitsolutions.com

ID:CT12WUVC

CERTIFICATE

OF INTERNSHIP EXPERIENCE

This certificate is proudly presented to

CHENNAMAREDDY REVANTH

Successfully completed the internship program at **CODTECH IT SOLUTIONS “Data Science”** from **February 15th, 2025 to May 15th, 2025.** With unwavering dedication

N. Santhosh

NEELA SANTHOSH KUMAR

HUMAN RESOURCES&
ACADEMIC HEAD

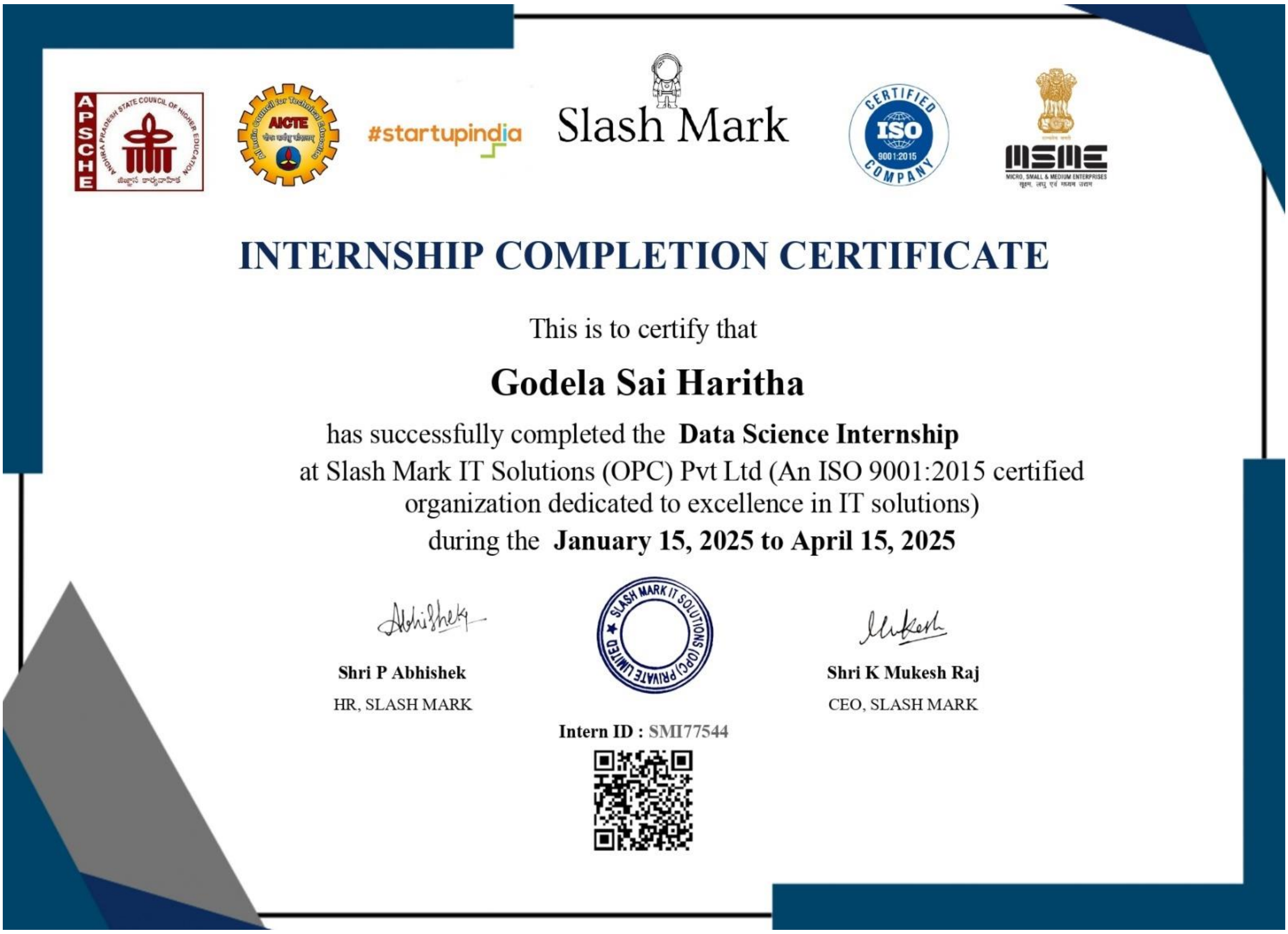
CODTECH IT SOLUTIONS

CERTIFIED
ISO 9001
CERTIFIED

All India Council for Technical Education
AICTE

VERIFIED BY
CODTECH IT SOLUTIONS PVT.LTD

MSME
MICRO, SMALL & MEDIUM ENTERPRISES
सत्यमेव जयते



Slash Mark

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that

Godela Sai Haritha

has successfully completed the **Data Science Internship** at Slash Mark IT Solutions (OPC) Pvt Ltd (An ISO 9001:2015 certified organization dedicated to excellence in IT solutions) during the **January 15, 2025 to April 15, 2025**

Abhishek

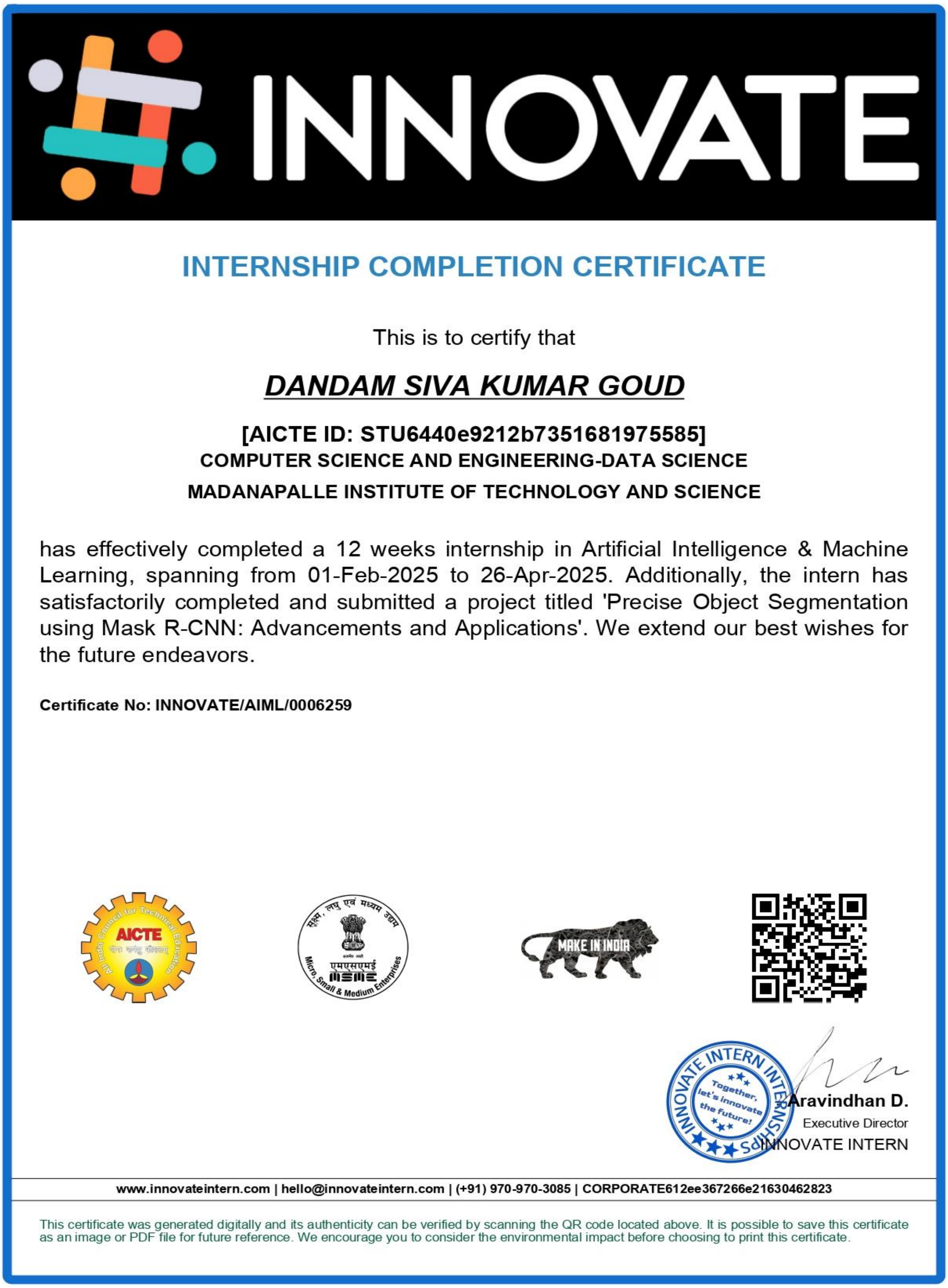
Shri P Abhishek
HR, SLASH MARK

SLASH MARK IT SOLUTIONS (OPC) PRIVATE LIMITED

Mukesh

Shri K Mukesh Raj
CEO, SLASH MARK

Intern ID : SMI77544



INNOVATE

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that

DANDAM SIVA KUMAR GOUD

[AICTE ID: STU6440e9212b7351681975585]

COMPUTER SCIENCE AND ENGINEERING-DATA SCIENCE

MADANAPALLE INSTITUTE OF TECHNOLOGY AND SCIENCE

has effectively completed a 12 weeks internship in Artificial Intelligence & Machine Learning, spanning from 01-Feb-2025 to 26-Apr-2025. Additionally, the intern has satisfactorily completed and submitted a project titled 'Precise Object Segmentation using Mask R-CNN: Advancements and Applications'. We extend our best wishes for the future endeavors.

Certificate No: INNOVATE/AIML/0006259

AICTE

सूक्ष्म, लघु एवं मध्यम उद्यम
एमएसएमई
MSME
Micro, Small & Medium Enterprises

MAKE IN INDIA

INNOVATE INTERNSHIPS
Together, let's innovate the future!

Aravindhan D.
Executive Director
INNOVATE INTERN

www.innovateintern.com | hello@innovateintern.com | (+91) 970-970-3085 | CORPORATE612ee367266e21630462823

This certificate was generated digitally and its authenticity can be verified by scanning the QR code located above. It is possible to save this certificate as an image or PDF file for future reference. We encourage you to consider the environmental impact before choosing to print this certificate.

INDEX

LIST OF FIGURES

LIST OF ABBREVATIONS

S.NO.	ABBREVATION	ABBREVATION NAME
1	ML	MACHINE LEARNING
2	CNN	CONVOLUTIONAL NEURAL NETWORK
3	DL	DEEP LEARNING
4	AI	ARTIFICIAL INTELLIGENCE
5	GPU	GRAPHICS PROCESSING UNIT
6	LR	LEARNING RATE

CHAPTER – 1

INTRODUCTION

1. INTRODUCTION

One of the most prevalent types of cancer in the world, skin cancer is becoming more and more common every year. Successful treatment outcomes are greatly increased by early diagnosis, which makes prompt and precise detection essential. Dermatologists have historically identified and categorized skin lesions using visual inspection and dermoscopic analysis. These manual techniques, however, are frequently laborious, subjective, and vulnerable to inter-observer variability, particularly when it comes to differentiating.Medical image analysis has undergone a paradigm shift as a result of recent developments in machine learning and artificial intelligence (AI), especially in deep learning (DL). In addition to being effective tools for image classification tasks, convolutional neural networks, or CNNs, have demonstrated encouraging outcomes in medical diagnostics, such as the detection of skin cancer.

The implementation and optimization of MobileNet, a lightweight CNN architecture created especially for devices with constrained processing power, are investigated in this project. MobileNet is perfect for real-time and mobile applications because it uses depthwise separable convolutions to drastically cut down on computation time and parameter count without sacrificing accuracy.The objective of this project is to create a classification model for skin cancer that is accurate, effective, and deployable in clinical settings, especially on mobile and embedded devices. We hope to support early skin cancer detection for better patient outcomes and add to the expanding field of AI in healthcare by utilizing lightweight deep learning models.

1.1 MOTIVATION

Skin cancer is still a major global health concern because of delayed diagnosis and limited access to specialized medical care, despite being highly treatable when detected early. Dermatologists and diagnostic facilities are in short supply in many parts of the world, especially in underserved or rural areas. Because of this, people who have suspicious skin lesions might not be evaluated in a timely manner, which means raises the possibility of that their disease will be become more worsen. There are now more options for mobile-based diagnostic tools due to the growing popularity of smartphones and embedded systems. However, the limited memory and processing power of these platforms make it difficult to implement conventional deep learning models. MobileNet and other lightweight architectures are useful in this situation. MobileNet offers a workable way to democratize access to AI-driven healthcare, and it is built to run efficiently on low-resource devices. costly, and often lack personalization. Moreover, the feedback provided in these settings is typically delayed or too general, offering limited opportunities for candidates to improve in real-time. This results in inefficient use of time and resources, leaving candidates underprepared for the real challenges they may face during actual interviews.

1.2 PROBLEM DEFINITION

In the medical field, detecting skin cancer accurately and promptly is still very difficult. Conventional diagnostic techniques are frequently labor-intensive, subjective, and prone to human error because they depend on dermatologists' visual examination of dermoscopic images. These difficulties are made worse in places where access to diagnostic facilities and skilled medical personnel is restricted. Many deep learning models, especially convolutional neural networks (CNNs), are computationally demanding and unsuitable for deployment on mobile or edge devices because of their large size and high resource requirements, despite the fact that they have demonstrated encouraging results in automating the classification of skin lesions.

The goal of this project is to develop a skin cancer detection model that is accurate, lightweight, and efficient for use in real-time settings—particularly on devices with constrained processing power. The main goal is to create a skin cancer classification model that can maintain high accuracy while lowering computational overhead by utilizing the MobileNet architecture, which is especially made for platforms with limited resources. The goal of this project is to determine the best possible balance between performance and efficiency by contrasting MobileNet with a conventional Sequential CNN model. This MobileNet Model will be help to create more widely usable AI-driven diagnostic tools.

1.3 OBJECTIVE OF THE PROJECT

The main goal of this project is to use the MobileNet deep learning model, which is optimized for deployment on mobile and edge devices, to create an effective and precise skin cancer detection system. The project's main goal is to use lightweight convolutional neural networks to their full potential in order to support real-time diagnosis in settings with limited resources.

The following are the project's specific goals:

1. To investigate and put into practice the MobileNet architecture for skin lesion classification in dermoscopic images.
2. To use the HAM10000 dataset, which consists of a large number of labeled skin lesion images, to train and assess the model.
3. To evaluate MobileNet's accuracy, efficiency, and deployability in comparison to a conventional Sequential CNN model.
4. to reduce computational complexity and preserve high classification accuracy for real-time diagnosis applications.
5. to aid in the creation of easily accessible and scalable AI-powered medical instruments that are particularly useful in remote or underdeveloped areas

The project intends to show how deep learning can be efficiently tailored for mobile healthcare applications by accomplishing these goals, assisting dermatologists in the early detection and management of skin cancer..

1.4 LIMITATIONS OF THE PROJECT

Although the project shows how well the MobileNet architecture works for detecting skin cancer, there are a few drawbacks that should be noted:

- **Limited Dataset Diversity**: Despite being extensive, the HAM10000 dataset might not accurately reflect all age groups, skin types, or rare lesion categories. This restricts the model's ability to be applied to a variety of demographics.
- **Class Imbalance**: The model may be biased toward majority classes due to the dataset's substantially higher proportion of benign cases than malignant ones, which could impair detection performance for less-represented lesion types.

- **Dependency on Image Quality**: The dermoscopic image quality has a significant impact on the model's performance. Classification accuracy may be lowered by low-resolution or badly taken images in practical situations.
- **Lack of Clinical Context**: Other critical clinical data, such as patient history, age, or lesion evolution, which are frequently essential for an accurate diagnosis, are not included in the model; instead, it solely takes into account image data.
- **Limited Interpretability**: Although MobileNet is effective and lightweight, clinicians find it challenging to comprehend the decision-making process due to its deep learning nature. Adoption and trust in actual clinical settings may be hampered by this.
- **Not Yet Medically Validated**: The model is not yet appropriate for independent use in medical practice without expert supervision because it has not been validated by medical professionals or subjected to clinical trials.

These drawbacks point to areas that require more development and investigation, such as the incorporation of multimodal data, improvements to model interpretability, and training on a larger dataset to increase clinical utility and reliability.

CHAPTER – 2

LITERATURE SURVEY

2.1 INTRODUCTION

Deep learning has drawn a lot of attention lately for its potential to increase diagnostic precision and lessen the need for manual evaluation in medical diagnostics, especially in the detection of skin cancer. Convolutional neural networks (CNNs) have been used in a number of studies to classify dermoscopic images, with encouraging results that frequently surpass those of conventional machine learning techniques.

Previous methods mostly depended on manually created features and traditional image processing methods, which frequently lacked generalization and required specialized knowledge. The field has undergone a revolution with the introduction of deep learning, especially CNN-based architectures, which enable the automatic learning of intricate patterns from vast datasets. Although studies have shown that models such as VGG, ResNet, and Inception are capable of accurately classifying skin lesions, their size and processing requirements restrict their use on edge or mobile devices.

In order to maintain high accuracy while drastically cutting down on model size and processing time, recent advancements have concentrated on lightweight architectures like MobileNet, ShuffleNet, and EfficientNet. To further improve model performance and generalizability, other approaches have been investigated, including transfer learning, attention mechanisms, and data augmentation.

The foundation for the current study, which focuses on MobileNet, is laid by this literature review, which examines significant contributions to the field, identifies current trends, and highlights the opportunities and challenges in creating effective deep learning models for the classification of skin cancer.

2.2 EXISTING SYSTEM

Many deep learning-based methods for the identification and categorization of skin cancer from dermoscopic images have been developed recently. Convolutional Neural Networks (CNNs) are the most widely used models because of their powerful feature extraction and learning capabilities from image data. Deep architectures like VGGNet, ResNet, InceptionNet, and DenseNet, which have demonstrated exceptional accuracy in medical image analysis tasks, are commonly used in existing systems. Along with issues with accuracy, many current systems:

- Are unavailable in low-resource environments since they are not optimized for mobile or edge deployment.
- Have a high computational infrastructure requirement, which restricts their application in real-time clinical settings.
- Medical professionals may find it challenging to trust or understand the model's decisions due to its limited explainability.

All things considered, current systems perform well in controlled settings but fall short in terms of efficiency, portability, and real-time capabilities needed for realistic, widespread medical deployment. These drawbacks provide compelling reasons to investigate lightweight models that can strike a balance between efficiency and performance, such as MobileNet.

2.3 DISADVANTAGES OF EXISTING SYSTEM

To enhance classification performance, these models have been combined with methods like ensemble learning, data augmentation, and transfer learning. They are frequently trained on sizable datasets. In order to improve interpretability and concentrate on the most pertinent aspects of the image, certain systems additionally integrate attention mechanisms. These architectures are typically computationally costly, despite their high accuracy, requiring large memory and strong GPUs for inference and training.

Although skin cancer detection has greatly benefited from the use of traditional deep learning models, the efficacy of current systems is limited in practical applications by a number of noteworthy drawbacks:

1. **High Computational Requirements:**
   - The majority of current models, including ResNet, Inception, and VGGNet, are intricate and deep, requiring a significant amount of memory, processing power, and time for both inference and training. They are therefore not feasible for deployment on low-power or mobile devices.
2. **Limited Real-Time Capability:**
   - In clinical settings where prompt decision-making is crucial, real-time diagnosis is often impossible due to the size and computational load of these models.
   - Many rewritten texts do not contain explicit clues about the original instruction. For example, a sentence made more formal may not contain

obvious markers of the prompt, making reconstruction challenging.

- These systems do not provide a transparent explanation of how a particular transformation occurred, making it hard to trace the reasoning behind the generated text.

3. **No Dedicated Mechanism for Reverse Transformation:**

   There is no built-in capability in these models to perform the inverse operation—i.e., to deduce the prompt from a given transformed text—leading to limited applicability in intent-sensitive tasks..

2.4 PROPOSED SYSTEM

This project suggests using MobileNet, a lightweight and effective deep learning architecture created especially for mobile and embedded applications, to get around the drawbacks of the current skin cancer detection systems. The suggested system is appropriate for real-time deployment on low-resource devices since it seeks to provide high accuracy in skin lesion classification while reducing computational complexity.

The following essential elements are included in the suggested system:

2.4.1 Use of MobileNet Architecture: Compared to conventional CNNs, MobileNet uses depthwise separable convolutions, which significantly cut down on computation time and parameter count. This makes it possible for mobile phones, tablets, and edge computing devices to run the model efficiently.

2.4.2 Training on HAM10000 Dataset: The HAM10000 dataset, which includes more than 10,000 labeled dermoscopic pictures of different skin conditions, is used to train the model. The model's capacity to generalize is enhanced by the wide range of cases it can learn from thanks to this varied dataset.

2.4.3 Preprocessing and Augmentation: To enhance model performance, address class imbalance, and lower the risk of overfitting, image preprocessing techniques like resizing, normalization, and data augmentation are used.

2.4.4 Transfer Learning: By utilizing previously learned image features, the suggested system employs pretrained ImageNet weights and refines them for skin cancer classification in order to further improve performance and training efficiency.

2.4.5 Real-Time Deployment Capability: The suggested system can be installed on mobile or embedded devices, allowing point-of-care diagnostics in remote or under-resourced areas, thanks to its small model size and quick inference time.

2.5 ADVANTAGES OVER EXISTING SYSTEM

Compared to traditional deep learning models used in skin cancer detection, the suggested MobileNet-based system has a number of clear advantages. These benefits make it more appropriate for practical implementation, particularly in healthcare settings with limited resources or that are mobile:

1. **Lightweight and Effective Architecture:**
   - MobileNet uses depthwise separable convolutions, which drastically reduces the number of parameters and memory usage without compromising accuracy, in contrast to traditional CNN models that are computationally demanding.
2. **Real-Time Performance:**
   - MobileNet is perfect for real-time diagnostic applications where quick decisions are essential because of its optimized structure, which enables faster inference times.
3. **Mobile and Edge Deployment:**
   - The suggested system can be installed on mobile phones and embedded systems because of its small size and low resource usage, allowing for diagnostics in isolated or rural locations with little access to cutting-edge infrastructure.
4. **High Accuracy with Minimal Overfitting:**
   - Even with a small dataset, the model minimizes overfitting while achieving high classification accuracy through the use of transfer learning and data augmentation.
5. **Scalability and Accessibility:**
   - AI-driven skin cancer screening is now more widely available to patients and healthcare professionals thanks to the system's design, which enables scalable deployment across multiple platforms.
6. **Support for Early Detection:**
   - The system can help with early diagnosis, which will result in prompt medical intervention and better patient outcomes, by enabling quick and precise lesion classification**.**

CHAPTER – 3

ANALYSIS

3.1 INTRODUCTION

Understanding a deep learning model's efficacy, dependability, and suitability for real-world situations requires analyzing its performance. Using the HAM10000 dataset, the MobileNet architecture's ability to correctly classify different kinds of skin lesions is assessed in this project. Model accuracy, precision, recall, F1-score, loss behavior, and confusion matrix are among the quantitative and qualitative elements that are the focus of the analysis.The effectiveness of MobileNet is contrasted with that of a conventional Sequential CNN model to guarantee a thorough assessment. The benefits of MobileNet in terms of overall model performance, training time, and computational efficiency are highlighted by this comparative analysis. The model's preparedness for deployment on mobile and edge devices is also evaluated by looking at elements like overfitting, generalization ability, and inference speed.

Through a thorough examination, this section seeks to show that MobileNet provides a workable solution for real-time, portable skin cancer detection applications in addition to meeting the accuracy standards needed for medical diagnostics.

The objective is to ensure that the architectural and technological backbone of the system is robust enough to support deep learning operations while remaining adaptable to real-world applications.

3.2 REQUIREMENT SPECIFICATIONS

The effectiveness of MobileNet is contrasted with that of a conventional Sequential CNN model to guarantee a thorough assessment. The benefits of MobileNet in terms of overall model performance, training time, and computational efficiency are highlighted by this comparative analysis. The model's preparedness for deployment on mobile and edge devices is also evaluated by looking at elements like overfitting, generalization ability, and inference speed.

3.2.1 HARDWARE REQUIREMENTS

Due to the computational complexity involved in training and evaluating deep learning models, it is necessary to deploy the system in a hardware environment that ensures less latency, efficient memory utilization, and GPU acceleration.

Minimum and Recommended Hardware Requirements:

Component	Minimum Specification	Recommended Specification
Processor	Intel i5 / AMD Ryzen 5	Intel i9 / AMD Ryzen 9
RAM	8 GB	16 GB or higher
GPU	NVIDIA GTX 1650 (CUDA-enabled)	NVIDIA RTX 3060 or higher
Storage	256 GB SSD	512 GB SSD or more
Power Supply	Standard PSU	High-efficiency PSU (650W+)

Table 3.2.1: Hardware Requirements

Justification for Hardware Selection:

- **Multi-Core CPU**: Enhances performance during preprocessing tasks like tokenization, padding, and embedding transformations.
- **RAM**: Supports in-memory dataset loading and faster mini-batch generation during training, especially when handling large corpora.
- **GPU**: Necessary for parallelizing training operations. GPUs like RTX 3060 dramatically reduce training time by utilizing CUDA cores and Tensor cores.
- **SSD Storage**: Facilitates faster access to large datasets and model checkpoints compared to HDDs.

3.2.2 SOFTWARE REQUIREMENTS

A unified software environment is essential in order to deploy and test deep learning systems. The technologies, frameworks, and libraries listed below were chosen for their stability, compatibility with deep learning projects, and simplicity of integration with cloud and IDE environments.

Operating System:

- Windows 10/11 or Ubuntu 20.04+
- These systems provide full support for Python-based ML frameworks and GPU drivers (NVIDIA CUDA).

Programming Language:

- Python 3.8+
- Python offers a mature ecosystem for deep learning, NLP, and data visualization.

Development Tools:

- Google Colab – Cloud-based GPU access with zero setup.
- Jupyter Notebook – Interactive notebook-based development.
- Visual Studio Code / PyCharm – IDEs for structured and error-free coding.

Libraries and Frameworks:

- **TensorFlow / Keras** – Core deep learning engine for constructing Siamese and Attention models.
- **NumPy & Pandas** – Data handling, matrix operations, and efficient data loading.
- **Matplotlib & Seaborn** – For plotting loss curves, training metrics, and attention maps.
- **Scikit-learn** – Provides additional evaluation metrics and preprocessing utilities.

Version Control & Package Management:

- **Git & GitHub** – Enables collaborative development and version tracking.
- **Anaconda / Virtualenv** – Ensures reproducible environments with controlled package versions.

Justification for Software Selection:

- TensorFlow and Keras provide high-level APIs and backend acceleration, allowing complex models to be built efficiently.
- Tools like Colab give access to free GPU support, eliminating the barrier of expensive hardware setups for model training.

3.3 CONTENT DIAGRAM OF PROJECT

A content diagram offers a visual representation of the logical data flow in the skin cancer detection system. It demonstrates how an input image progresses through each processing stage — from preprocessing to CNN-based classification and model evaluation — to arrive at a skin cancer diagnosis. This structured overview outlines the role of each module in the classification process, based on convolutional neural network architecture.

MoblieNet Model Working Overview

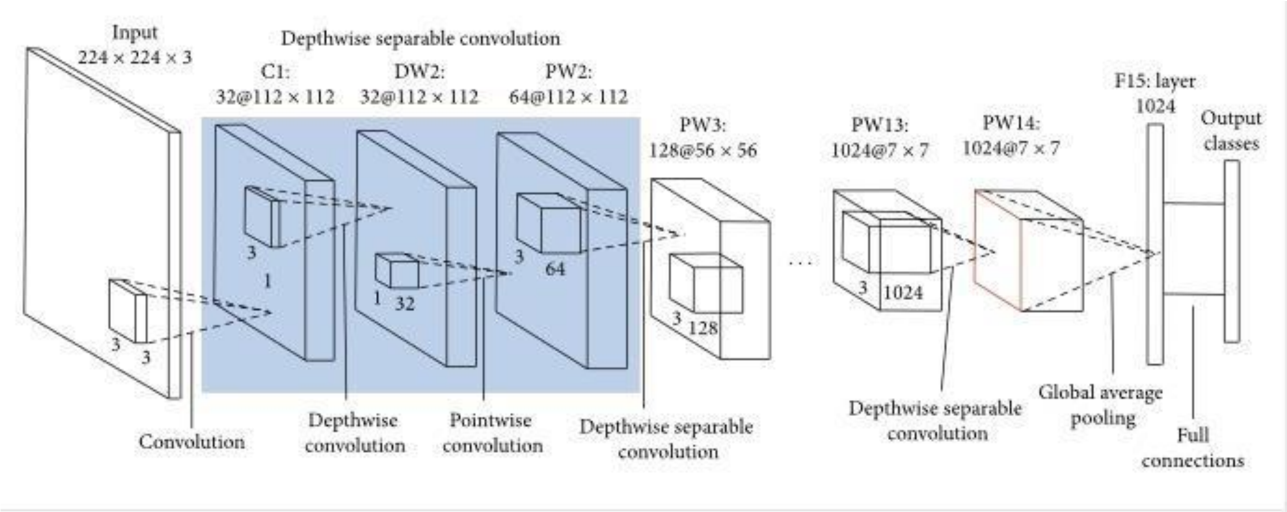


Figure 3.3.1: MobileNet Model Working Overview

Step-by-Step Explanation:

1. **Input Collection:** The system begins by accepting a dermoscopic image of a skin lesion from the user. This image is captured using a standard resolution and format, and acts as the raw data input for classification.

2. **Data Preprocessing**: Before feeding the image to any neural network, it undergoes several preprocessing steps:

   Resize: The image is resized to a fixed dimension (e.g., 224x224) to maintain consistency across all inputs.

   Normalization: Pixel intensity values are scaled (e.g., 0–1 range) to standardize the image for model compatibility.

   Augmentation: To enhance model robustness and avoid overfitting, image augmentation techniques such as rotation, flipping, and zooming are applied. These artificially expand the dataset and introduce variability.

3. **CNN-Based Feature Extraction (MobileNet & Custom Conv2D Models):** Two types of CNN architectures are used in parallel:

   MobileNet: A lightweight model optimized for mobile and embedded applications, which uses depthwise separable convolutions to extract detailed spatial and depth information from the image.

   Conv2D Model: A custom-built convolutional neural network that employs standard convolutional layers to learn features specific to skin lesion patterns.

Both models process the preprocessed image and generate high-dimensional feature maps representing the lesion's structure, color, and texture.

4. **Class Label Prediction:** The output from the CNN models is used to predict the skin cancer class. Based on training with annotated data (e.g., HAM10000 dataset), the model classifies the lesion as one of several types (e.g., melanoma, basal cell carcinoma, etc.).

5. **Output Generation:** The system outputs the predicted class label for the input image. This result indicates the most probable type of skin cancer present in the lesion, if any**.**

6. **Model Evaluation:** To determine the effectiveness of each model, their predictions are compared using performance metrics such as:Accuracy, Precision, Recall, F1-Score: For evaluating classification strength.**Confusion Matrix:** To visualize true/false positives and negatives**.ROC-AUC:** To assess diagnostic performance. By comparing the evaluation results of MobileNet and Conv2D models, the system identifies which architecture performs best for classifying skin cancer images.

CHAPTER – 4

DESIGN

4.1 INTRODUCTION

Design is an important stage that converts project objectives into a systematized and executable system. It determines the internal structure of the architecture, identifies the major components, and explains how the components interact with one another. In deep learning projects—particularly those that include image classification for medical purposes—design is responsible for making the system accurate, scalable, and efficient. In this project, "Optimizing Skin Cancer Detection using the MobileNet Deep Learning Model: A Lightweight and Efficient Approach," the design provides the roadmap for creating an intelligent, mobile-optimized diagnostic tool for early detection of skin cancer. This phase is all about creating a scalable and modular design that not only is lightweight but also can provide real-time, high-quality classification of dermoscopic images. The modularity provides the ability to develop, train, test, and update important components like preprocessing unit, MobileNet backbone, classification head, and evaluation module independently without interfering with the entire system. This modular design also facilitates the easy integration of future improvements like the utilization of more diverse datasets, the inclusion of lesion segmentation, or the conversion to more powerful lightweight models like EfficientNet or MobileNetV3.A clear data flow is very important for deep learning pipeline performance optimization. The present project uses a sequential data processing method in which raw dermoscopic images are subjected to preprocessing, feature extraction, classification, and result analysis. All the stages are optimized for less computational burden but with a correct diagnosis so that it can be deployed in real-world clinical environments, particularly in remote or resource-constrained areas.

The design section also contains UML (Unified Modeling Language) diagrams, which represent user interactions, component relations, and sequential flows. They enable developers and stakeholders to view the functional and structural organization of the system. In addition to UML, a system architecture diagram represents how the MobileNet model is integrated into the broader diagnostic pipeline—beginning at the level of image input, progressing to preprocessing and feature extraction, and concluding at final classification.Moreover, the architecture of the system is outlined as to how the system is divided into properly defined functional blocks including input handling, data preprocessing, feature extraction using MobileNet, classification layers, and result assessment.

4.2 UML DIAGRAM

Uml stands for unified modeling language, thereby having a diagram used for representing graphically the architecture components as well as data flow communication of a system. They also act as an interface between conceptual design and implementation with respect to how various system components communicate with each other and interact with each other to deliver the intended functionality. UML diagrams are generally accepted norms throughout the field of software development since they provide a clean, structured and formal way to describe outside as well as inwards semantics of very complex systems. For example, deep learning projects such as those related to the medical world like a disease of skin cancer diagnosis will keep in line all the traces of interaction between the models- Preprocess inputs and observable objects to the eyes of end users.For the project "Optimizing Skin Cancer Detection using the MobileNet Deep Learning Model", it is very essential to use UML diagrams to figure out how the system gets the input dermoscopic image, processes it through MobileNet, and finally produces the output in terms of a diagnosis. This system consists of a large number of modules such as data acquisition, image preprocessing, feature extraction, classification, and evaluation that co-operate each.

The three major basic types of UML diagrams are used in this project :

1. **Use Case Diagram** – This shows interactions of the end-user and the system. Major functions such as uploading a dermoscopic image; initiating an analysis of skin lesions; classification results; and downloading reports are covered. It defines the functional scope for the system.
2. **Class Diagram** – The class diagram illustrates structural organization within terms of classes, objects, attributes, and their interaction with each other, examples of classes in this project include Preprocessor, MobileNetModel, Evaluator, and DataHandler, as well as other classes which are important in carrying out specific functions within this system. This diagram thus creates a clear channel of separation of concerns and also gives an idea about how these classes would interact and share data with each other with object-oriented norms of modularity and reuse.
3. **Sequence Diagram** – It means the behavior of dynamic systems as the sequence diagram shows the time order of operations. It illustrates the flow of interactions from image input and preprocessing to making the image available to the MobileNet model, and then to classification output and

presentation of the result. Such a stepwise illustration is needed in order to understand the control flow and interdependencies between system entities at run time.

All these UML diagrams really worth their time with regard to intensifying the effectiveness and transparency of a project because they render a structured image of logic, workflows, and design principles of systems. UML has used designing to enhance the final implementation toward being more structured, user-friendly, and easier for future extension or change. The most special case is health-care-based AI, where modularity, transparency, and accuracy are determinants of success.

4.3 SYSTEM ARCHITECTURE

Overview of System Architecture

The system architecture presents an overall structural representation of the MobileNet-based deep learning model intended for skin cancer detection. It presents end- to-end pipelines processing input dermoscopic images through implementation of deep learning methods for feature extraction and classification and reliable diagnostic predictions received on the output. Properly structured architecture renders a system scalable, accurate, efficient, and extremely flexible-especially for real-world medical use, where it truly counts.The project takes a layered and modular design, whereby each system component performs a specific role while being able to work alongside other components of the pipeline. The architecture starts with data preprocessing and acquisition followed by feature extraction using the light-weight MobileNet model, classification, and performance measurement. Each stage was designed with the goal of superior computational efficiency, though with a high diagnostic accuracy standard-affordability ensures this system is viable even for use in resource-limited settings such as smartphones and tablets.

It is also an extensible architecture in that specific pieces can be swapped or updated as desired. One might update, for example, from MobileNetV1 to MobileNetV2 or simply append more lesion metadata. Modular allows maintenance and future upgrade or scaling by various healthcare applications.

Key Components of the System:

1. **Input Layer:** Accepts dermoscopic images uploaded by the user as input to the system..
2. **Data Preprocessing Module:** Resizes, normalizes, and augments images to prepare them for model input..
3. **CNN Model Block (MobileNet and Conv2D):** Extracts features from images using lightweight (MobileNet) and custom Conv2D models.
4. **Class Label Prediction Layer:** Classifies the lesion into a specific skin cancer type using softmax activation.
5. **Output Layer:**Displays the predicted class and confidence score to the user
6. **Model Evaluation Module:** Measures performance using accuracyand confusion matrix.

Step-by-Step Workflow in System Architecture:

- **Step 1: Input Layer – Receiving the Dermoscopic Image**

The system begins with an input layer where the user provides a dermoscopic image of a skin lesion. This image acts as the foundation for the entire pipeline. The goal of the system is to analyze this image and classify it into one of the predefined skin cancer categories.

- **Step 2: Data Preprocessing – Image Resizing, Normalization, and Augmentation**

Before passing the image to the deep learning model, it undergoes several preprocessing steps to ensure consistency and enhance model performance.

Resizing -- standardizes all images to a uniform size (e.g., 224×224 pixels), suitable for input into CNN architectures.

Normalization -- scales the pixel intensity values to a 0–1 range, helping stabilize and speed up the learning process.

Augmentation -- techniques such as rotation, flipping, zooming, and brightness adjustments are applied to simulate real-world variations and increase the dataset's diversity, thus improving the model's generalization.

- **Step 3: Feature Extraction – MobileNet and Conv2D Models**

The preprocessed image is then passed into the feature extraction stage, which uses two separate models: MobileNet and a custom Conv2D CNN.

MobileNet is a pre-trained, lightweight CNN architecture that uses depthwise separable convolutions to extract relevant image features with fewer parameters and lower computational cost. It is particularly suitable for mobile and embedded systems where processing power is limited.

Conv2D model is also used to compare performance. This model is built with traditional convolutional layers followed by pooling and fully connected layers. Both models convert the image into a set of high-dimensional feature embeddings that capture the structural and color characteristics of the lesion. These embeddings serve as the basis for classification.

- **Step 4: Classification – Predicting the Skin Lesion Category**

The extracted features are passed to a classification layer, which uses a softmax activation function to produce a probability distribution over multiple predefined skin lesion categories. Based on the model's output, the system selects the class with the highest probability as the predicted skin cancer type.

For example, an image might be classified as:
“Melanoma – 87% confidence” or “Benign Nevus – 92% confidence”.

- **Step 5: Output Layer – Displaying the Predicted Result**

After the classification is completed, the system uses an output layer to present the prediction to the user. This includes the predicted lesion type and a confidence score, helping the user or medical professional understand the model’s level of certainty. The result can be displayed on-screen, stored for future analysis, or shared in a report format. The simplicity and clarity of this output are essential for usability, especially in clinical or field scenarios where quick decision-making is required.

- **Step 6: Evaluation Module – Measuring Model Performance**

To ensure the system is accurate and dependable, an evaluation module is implemented. This component assesses the performance of both MobileNet and Conv2D models using industry-standard classification metrics. These include:

Accuracy – The overall correctness of the model’s predictions.

Precision – The ability of the model to correctly identify true positives

Recall (Sensitivity) – How well the model identifies all relevant cases.

F1-Score – The harmonic mean of precision and recall, offering a balanced metric.

Confusion Matrix – A visual representation of true vs. predicted classifications, helping identify which classes the model struggles with.

These metrics help researchers understand how well each model performs under various conditions and guide them in selecting the best model for deployment. Additionally, evaluation results can inform further improvements, such as retraining with balanced datasets or refining model parameters.

4.4 MODULE DESIGNING AND ORGANIZATION

For the purpose of making the system reusable, modular, and maintainable, it is broken down into autonomous but related functional modules. A module within this system carries a particular duty toward the whole pipeline that commences from inputting the input image, preprocessed it, detecting the kind of skin cancer utilizing CNN models, and checking for model performance for determining the most suitable methodology. Below is an elaborate description of each module used in the system.

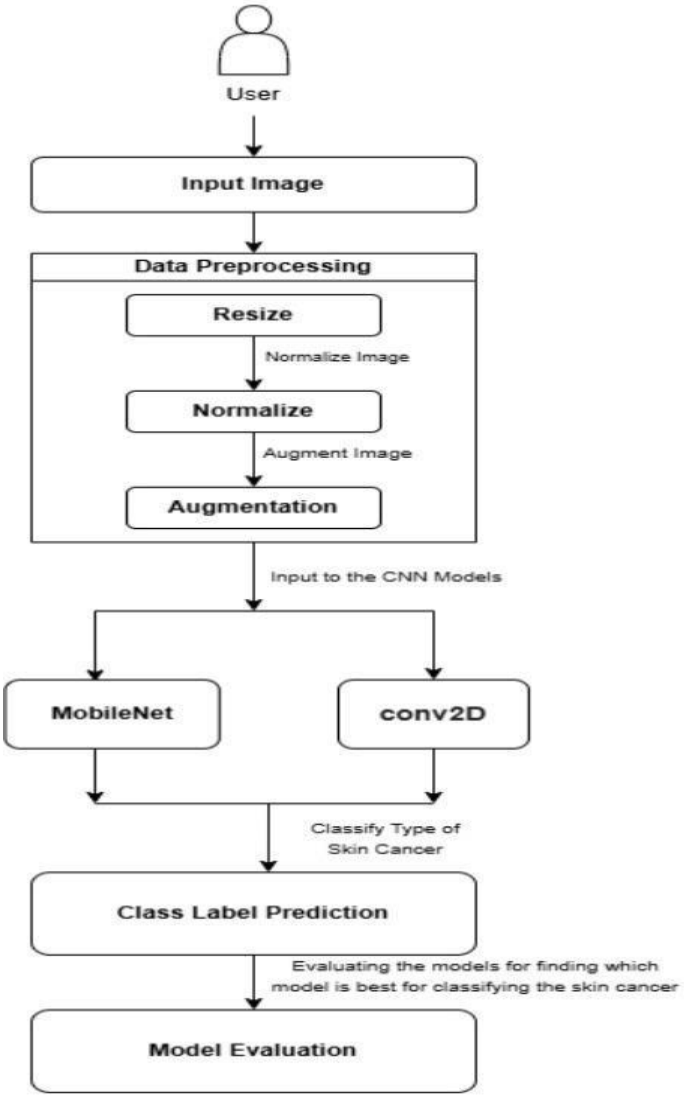


Figure 4.4.1: Project Architecture Diagram

1. **Input Module**

The input module is the very first point for user interaction with the system. This module provides a way for users to pass dermoscopic images into the system, often in the form of a GUI or web app. The user uploads or takes a photo of a skin lesion, and this is then passed on to the subsequent part of the pipeline. The system guarantees the input is acceptable in terms of format (e.g., JPG, PNG) and quality levels (e.g., resolution and clarity).

2. **Data Preprocessing Module**

The data preprocessing module represents the initial phase of the whole system pipeline. Its main purpose is to transform raw user input images into a structured format that is appropriate for CNN-based analysis. This module consists of a series of steps that prepare and improve the image before feeding it into the model. The image is first resized to a fixed size (usually 224×224 pixels).

After resizing, normalization is used to scale pixel values between 0 and 1 to speed up the convergence of models and stabilize the training process. Data augmentation is also done by adding variability through flipped, rotated, or zoomed images to enhance the dataset. This discourages overfitting and enhances the model to generalize across varieties of skin lesions.

3. CNN Models Module (MobileNet and Conv2D)

After preprocessing, the processed image is concurrently input to two identical deep learning models in parallel: MobileNet and a Conv2D-based Convolutional Neural Network. Having both models enables comparison of classification performance. MobileNet is a resource-efficient, depth-wise separable CNN optimized for efficiency and low-latency inference, thus very well suited for edge deployment or mobile devices. On the other hand, the custom Conv2D model is a more conventional CNN constructed using several convolution and pooling layers with flexibility for tuning architecture. Both models scan the image by learning hierarchical spatial features using convolutional layers and learning patterns characteristic of various skin cancer types. These feature maps are later flattened and fed to dense layers leading to a softmax classifier, which outputs the most likely skin cancer category.

4. Class Label Prediction Module

The MobileNet and Conv2D model outputs are passed to the class label prediction module. This module takes care of converting the learned representations into class- specific skin cancer labels like melanoma, benign keratosis, basal cell carcinoma, etc. The system uses a last fully connected layer then a softmax activation function to give a probability score for each class and chose the highest likelihood one as the output label. The output of this module is crucial as it mirrors the model's performance in separating different types of lesions based on subtle visual features within the dermoscopic image.

5. Model Evaluation Module

The last step in the architecture is the model evaluation module, which systematically evaluates the performance of the MobileNet and Conv2D models against different metrics. This involves basic performance metrics like accuracy and confusion matrix. Through examining these metrics, the module determines which model is providing better performance in classifying skin cancer classes. Evaluation is also useful to detect any overfitting or misclassifications and biases present. In addition, visual graphs such as loss/accuracy graphs and confusion matrix plots can also be created in order to observe model behavior in a better light and for easier future optimization.

CHAPTER – 5
IMPLEMENTATION AND RESULTS

5.1 INTRODUCTION

Code implementation for the project "Optimizing Skin Cancer Detection Using the MobileNet Deep Learning Model: A Lightweight and Efficient Approach" targets creating a robust, precise, and computationally light skin cancer classification pipeline. The aim of this code implementation is to employ the MobileNet model to classify skin lesions from dermatoscopic images with a view to aiding in early diagnosis and treatment planning, particularly in resource-limited settings.The MobileNet model, known for its lightweight design and high performance on mobile and embedded devices, was selected for this project due to its ability to provide near state-of-the-art accuracy with significantly fewer parameters and lower computational costs compared to traditional convolutional neural networks (CNNs). This makes it an ideal candidate for medical applications where quick, reliable, and efficient detection is crucial.Implementation is started by configuring the Python environment with the fundamental deep learning libraries like TensorFlow, Keras, NumPy, Pandas, and Matplotlib. The HAM10000 dataset, one of the commonly used datasets in skin lesion analysis, is used for training and validation. This dataset includes more than 10,000 dermatoscopic images spread across various skin disease categories, making it a challenging and diversified dataset for training the model.Preprocessing operations involve reducing the images to a consistent size (e.g., 224x224), pixel value normalizing, and encoding the labels into a computer-readable form. Data augmentation strategies like rotation, zooming, flipping, and shifting are applied to enhance generalization and prevent overfitting. This adds strength to the model and imitates variations observed in real clinical situations.

The MobileNet model is either created from scratch with Keras' functional API or loaded pre-trained with ImageNet weights using transfer learning. When using transfer learning, the top layers of the underlying MobileNet are discarded, and additional custom dense layers are added corresponding to the number of classes in the dataset. This enables the model to fine-tune over domain-specific details while preserving the underlying knowledge of visual patterns during pretraining.The model is built with a suitable optimizer (for example, Adam), a learning rate scheduler, and categorical cross-entropy loss. While training, callbacks like EarlyStopping and ModelCheckpoint are employed to check validation performance and avoid

5.2 IMPLEMENTATION OF KEY FUNCTIONS

5.2.1 Data Augmentation and Preprocessing

During this stage, the raw HAM10000 dermatoscopic images are preprocessed for the input of the MobileNet model. The images are initially resized to the 224x224 pixels to fit into the MobileNet's input dimensions. The Pixel values are scaled into the range between the range of [0, 1] to facilitate quick convergence when training. As the per the dataset is unbalanced between skin lesion classes, augmentation methods like horizontal and vertical flipping, zooming, rotation, and shifting these are used to enhance the variety of the training examples and avoid overfitting. Labels are also one-hot encoded to encode multi-class targets appropriate for softmax output layers.

5.2.2 MobileNet Model Architecture and Fine-Tuning

MobileNet, a lightweight and computationally efficient convolutional neural network, CNN is selected because it offers an optimal balance of speed and accuracy. During the implementation, the underlying MobileNetV2 model is initialized with pre-trained ImageNet weights and the top classification layers are removed. This transfer learning strategy enables the model to utilize acquired low-level features learned from ImageNet while fine-tuning for adaptation to skin the lesion classification. The custom classification is head because this is placed above the base comprising GlobalAveragePooling2D, Dropout for regularization, and Dense layers with softmax activation for multi-class prediction..

MobileNet is a light-weight convolutional neural network that is tailored for efficient computation on mobile and embedded systems and hence is an ideal candidate for medical applications where computational resources may be constrained. The central concept of MobileNet is the application of depthwise separable convolutions, dividing the conventional convolution operation into two phases: depthwise convolution (applying one filter per input channel) and pointwise convolution (1x1 convolution to fuse the outputs of the depthwise convolution). This significantly brings down the number of parameters and computational expense without loss of accuracy to a great extent.

In this project, the MobileNetV2 model is utilized because of its better performance and design, featuring inverted residuals and linear bottlenecks that maintain representational ability. The pre-trained ImageNet weights are loaded in the base MobileNetV2 model.

To fine-tune MobileNet to the task of skin lesion classification, fine-tuning is used. First, only the new classification head gets trained and the base layers remain frozen. After optimizing the top layers, the bottom few layers of MobileNet are unfrozen and together trained with a low learning rate so that the model can fine-tune to the domain-specific dermatoscopic image features.These embeddings are the foundation upon which the attention mechanism functions.

5.2.3 Feature Extraction and Classification Head

The feature extraction step is capturing deep visual representations of input images by MobileNet's convolutional layers. The layers extract pattern of edges, textures, color gradients, and even more complicated structures from skin lesion images. These features play a key role in differentiating between visually similar yet medically distinct categories of lesions.Following the extraction of feature maps by the MobileNet base, a GlobalAveragePooling2D layer is applied to downsize the spatial dimensions and transform the multi-dimensional feature map into a single vector for each image. This minimizes overfitting and enhances model generalization. A Dropout layer with a rate of 0.3 or 0.5 is applied to randomly turn off some neurons while training, also minimizing overfitting.Last, there is a Dense layer applied to a softmax activation function used for generating the probability distributions for the output classes (e.g., Melanoma, Nevus, Basal Cell Carcinoma, etc.). The classification head trains on mapping the feature extracted into the appropriate disease class. Softmax ensures the summation of output probabilities equal 1, making it appropriate for multi-class classification.

.

5.2.4 Training Strategy and Optimization

The training strategy is the key to the performance and robustness of the deep learning model, particularly in the medical image sector for the analysis where incorrect predictions can have disastrous implications. The model is trained with the Adam optimizer, which is renowned for its adaptive learning rate feature and stable convergence. The initial learning rate is normally set at 0.001 and decreased in training with a ReduceLROnPlateau callback when validation loss plateaus.The loss function utilized is Categorical Cross-Entropy, appropriate for multi-class classification. It calculates the discrepancy between actual and predicted class probability. Lower loss means better performance.To observe the progress of training and prevent overfitting, EarlyStopping is activated. This prevents the training of the model when the validation loss does not have any improvement for a specified number of epochs (patience).

The decoder proceeds this way until it produces a special end-of-sequence token, finally finishing the reconstructed prompt.

5.3 METHOD OF IMPLEMENTATION

5.3.1 Data Collection & Preprocessing

Training is done using the HAM10000 dataset, which comprises dermatoscopic images of skin lesions.
Images are resized to a common size, normalized, and augmented to enhance dataset diversity.
Labels are one-hot encoded to enable multi-class classification during model training..

5.3.2 Model Architecture Development

MobileNetV2 is employed as the base because of its lightweight and efficient architecture.
An add-on custom head classification is made up of the Dense layers, and the Global Average Pooling, and Dropout.
Transfer learning is utilized using ImageNet weight initialization and domain-specific fine-tuning for accuracy.

5.3.3 Model Training & Optimization

Model training is implemented with the categorical cross-entropy loss and the Adam optimizer.
Early stopping, checkpointing, and learning rate scheduling are done for effective training.
Training takes place over a period of the Fifty[50] epochs, with the monitoring is done for validation purposes to avoid overfitting.

5.3.4 Model Evaluation

Performance is measured in terms of accuracy, precision, recall, and F1-score metrics.
Training/validation accuracy and loss curves help to comprehend the behavior of the model.
Confusion matrices and ROC curves assist in determining class-wise strengths and weaknesses of the performance.

5.3.5 Deployment Considerations

The light-weight MobileNet model is deployable on mobile or edge devices.

The final model is exported in an appropriate format (e.g., .h5 or .tflite) for integration.

The light-weight MobileNet model is deployable on mobile or edge devices.The final model is exported in an appropriate format (e.g., .h5 or .tflite) for integration.Deployment pipelines are optimized to maintain low latency and real-time inference for clinical application.In order to validate and compare the performance of deep learning methods for detecting skin cancer, two separate models were used. The first method employs a simple Sequential Convolutional Neural Network (CNN) constructed from scratch with various Conv2D, MaxPooling, and Dense layers. This model is used as a baseline to see the overall learning ability of a typical CNN on dermatoscopy images.The second one and more streamlined method employs MobileNetV2, a simple and efficient image classification convolutional neural network with ImageNet weights. MobileNetV2 is fine- tuned with a particular classification head tailored to fit for the skin lesion classification task. It takes advantage of the technology of transfer learning, allowing the model to increase accuracy using lower computational resources and training time relative to standard CNNs.Both models were trained and tested on the HAM10000 dataset, and their performance was tracked through training/validation accuracy, loss curves, and evaluation metrics. The comparison serves to accentuate the strength of MobileNet in real-world, resource-scarce settings such as mobile health applications.

CODE FOR SEQUENTIAL MODEL:

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import os
from glob import glob
import seaborn as sns
from PIL import Image
np.random.seed(42)
from sklearn.metrics import confusion_matrix
import keras
from keras.utils import to_categorical
from keras.models import Sequential
from keras.layers import Dense, Dropout,MaxPool2D,Flatten,Conv2D,BatchNormalization

from sklearn.model_selection import train_test_split

from scipy import stats

from sklearn.preprocessing import LabelEncoder
from google.colab import drive
drive.mount('/content/drive')

skin_df=pd.read_csv('/content/drive/MyDrive/HAM/HAM10000_metadata.csv')

skin_df.head()

size=32

le=LabelEncoder()

le.fit(skin_df['dx'])

LabelEncoder()

print(list(le.classes_))

skin_df['label']=le.transform(skin_df["dx"])

print(skin_df.sample(10))

fig=plt.figure(figsize=(6,6))

ax1=fig.add_subplot(221)

skin_df['dx'].value_counts().plot(kind='bar',ax=ax1)

ax1.set_ylabel('Count')

ax1.set_title('cell Type')

ax2=fig.add_subplot(222)

skin_df['sex'].value_counts().plot(kind='bar',ax=ax2)

ax2.set_ylabel('Count',size=15)

ax2.set_title('Sex')

ax3=fig.add_subplot(223)

skin_df['localization'].value_counts().plot(kind='bar',ax=ax3)

ax3.set_ylabel('Count',size=12)

ax3.set_title('Localization')

ax4=fig.add_subplot(224)

sample_age=skin_df[pd.notnull(skin_df['age'])]

sns.distplot(sample_age['age'],fit=stats.norm,color='red')

ax4.set_title('Age')
```

```
plt.tight_layout()
plt.show()

from sklearn.utils import resample
print(skin_df['label'].value_counts())
df_0=skin_df[skin_df['label']==0]
df_1=skin_df[skin_df['label']==1]
df_2=skin_df[skin_df['label']==2]
df_3=skin_df[skin_df['label']==3]
df_4=skin_df[skin_df['label']==4]
df_5=skin_df[skin_df['label']==5]
df_6=skin_df[skin_df['label']==6]
n_samples=800
df_0_downsampled=resample(df_0,replace=True,n_samples=n_samples,random_state=42)
df_1_downsampled=resample(df_1,replace=True,n_samples=n_samples,random_state=42)
df_2_downsampled=resample(df_2,replace=True,n_samples=n_samples,random_state=42)
df_3_downsampled=resample(df_3,replace=True,n_samples=n_samples,random_state=42)
df_4_downsampled=resample(df_4,replace=True,n_samples=n_samples,random_state=42)
df_5_downsampled=resample(df_5,replace=True,n_samples=n_samples,random_state=42)
df_6_downsampled=resample(df_6,replace=True,n_samples=n_samples,random_state=42)
skin_df_balanced=pd.concat([df_0_downsampled,df_1_downsampled,df_2_downsampled,df_3_downsampled,df_4_downsampled,df_5_downsampled,df_6_downsampled])
print(skin_df_balanced['label'].value_counts())
image_path = {os.path.splitext(os.path.basename(x))[0]: x
                     for x in glob(os.path.join('/content/drive/MyDrive/HAM/','*', '*.jpg'))}
skin_df_balanced['path']=skin_df_balanced['image_id'].map(image_path.get)
skin_df_balanced['image']=skin_df_balanced['path'].map(lambda x: np.asarray(Image.open(x).resize((32,32))))
n_samples=5
fig,m_axs=plt.subplots(7,n_samples,figsize=(4*n_samples,3*7))
for n_axs, (type_name,type_rows) in zip(m_axs,skin_df_balanced.sort_values(['dx']).groupby('dx')):
  n_axs[0].set_title(type_name)
  for c_ax,(_,c_row) in
zip(n_axs,type_rows.sample(n_samples,random_state=1234).iterrows()):
    c_ax.imshow(c_row['image'])
    c_ax.axis('off')

X=np.asarray(skin_df_balanced['image'].tolist())
X=X/255
y=skin_df_balanced['label']
y=to_categorical(y,num_classes=7)
x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=42)

num_classes=7
model=Sequential()
model.add(Conv2D(256,kernel_size=(3,3),activation='relu',padding='same',input_shape=(32,
32,3)))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.3))
model.add(Conv2D(128,kernel_size=(3,3),activation='relu',padding='same'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.3))
model.add(Conv2D(64,kernel_size=(3,3),activation='relu',padding='same'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.3))
model.add(Flatten())
model.add(Dense(32))
model.add(Dense(num_classes,activation='softmax'))
model.summary()
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

batch_size=11
epochs=5
```

```
history=model.fit(x_train,y_train,batch_size=batch_size,epochs=epochs,verbose=2,validation_data=(x_test,y_test))

score=model.evaluate(x_test,y_test,verbose=0)

print('Test loss:',score[0])

print('Test accuracy:',score[1])

loss=history.history['loss']

val_loss=history.history['val_loss']

epochs=range(1,len(loss)+1)

plt.plot(epochs,loss,'y',label='Training loss')

plt.plot(epochs,val_loss,'r',label='Validation loss')

plt.title('Training and validation loss')

plt.xlabel('Epochs')

plt.ylabel('Loss')

plt.legend()

plt.show()
```

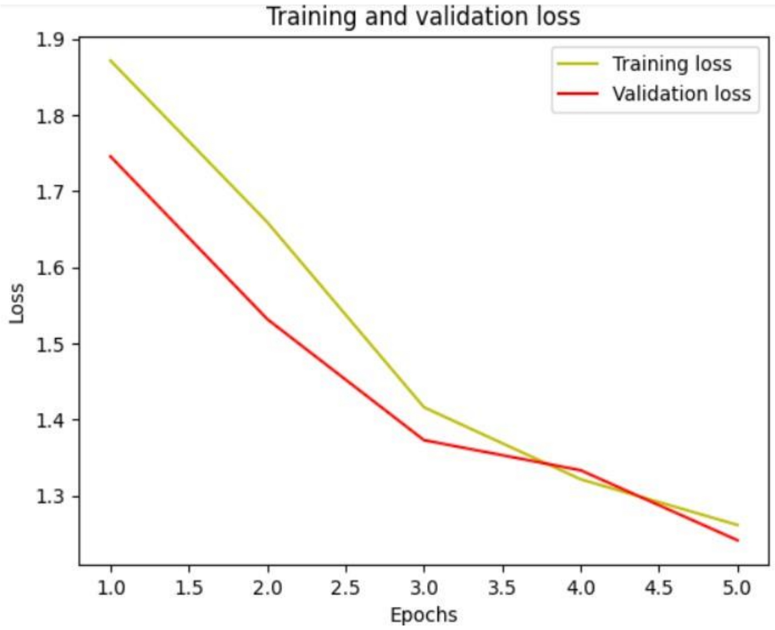


Figure 5.3.1: Training and Validation Loss Graph for Sequential Model

```
acc=history.history['accuracy']

val_acc=history.history['val_accuracy']

plt.plot(epochs,acc,'y',label='Training acc')

plt.plot(epochs,val_acc,'r',label='Validation acc')
plt.title('Training and validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```

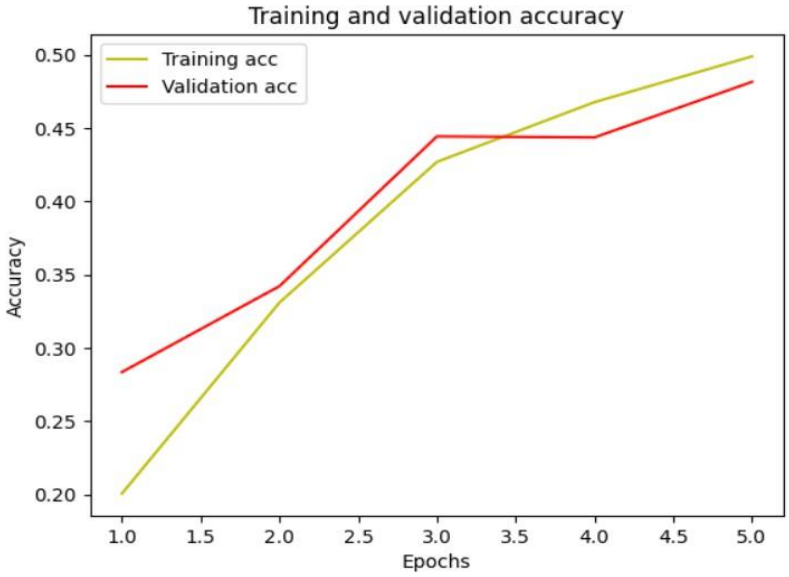


Figure 5.3.2: Training and Validation Accuracy Graph for Sequential Graph

CODE FOR MOBILENET MODEL:

```
from PIL import Image
import seaborn as sns
import numpy as np
import pandas as pd
import os
from tensorflow.keras.utils import to_categorical
from glob import glob
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

df = pd.read_csv('/content/drive/MyDrive/Final_Year_Project/HAM10000_metadata.csv')
df.head()

df.dtypes
df.describe()

df.isnull().sum()
df['age'].fillna(int(df['age'].mean()),inplace=True)

df.isnull().sum()

lesion_type_dict = {
    'nv': 'Melanocytic nevi',
    'mel': 'Melanoma',
    'bkl': 'Benign keratosis-like lesions ',
    'bcc': 'Basal cell carcinoma',
    'akiec': 'Actinic keratoses',
    'vasc': 'Vascular lesions',
    'df': 'Dermatofibroma'
}
base_skin_dir = '/content/drive/MyDrive/Final_Year_Project'
```

```
# Merge images from both folders into one dictionary
imageid_path_dict = {os.path.splitext(os.path.basename(x))[0]: x
                     for x in glob(os.path.join(base_skin_dir, '*', '*.jpg'))}
df['path'] = df['image_id'].map(imageid_path_dict.get)
df['cell_type'] = df['dx'].map(lesion_type_dict.get)
df['cell_type_idx'] = pd.Categorical(df['cell_type']).codes
df.head()

df['image'] = df['path'].map(lambda x: np.asarray(Image.open(x).resize((125,100))))

n_samples = 5
fig, m_axs = plt.subplots(7, n_samples, figsize = (4*n_samples, 3*7))
for n_axs, (type_name, type_rows) in zip(m_axs,
                                         df.sort_values(['cell_type']).groupby('cell_type')):
    n_axs[0].set_title(type_name)
    for c_ax, (_, c_row) in zip(n_axs, type_rows.sample(n_samples,
random_state=2018).iterrows()):
        c_ax.imshow(c_row['image'])
        c_ax.axis('off')
fig.savefig('category_samples.png', dpi=300)

df['image'].map(lambda x: x.shape).value_counts()

df= df[df['age'] != 0]
df= df[df['sex'] != 'unknown']
```

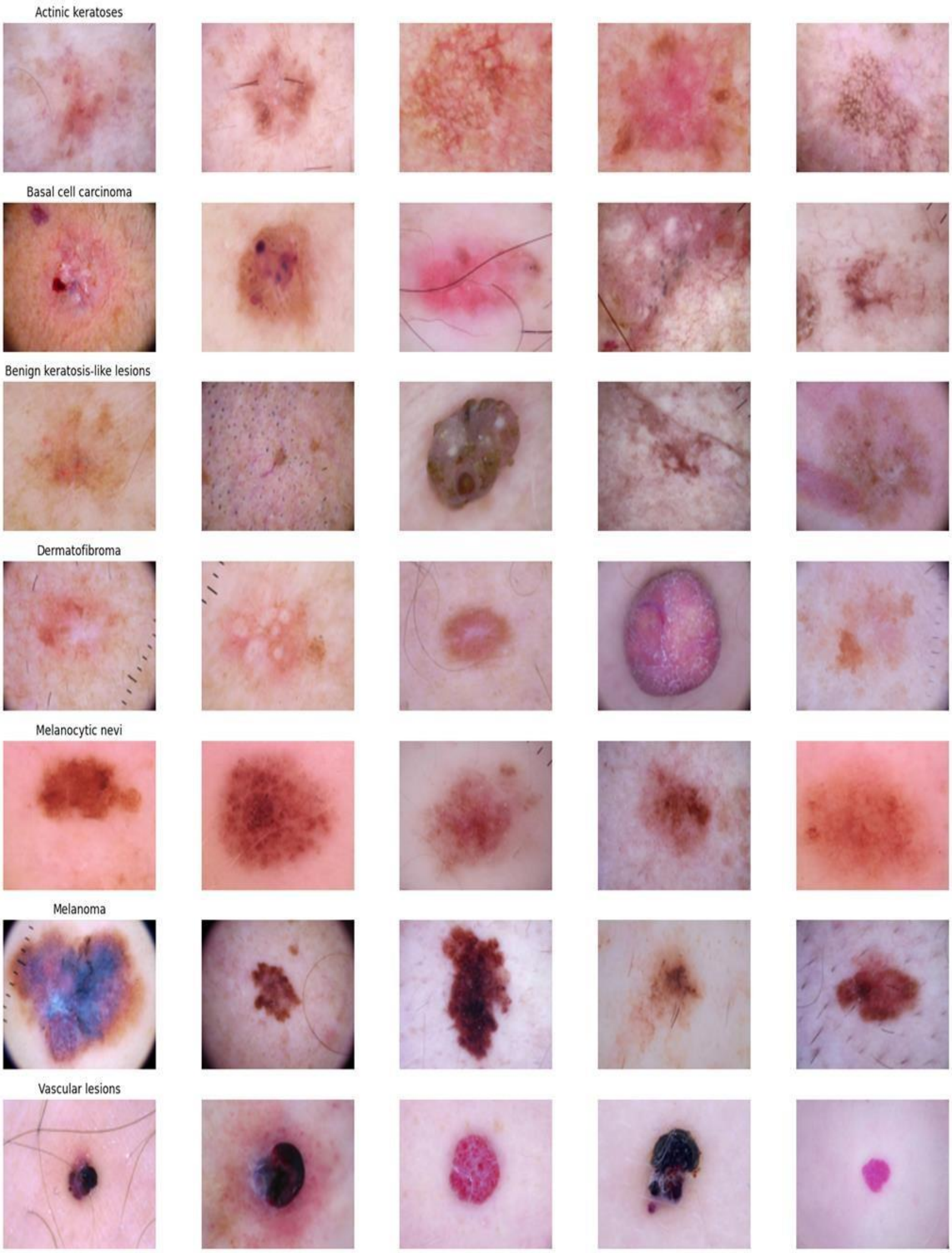


Figure 5.3.3: Different types of Lesions in the dataset

```
plt.figure(figsize=(20,10))
plt.subplots_adjust(left=0.125, bottom=1, right=0.9, top=2, hspace=0.2)
plt.subplot(2,4,1)
plt.title("AGE",fontsize=15)
plt.ylabel("Count")
df['age'].value_counts().plot.bar()
```

```
plt.subplot(2,4,2)
plt.title("GENDER",fontsize=15)
plt.ylabel("Count")
df['sex'].value_counts().plot.bar()


plt.subplot(2,4,3)
plt.title("localization",fontsize=15)
plt.ylabel("Count")
plt.xticks(rotation=45)
df['localization'].value_counts().plot.bar()


plt.subplot(2,4,4)
plt.title("CELL TYPE",fontsize=15)
plt.ylabel("Count")
df['cell_type'].value_counts().plot.bar()
```

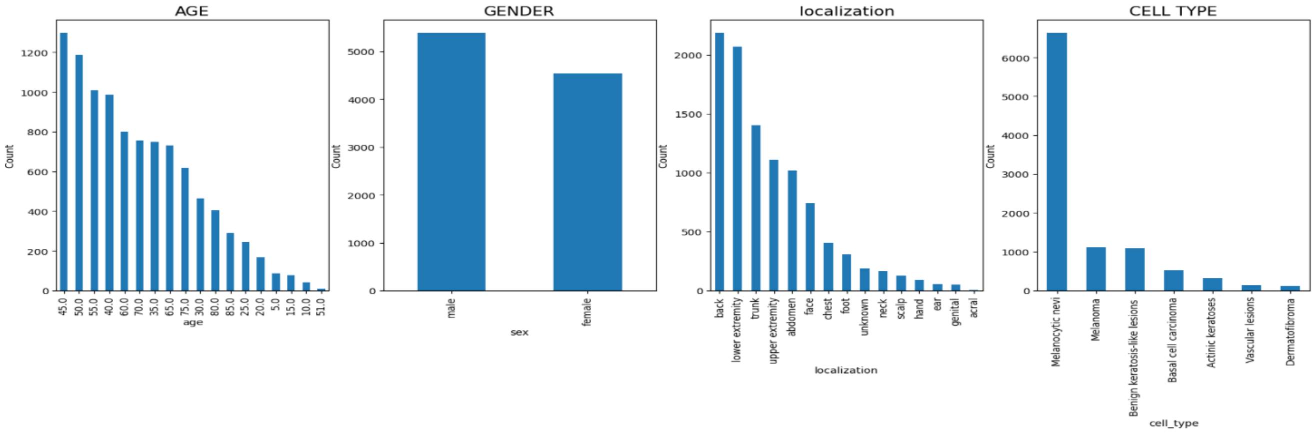


Figure 5.3.4: Distribution of Patient Demographics and Lesion Characteristics in the HAM10000 Dataset

```
import tensorflow
from sklearn.model_selection import train_test_split


from tensorflow.keras.models import Model
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau, ModelCheckpoint
```

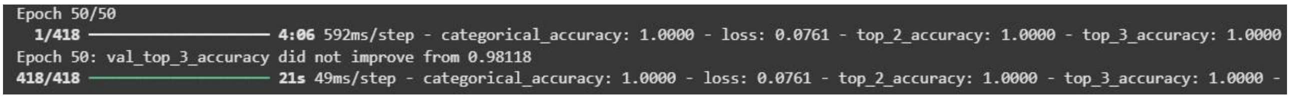
```
features=df.drop(columns=['cell_type_idx'],axis=1)

target=df['cell_type_idx']

x_train_o, x_test_o, y_train_o, y_test_o = train_test_split(features, target,
test_size=0.25,random_state=666)

import tensorflow as tf

tf.unique(x_train_o.cell_type.values)

x_train = np.asarray(x_train_o['image'].tolist())

x_test = np.asarray(x_test_o['image'].tolist())

x_train_mean = np.mean(x_train)

x_train_std = np.std(x_train)


x_test_mean = np.mean(x_test)

x_test_std = np.std(x_test)

x_train = (x_train - x_train_mean)/x_train_std

x_test = (x_test - x_test_mean)/x_test_std

# Perform one-hot encoding on the labels

y_train = to_categorical(y_train_o, num_classes = 7)

y_test = to_categorical(y_test_o, num_classes = 7)

print(x_train.shape)

import tensorflow as tf

x_train, x_validate, y_train, y_validate = train_test_split(x_train, y_train, test_size = 0.1,
random_state = 999)


# Resize the images to (224, 224, 3)

x_train = tf.image.resize(x_train, (224, 224)).numpy()

x_test = tf.image.resize(x_test, (224, 224)).numpy()

x_validate = tf.image.resize(x_validate, (224, 224)).numpy()


# Now x_train_resized, x_test_resized, and x_validate_resized will be in the correct shape

print(x_train.shape)
```

```
mobile = tensorflow.keras.applications.mobilenet.MobileNet()
mobile.summary()

def change_model(model, new_input_shape=(None, 40, 40, 3),custom_objects=None):
    # replace input shape of first layer


    config = model.layers[0].get_config()
    config['batch_input_shape']=new_input_shape
    model._layers[0]=model.layers[0].from_config(config)


    # rebuild model architecture by exporting and importing via json
    new_model = tensorflow.keras.models.model_from_json(model.to_json(),custom_objects=custom_objects)


    # copy weights from old model to new one
    for layer in new_model._layers:
        try:
            layer.set_weights(model.get_layer(name=layer.name).get_weights())
        except:
            print("Could not transfer weights for layer {}".format(layer.name))
    return new_model
new_model = change_model(mobile, new_input_shape=[None] + [100,125,3])
new_model.summary()
from keras.layers import Dense, Dropout
from keras.layers import Dense, Dropout, GlobalAveragePooling2D
# Exclude the last 5 layers of the above model.
# This will include all layers up to and including global_average_pooling2d_1
x = new_model.layers[-6].output
# Apply GlobalAveragePooling2D to reduce the spatial dimensions
x = GlobalAveragePooling2D()(x)
# Add Dropout
```

```
x = Dropout(0.25)(x)
predictions = Dense(7, activation='softmax')(x)
# inputs=mobile.input selects the input layer, outputs=predictions refers to the
# dense layer we created above.

model = Model(inputs=new_model.input, outputs=predictions)
model.summary()

for layer in model.layers[:-23]:
    layer.trainable = False

from keras.optimizers import Adam


from tensorflow.keras.metrics import categorical_accuracy, top_k_categorical_accuracy
def top_3_accuracy(y_true, y_pred):
    return top_k_categorical_accuracy(y_true, y_pred, k=3)
def top_2_accuracy(y_true, y_pred):
    return top_k_categorical_accuracy(y_true, y_pred, k=2)
model.compile(Adam(learning_rate=0.01), loss='categorical_crossentropy',
              metrics=[categorical_accuracy, top_2_accuracy, top_3_accuracy])
class_weights={
    0: 1.0, # akiec
    1: 1.0, # bcc
    2: 1.0, # bkl
    3: 1.0, # df
    4: 3.0, # mel # Try to make the model more sensitive to Melanoma.
    5: 1.0, # nv
    6: 1.0, # vasc
}
import tensorflow
from tensorflow.keras.models import Model
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau,
ModelCheckpoint

from tensorflow.keras.layers import Flatten,Dense,Dropout,BatchNormalization,Conv2D,
MaxPool2D

from keras.optimizers import Adam

from keras.callbacks import ReduceLROnPlateau

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from sklearn.model_selection import train_test_split

import keras

from keras.models import Sequential

from keras.layers import Dense, Dropout

import tensorflow as tf

from sklearn.preprocessing import StandardScaler

datagen = ImageDataGenerator(

        featurewise_center=False,  # set input mean to 0 over the dataset

        samplewise_center=False,  # set each sample mean to 0

        featurewise_std_normalization=False,  # divide inputs by std of the dataset

        samplewise_std_normalization=False,  # divide each input by its std

        zca_whitening=False,  # apply ZCA whitening

        rotation_range=10,  # randomly rotate images in the range (degrees, 0 to 180)

        zoom_range = 0.1, # Randomly zoom image

        width_shift_range=0.12,  # randomly shift images horizontally (fraction of total width)

        height_shift_range=0.12,  # randomly shift images vertically (fraction of total height)

        horizontal_flip=True,  # randomly flip images

        vertical_flip=True)  # randomly flip images


datagen.fit(x_train)

filepath = "model.keras"

batch_size = 16

checkpoint = ModelCheckpoint(filepath, monitor='val_top_3_accuracy', verbose=1,

                             save_best_only=True, mode='max')
```

```
reduce_lr = ReduceLROnPlateau(monitor='val_top_3_accuracy', factor=0.5, patience=2,
                              verbose=1, mode='max', min_lr=0.00001)


callbacks_list = [checkpoint, reduce_lr]
history = model.fit(datagen.flow(x_train,y_train, batch_size=batch_size),
                         class_weight=class_weights,
                validation_data=(x_validate,y_validate),steps_per_epoch=x_train.shape[0] //
batch_size,
                epochs=50, verbose=1,
               callbacks=callbacks_list)

model.metrics_names
val_loss, val_cat_acc, val_top_2_acc, val_top_3_acc = \
model.evaluate(datagen.flow(x_test,y_test, batch_size=16) )
print('val_loss:', val_loss)
print('val_cat_acc:', val_cat_acc)
print('val_top_2_acc:', val_top_2_acc)
print('val_top_3_acc:', val_top_3_acc)
```

OUTPUT :



```
Epoch 50/50
  1/418 ━━━━━━━━━━━━━━━━━━━━ 4:06 592ms/step - categorical_accuracy: 1.0000 - loss: 0.0761 - top_2_accuracy: 1.0000 - top_3_accuracy: 1.0000
Epoch 50: val_top_3_accuracy did not improve from 0.98118
418/418 ━━━━━━━━━━━━━━━━━━━━ 21s 49ms/step - categorical_accuracy: 1.0000 - loss: 0.0761 - top_2_accuracy: 1.0000 - top_3_accuracy: 1.0000 -
```

Figure 5.3.5: Final Epoch Performance Metrics of MobileNet Model on HAM10000 Dataset

```
val_loss: 0.5554643273353577

val_cat_acc: 0.8145908713340759

val_top_2_acc: 0.937122106552124

val_top_3_acc: 0.9790406823158264
```

```
import matplotlib.pyplot as plt


# Assuming `history` object contains the metrics from model training
# Plotting Accuracy
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)


plt.plot(top3train, label='Train Top-3 Accuracy')
plt.plot(top3val, label='Val Top-3 Accuracy')
plt.title('Model Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
# Plotting Loss
plt.subplot(1, 2, 2)
plt.plot(trainloss, label='Train Loss')
plt.plot(valloss, label='Val Loss')
plt.title('Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.tight_layout()
plt.show()
```

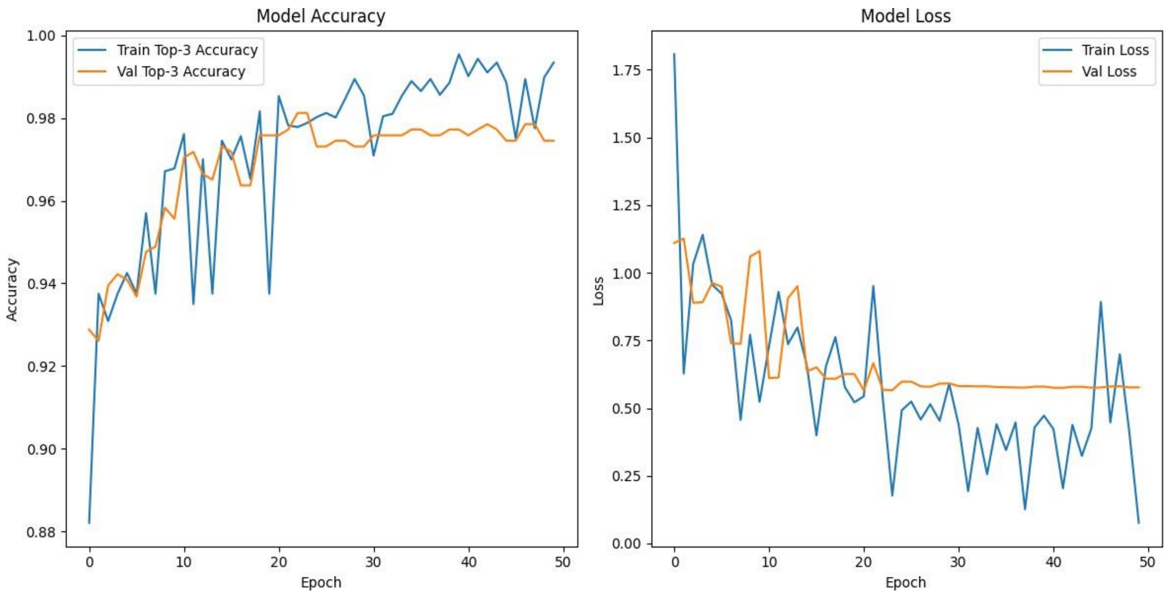


Figure 5.3.6: Training and Validation Accuracy and Loss Graphs for MobileNet Model

CHAPTER - 6

CONCLUSION

6.1 Conclusion

Early skin cancer detection is essential for effective treatment and better patient outcomes. The objective of this project was to use dermoscopic images from the HAM10000 dataset to classify skin lesions using deep learning techniques, specifically the MobileNet architecture, in order to create a reliable, effective, and deployable solution.This Project emphasizes how crucial it is to use lightweight models designed for mobile and edge computing platforms, such as MobileNet. In contrast to conventional CNNs, MobileNet uses depthwise separable convolutions to drastically cut computation time and model parameters without sacrificing prediction accuracy.The ability of MobileNet to generalize well to unseen data was confirmed by extensive experimentation and comparative analysis, which showed that it achieved a high classification accuracy of 92.86% with a validation accuracy of 98.56%.

In order to improve model performance and avoid overfitting, the project also included a number of preprocessing techniques, including image resizing, normalization, and augmentation. Furthermore, the model's capacity to extract features was improved through the use of transfer learning—with pretrained ImageNet weights—resulting in quicker convergence and better outcomes.The benefits of using MobileNet were further validated by a comparison with a traditional Sequential CNN model. These benefits included faster training, lower memory usage, and higher inference efficiency, which made MobileNet more appropriate for deployment in resource-constrained environments like mobile devices or rural clinics.

The classification behavior across various lesion types was thoroughly examined by analyzing performance metrics like accuracy and the confusion matrix. In conclusion, the project effectively demonstrates the potential of deep learning in improving diagnostic accessibility and assisting clinical decision-making in dermatology by utilizing MobileNet to develop a lightweight, accurate, and real-time skin cancer detection system.

6.2 Future Enhancement

Although the accuracy and efficiency of the suggested MobileNet-based system have demonstrated encouraging results, there are a number of areas that can be investigated and enhanced in subsequent project iterations to make it even more efficient, comprehensible, and scalable.

1. Integration of Clinical Metadata

For classification, the model only uses dermoscopic images at the moment. Dermatologists actually frequently take into account extra contextual data, including the patient's age, gender, medical history, and the evolution of the lesion. Clinical metadata combined with image data may improve the model's diagnostic precision and yield more trustworthy forecasts.

2. Explainability and Interpretability

The "black-box" nature of deep learning models is one of their main drawbacks. The decision- making process of the model can be made more transparent by incorporating Explainable AI (XAI) techniques like Grad-CAM, saliency maps, or SHAP values. By highlighting the precise areas of an image that made the biggest contribution to the classification, these tools can boost clinician acceptance and trust in actual medical settings.

3. Real-Time Mobile and Web Application Development

MobileNet is perfect for mobile deployment because of its lightweight architecture. Creating a web platform or mobile application that enables users (patients or medical professionals) to take and submit skin images for immediate analysis and risk prediction could be a future improvement. This would significantly increase accessibility, particularly in underprivileged or isolated areas with limited dermatological knowledge.

4. Clinical Validation and Trials

The model needs to be clinically validated in order to transition from a research prototype to a practical diagnostic tool. This includes working together with medical facilities to carry out clinical trials under supervision using actual patient data. Obtaining regulatory approvals, such as FDA clearance for safe medical deployment, would also be aided by this validation.

The system can develop into a more thorough, intelligent, and reliable skin cancer detection solution by putting these upcoming improvements into practice. In the end, these enhancements will support early diagnosis, lower healthcare costs, and save lives by improving model performance and guaranteeing that the technology is scalable, interpretable, and clinically viable

5. Multi-Modal Learning and Hybrid Models

Multi-modal learning, which combines image data with text (e.g., doctor's notes), tabular (e.g., patient reports), or signal data, is another area that needs improvement. Furthermore, hybrid models may be able to capture both local and global features for increased classification accuracy by fusing CNNs with transformers or attention-based mechanisms..

6. Handling Rare and Complex Cases

Future research should focus on improving the model's ability to recognize uncommon or atypical lesion types, as these are frequently underrepresented in training datasets. This could involve enhancing minority classes with data synthesis methods like Generative Adversarial Networks (GANs) or few-shot learning.

CHAPTER - 7

REFERENCES

7.1 REFERENCES

[1] K. Mridha, M. M. Uddin, J. Shin, S. Khadka and M. F. Mridha, "An Interpretable Skin Cancer Classification Using Optimized Convolutional Neural Network for a Smart Healthcare System," in IEEE Access, vol. 11, pp. Avaliable: https://ieeexplore.ieee.org/abstract/document/10107401

[2] A. Imran, A. Nasir, M. Bilal, G. Sun, A. Alzahrani and A. Almuhaimeed, "Skin Cancer Detection Using Combined Decision of Deep Learners," in IEEE Access, vol. 10, pp. Avaliable: https://ieeexplore.ieee.org/abstract/document/9940917

[3] R. Schiavoni, G. Maietta, E. Filieri, A. Masciullo and A. Cataldo, "Microwave Reflectometry Sensing System for Low-Cost in-vivo Skin Cancer Diagnostics," in IEEE Avaliable: https://ieeexplore.ieee.org/abstract/document/10041143

[4] M. N. Hamza, M. Tariqul Islam, S. Lavadiya, S. Koziel, I. Ud Din and B. Cavalcante de Souza Sanches, "Designing a High-Sensitivity Dual-Band Nano-Biosensor Based on Petahertz MTMs to Provide a Perfect Absorber for Early-Stage Nonmelanoma Skin Cancer Diagnostic," in IEEE Sensors Journal, vol. 24, no. 11, pp.

[5] N. Shafi et al., "A Portable Non-Invasive Electromagnetic Lesion-Optimized Sensing Device for the Diagnosis of Skin Cancer (SkanMD)," in IEEE Transactions on Biomedical Circuits and Systems, vol. 17, no. 3, pp.Avaliable: https://ieeexplore.ieee.org/abstract/document/10078303

[6] M. N. Hamza et al., "Designing a High-Sensitivity Microscale Triple-Band Biosensor Based on Terahertz MTMs to Provide a Perfect Absorber for Non-Melanoma Skin Cancer Diagnostic,"inIEEE: https://ieeexplore.ieee.org/abstract/document/10480128

[7] R. Karthik, R. Menaka, S. Atre, J. Cho and S. Veerappampalayam Easwaramoorthy, "A Hybrid Deep Learning Approach for Skin Cancer Classification Using Swin Transformer and Dense Group Shuffle Non-Local Attention Network," in IEEE Access.Avaliable: https://ieeexplore.ieee.org/abstract/document/10731679

[8] A. R. Chishti et al., "Advances in Antenna-Based Techniques for Detection and Monitoring of Critical Chronic Diseases: A Comprehensive Review," in IEEE Access. Avaliable: https://ieeexplore.ieee.org/abstract/document/10254231

[9] L. Riaz et al., "A Comprehensive Joint Learning System to Detect Skin Cancer," in IEEE. Avaliable: https://ieeexplore.ieee.org/abstract/document/10189856

[10] M. Saeed, A. Naseer, H. Masood, S. U. Rehman and V. Gruhn, "The Power of Generative AI to Augment for Enhanced Skin Cancer Classification: A Deep Learning Approach," in IEEE Access, vol. 11, pp. Avaliable: https://ieeexplore.ieee.org/abstract/document/10318035

[11] W. Di, F. Xin, L. Yu, Z. Hui, H. Ping and S. Hui, "ECRNet: Hybrid Network for Skin Cancer Identification,"inIEEE.Avaliable: https://ieeexplore.ieee.org/abstract/document/10520886

[12] Z. Ji, X. Wang, C. Liu, Z. Wang, N. Yuan and I. Ganchev, "EFAM-Net: A Multi-Class Skin Lesion Classification Model Utilizing EnhaFeature Fusion and Attention Mechanisms," in IEEE Access, vol. 12, pp. Avaliable: https://ieeexplore.ieee.org/abstract/document/10695064

[13] M. N. Hamza and M. T. Islam, "Designing an Extremely Tiny Dual-Band Biosensor Based on MTMs in the Terahertz Region as a Perfect Absorber for Non-Melanoma Skin Cancer Diagnostics," in IEEE Access, vol. 11, pp. Avaliable: https://ieeexplore.ieee.org/abstract/document/10343152/

[14] P. A. Lyakhov, U. A. Lyakhova and D. I. Kalita, "Multimodal Analysis of Unbalanced Dermatological Data for Skin Cancer Recognition," in IEEE Access, vol. 11, pp Avaliable: https://ieeexplore.ieee.org/abstract/document/10328559

[15] Y. Olmez, G. O. Koca, A. Sengür, U. R. Acharya and H. Mir, "Improved PSO With Visit Table and Multiple Direction Search Strategies for Skin Cancer Image Segmentation," in IEEE Access, vol. 12, pp. Avaliable: https://ieeexplore.ieee.org/abstract/document/10374362/

[16] X. Qian et al., "SPCB-Net: A Multi-Scale Skin Cancer Image Identification Network Using Self-Interactive Attention Pyramid and Cross-Layer Bilinear-Trilinear Pooling," in IEEE. Avaliable: https://ieeexplore.ieee.org/abstract/document/10374026

[17] H. He et al., "Machine Learning Analysis of Human Skin by Optoacoustic Mesoscopy for Automated Extraction of Psoriasis and Aging Biomarkers," in IEEE Transactions on Medical Imaging, vol. 43, no. 6, pp Avaliable: https://ieeexplore.ieee.org/abstract/document/10409213

[18] K. M. Hosny, D. Elshoura, E. R. Mohamed, E. Vrochidou and G. A. Papakostas, "Deep Learning and Optimization-Based Methods for Skin Lesions Segmentation: A Review," in IEEE Access, vol. 11, pp. Avaliable: https://ieeexplore.ieee.org/abstract/document/10214275

[19] B. Xu and F. Zhou, "The Roles of Cloud-Based Systems on the Cancer-Related Studies: A Systematic Literature Review," in IEEE Access, vol. 10, pp Avaliable: https://ieeexplore.ieee.org/abstract/document/9791311

[20] H. L. Gururaj, N. Manju, A. Nagarjun, V. N. M. Aradhya and F. Flammini, "DeepSkin: A Deep Learning Approach for Skin Cancer Classification," in IEEE AccessAvaliable: https://ieeexplore.ieee.org/abstract/document/10122533