VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi , Karnataka, INDIA

A Project Report
on

DermaGAN

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted By

SRI SAI TEJA M S 1GA20CS143
ULLAS P 1GA20CS156
VIKAS K R 1GA20CS168

Under the Guidance of
R C RAVINDRANATH
Assistant Professor

Department of Computer Science and Engineering
Accredited by NBA(2022-2025)
GLOBAL ACADEMY OF TECHNOLOGY
(Autonomous Institute, Affilaited to VTU, Belgavi)
Rajarajeshwarinagar, Bengaluru - 560 098

2023 – 2024

GLOBAL ACADEMY OF TECHNOLOGY

(Autonomous Institute, Affilaited to VTU, Belgavi)
Rajarajeshwarinagar, Bengaluru - 560 098

Department of Computer Science and Engineering

Accredited by NBA(2022-2025)

CERTIFICATE

Certified that the Project Entitled **"DermaGAN"** carried out by **SRI SAI TEJA M S**, bearing **USN 1GA20CS143, ULLAS P**,bearing **USN 1GA20CS156, VIKAS K R**, bearing **USN 1GA20CS168,** bonafide students of Global Academy of Technology, in partial fulfillment for the award of the **BACHELOR OF ENGINEERING** in **Computer Science and Engineering** from Visvesvaraya Technological University, Belagavi during the year 2023-2024. It is certified that all the corrections/suggestions indicated for Internal Assessment have been incorporated in the Report submitted to the department. The Partial Project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the said Degree.

R C Ravindranath Assistant Professor Dept. of CSE GAT, Bengaluru.	Dr. Kumaraswamy S Professor & Head Dept. of CSE GAT, Bengaluru.	Dr. N. Ranapratap Reddy Principal GAT, Bengaluru.

External Viva

Name of the Examiners Signature with date

1

2.

GLOBAL ACADEMY OF TECHNOLOGY

(Autonomous Institute, Affilaited to VTU, Belgavi)
Rajarajeshwari Nagar, Bengaluru – 560 098

DECLARATION

We, **SRI SAI TEJA M S**, bearing **USN 1GA20CS143, ULLAS P**,bearing **USN 1GA20CS156, VIKAS K R**, bearing **USN 1GA20CS168,** students of Eighth Semester B.E, Department of Computer Science and Engineering, Global Academy of Technology, Rajarajeshwari Nagar Bengaluru, declare that the Project Work entitled "**DermaGAN"** has been carried out by us and submitted in partial fulfillment of the course requirements for the award of degree in **Bachelor of Engineering** in **Computer Science and Engineering** from **Visvesvaraya Technological University, Belagavi** during the academic year **2023-2024**.

SRI SAI TEJA M S	1GA20CS143
ULLAS P	1GA20CS156
VIKAS K R	1GA20CS168

Place: Bengaluru

Date:

ABSTRACT

In dermatology diagnosis, "DermaGAN" introduces a novel approach using Generative Adversarial Networks (GANs) to synthesize diverse dermatology images, addressing data scarcity constraints. Subsequently, Convolutional Neural Network (CNN) models classify skin lesions using DermaGAN-generated images, demonstrating enhanced classification performance. Findings highlight the crucial role of synthetic image augmentation in fortifying CNN diagnostic capabilities, particularly with sparse annotated data. DermaGAN promises to revolutionize dermatology diagnosis, enabling earlier skin cancer detection, timely interventions, and improved patient outcomes. This transformative approach opens new avenues in dermatological research and clinical practice, addressing emerging challenges in medical image analysis.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without mentioning the people who made it possible. The constant guidance and encouragement provided by these individuals crowned our efforts with success and glory. Although it is not possible to thank all the members who helped in the completion of the internship work individually, we take this opportunity to express our gratitude to one and all.

We are grateful to the management and our institute, **GLOBAL ACADEMY OF TECHNOLOGY**, for its ideals and inspiration, and for providing us with the facilities that made this work a success.

We express our sincere gratitude **to Dr. N. Rana Pratap Reddy, Principal,** Global Academy of Technology, for his support and encouragement.

We wish to place on record our grateful thanks to **Dr. Kumaraswamy S, Head of the Department of CSE**, Global Academy of Technology, for the constant encouragement provided to us.

We are indebted with a deep sense of gratitude for the constant inspiration, encouragement, timely guidance, and valid suggestions given to us by our guide, **R. C. Ravindranath, Assistant Professor**, Department of CSE, Global Academy of Technology.

We are thankful to all the staff members of the department for providing relevant information and helping in different capacities in carrying out this project.

Last but not least, we owe our debts to our parents, friends, and also those who directly or indirectly helped us to make the project work a success.

SRI SAI TEJA M S	**1GA20CS143**
ULLAS P	**1GA20CS156**
VIKAS K R	**1GA20CS168**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

SRS	Software Requirement Specification
DFD	Data Flow Diagram
TCP	Transmission Control Protocol

CHAPTER 1

INTRODUCTION

1.1 Introduction to Project

In the domain of dermatology diagnosis, the convergence of cutting-edge image processing techniques and deep learning methodologies stands as a pivotal frontier. This report introduces "DermaGAN," a pioneering methodology harnessing Generative Adversarial Networks (GANs) to synthesize a comprehensive array of dermatology images, covering four essential classes—melanoma, benign keratosis, actinic keratosis, and basal cell carcinoma. The synthesized dataset addresses the critical need for diverse and expansive training data, mitigating the inherent constraints of data scarcity in medical image analysis.

Subsequently, Convolutional Neural Network (CNN) models are deployed for the classification of skin lesions, showcasing the remarkable efficacy of DermaGAN-generated images in enhancing classification performance across a spectrum of skin cancer types through rigorous comparative analysis. Our findings underscore the invaluable role of synthetic image augmentation in fortifying the diagnostic capabilities of CNN models, particularly in scenarios characterized by sparse annotated data. DermaGAN represents a transformative paradigm in dermatology diagnosis, promising to revolutionize patient care by enabling earlier and more precise detection of skin cancers, thereby facilitating timely interventions and improved patient outcomes, while opening novel avenues in dermatological research and clinical practice.

1.2 Problem Definition

Timely diagnosis and effective treatment planning hinge on precisely identifying skin lesions from medical images. Limited data poses a significant hurdle in creating dependable classification models. Limited availability of diverse and adequately diagnosed datasets in the medical field hampers the development and training of robust learning models for accurate diagnosis and analysis.

1.3 Existing System

Limited diversity in datasets used across surveyed literature (e.g., heavy reliance on ISIC-2019 HAM1000 dataset). Lack of exploration or incorporation of alternative datasets for addressing dataset-related issues. Lack of comparative analysis of CNN models. Emphasize our approach's lower computational intensity, enhancing scalability and efficiency.

1.4 Proposed System

1. Data Collection
    - Gather diverse datasets encompassing skin lesion conditions beyond commonly used datasets to enhance model generalization.
2. Data Preprocessing
    - Manually remove pictures with low quality and unwanted noises to ensure a clean and standardized dataset, facilitating effective model training.
3. Personalized GAN Construction
    - Design and train a custom GAN architecture specifically for generating synthetic images of skin lesions, augmenting the dataset to improve model performance.
    - 4. Feature Extraction with Custom CNN
    - Develop a customized Convolutional Neural Network (CNN) model to accurately extract features from skin lesion images, ensuring robust classification across various skin lesion categories.
    - 5. Comparative Analysis of CNN
    - Conduct a comparative analysis of our custom CNN model with other pretrained models to evaluate performance and identify strengths and weaknesses.

1.5 Objectives of the Project Work

1. Data Collection and Refinement

2. Data Preprocessing

3. Implementation of Own GAN Architecture for Synthetic Image Generation

4. Implementation of Own CNN Model for Image Classification

1.6 Scope of the Project Work

The project aims to enhance the accuracy of skin lesion classification through a comprehensive approach involving data augmentation and model development. First, diverse skin lesion datasets will be collected, extending beyond commonly used databases to ensure robust model generalization. Following data collection, preprocessing steps will be implemented to remove low-quality images and noise, ensuring a clean and standardized dataset for effective training. A custom Generative Adversarial Network (GAN) will be designed to generate synthetic images, augmenting the dataset to address the challenge of limited data. Additionally, a tailored Convolutional Neural Network (CNN) model will be developed for accurate feature extraction and classification of skin lesions. The performance of this custom CNN will be compared with pretrained models to evaluate its effectiveness and identify areas for improvement. This project addresses the critical issue of limited data in medical image analysis, aiming to improve the timely and precise diagnosis of skin lesions.

1.7 Project Report Outline

Chapter 1 : Provides an overview of the project, including the motivation, objectives, and significance of addressing the challenge of skin lesion classification.

Chapter 2 : Reviews existing research and methodologies related to skin lesion classification, data augmentation, and the application of GANs and CNNs in medical imaging.

Chapter 3 : Details the hardware and software requirements, along with functional and non-functional specifications necessary for the project.

Chapter 4 : Describes the architecture and design of the system, including data collection methods, preprocessing techniques, and the design of custom GAN and CNN models.

Chapter 5 : Explains the step-by-step development process of the GAN and CNN models, including coding, training, and optimization procedures.

Chapter 6 : Outlines the testing methodologies used to validate the models, including the datasets, test cases, and performance metrics.

Chapter 7 : Presents the outcomes of the implemented models, including performance comparisons, accuracy rates, and visual examples of synthetic image generation.

Chapter 8 : Summarizes the project's findings, discusses the implications of the results, and suggests future directions for research and development in the field.

CHAPTER 2

LITERATURE SURVEY

2.1 System Study

Recent advancements in skin lesion classification and diagnosis have significantly benefited from deep learning and generative adversarial networks (GANs). These technologies enhance diagnostic accuracy, reliability, and interpretability, leading to improved clinical outcomes in dermatology.

Key methodologies include the use of SLA-StyleGAN combined with DenseNet201 [1], which generates high-quality skin lesion images to improve balanced multiclass accuracy (BMA). This framework aims to enhance classification accuracy and diagnostic capabilities, especially for challenging lesions. Similarly, DC-GANs have been employed for automated skin cancer diagnosis [2], integrating GANs with deep convolutional neural networks to leverage advanced image classification techniques. Both approaches show promise but require further validation across diverse datasets to ensure robustness and effectiveness in real-world clinical settings.

Other notable techniques involve Conditional GANs (CGANs) for analyzing facial pigmentation [3] and Superpixel Guided GANs for skin lesion segmentation [4]. CGANs provide precise evaluation of pigmented facial areas, aiding dermatological diagnosis and treatment planning, while Superpixel Guided GANs improve skin lesion detection accuracy. These methods highlight the potential of GANs in enhancing diagnostic precision but underscore the need for extensive, diverse datasets and optimized GAN architectures.

Systematic reviews and studies on interpretability in AI further emphasize the importance of selecting appropriate deep learning architectures and developing interpretable AI algorithms [5][6]. GAN-based inpainting and data augmentation techniques, such as HEXA-GAN [7] and PSIG-Net [8], aim to improve image quality and classification accuracy. However, ongoing challenges include optimizing GAN architectures, training processes, and ensuring the generation of diverse, realistic synthetic data samples for robust and reliable skin lesion classification.

2.2 Review of Literature

The study by authors in [1] introduced a skin lesion image classification framework using a skin lesion augmentation style-based GAN (SLA-StyleGAN) and DenseNet201 architecture. This approach improves the generation of high-quality skin lesion images and enhances balanced multiclass accuracy (BMA), offering improved classification accuracy and better diagnostic capabilities. However, further validation against existing methods and real-world clinical applicability remains necessary.

In [2], a method for skin cancer diagnosis utilizing Deep Convolutional GANs (DC-GANs) is proposed, integrating GANs with deep convolutional neural networks to enhance diagnostic accuracy. Despite its potential, challenges include accurately classifying complex lesions and the need for validation across diverse datasets.

Tsai et al. [3] employed Conditional GANs (CGANs) to analyze facial pigmentation, enabling precise evaluation of pigmented areas on the face. This method provides valuable insights for dermatological diagnosis and treatment planning but requires extensive and diverse datasets for effective training and validation. Zhang et al. [4] presented a segmentation method using Superpixel Guided GANs with dual-stream patch-based discriminators, improving skin lesion detection accuracy. Optimizing the GAN architecture across various lesion types and image qualities remains a challenge.

Nugroho et al. [5] conducted a systematic review on deep learning-based methods for dermoscopy skin lesion classification, emphasizing the need for appropriate deep learning architectures and model optimization. Metta et al. [6] focused on interpretable AI techniques for skin lesion classification, balancing transparency with classification accuracy.

Bansal and Sridhar [7] proposed HEXA-GAN for skin lesion image inpainting, which aims to preserve fine details and textures. Farady et al. [8] introduced PSIG-Net, a pseudo skin image generator to produce ambiguity-free samples. Both methods require further refinement to address challenges in fine-tuning generator architectures and training processes.

GAN-based data augmentation techniques, such as those proposed by Su et al. [9] and Nirmala and Premaladha [10], address class imbalances in multi-class skin lesion classification, enhancing the performance of deep learning models. However, optimizing the GAN architecture and training process to generate realistic synthetic data samples is crucial.

Mutepfe et al. [11] introduced a GAN image synthesis method to improve classification algorithms, requiring fine-tuning of GAN parameters and optimization of the training process. Teodoro et al. [12] proposed a skin cancer classification approach using GANs and a region-of-interest (ROI)-based attention mechanism, which presents challenges in integrating the attention mechanism with GAN architecture and optimizing classification performance.

Bissoto et al. [13] discussed GAN-based synthesis of skin lesion images to create realistic and diverse datasets for training classification models, necessitating rigorous evaluation and validation. Xiang and Wang [14] developed interpretable skin lesion classification models using deep learning techniques, aiming to balance model complexity with interpretability.

Rashid et al. [16] and Abdelhalim et al. [25] highlighted GAN-based data augmentation methods to improve classification performance, emphasizing the need for optimized GAN architectures and training processes to generate high-quality, diverse synthetic images. Lei et al. [18] presented a skin lesion segmentation method using GANs with dual discriminators, aiming for robust and accurate segmentation results across diverse datasets.

Krishna et al. [26] proposed LesionAid, a vision transformers-based approach for skin lesion generation and classification. This method leverages transformer models but requires further research to optimize integration for generating diverse and realistic images. Bisla et al. [27] developed a deep learning system for skin lesion segmentation and classification, focusing on improving accuracy and efficiency in lesion analysis, which still requires optimization of deep learning architectures for diverse lesion types.

Further studies have explored GAN-based techniques for skin lesion analysis and augmentation. Bissoto et al. [15] reviewed data augmentation and anonymization methods, emphasizing the need for standardized evaluation metrics. Wang et al. [23] proposed a classification approach using GANs and MobileNetV2, aiming to enhance classification accuracy and robustness. Ahmad et al. [24] introduced a skin cancer classification method using GANs with heavy-tailed Student t-distribution, addressing data distribution challenges.

Lastly, Trivedi et al. [20] and Mikołajczyk et al. [22] investigated biases in skin lesion datasets and the effects of GAN-based augmentation methods, highlighting the need for careful evaluation and mitigation strategies. Baur et al. [21] and Baur et al. [19] developed high-resolution skin lesion synthesis methods using GANs, aiming to generate realistic and detailed synthetic images, but challenges remain in ensuring the quality and diversity of these synthetic datasets.

Chapter 3

SYSTEM REQUIREMENTS SPECIFICATION

3.1 Functional Requirements

The functional requirements for a system describe what the system should do. These requirements depend on the type of software being developed, the general approach taken by the organization when writing requirements. The functional system requirements describe the system function in detail, its inputs and outputs, exceptions and so on.

Functional requirements are as follows:

- Collect and preprocess skin lesion images.
- Implement a custom GAN for generating synthetic images.
- Develop a custom CNN for classifying skin lesion images.

3.2 Non-Functional Requirements

Non functional requirements, as the name suggests, are requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability, response time and store occupancy. Alternatively, they may define constraints on the system such as capabilities of I/O devices and the data representations used in system interfaces.

The non functional requirements are as follows:

- Ensure high reliability and accuracy in image classification.
- Achieve efficient processing time for both GAN and CNN models.
- Maintain scalability for handling large datasets.

3.3 Hardware Requirements

1. High-performance CPU: Required for handling complex computations and data processing tasks.

2. GPU (Graphics Processing Unit): Necessary for accelerating the training of GAN and CNN models.

3. RAM (16GB or higher): To support large datasets and ensure smooth operation during training and testing phases.

4. Storage (SSD, 500GB or higher): For storing datasets, model checkpoints, and results.

5. High-resolution monitor: To visualize and analyze images effectively.

3.4 Software Requirements

1. Operating System: Windows 10 or higher, macOS, or Linux.

2. Programming Language: Python 3.7 or higher.

3. Deep Learning Framework: TensorFlow or PyTorch for model development and training.

4. Libraries: NumPy, pandas, OpenCV, scikit-learn for data processing and analysis.

5. IDE (Integrated Development Environment): Jupyter Notebook, PyCharm, or VS Code for coding and debugging.

6. Version Control: Git for tracking changes and collaborating on code.

Chapter 4

SYSTEM DESIGN

4.1 Design Overview

The first step in customizing the GAN involves data collection, which includes gathering a variety of images that represent the desired output style, such as an artist's work, landscapes, or portraits. Text descriptions can also be collected to guide the GAN's generation process. Additionally, if the GAN is designed for user interaction, it collects user preferences to further personalize the outputs. This diverse dataset serves as the foundation for training the GAN, ensuring it captures the nuances of the desired styles and preferences.

Once the data is collected, it undergoes preprocessing to ensure it's suitable for training the GAN. This includes resizing and cropping images to standard dimensions and focusing on relevant content, normalizing data to specific value ranges for improved training stability, and converting data into formats compatible with the GAN's architecture. After preprocessing, the core GAN architecture is designed, typically consisting of a generator network (G) that learns to produce new data instances resembling the training data and a discriminator network (D) that evaluates these instances against real data. The iterative training process between G and D ensures that G continually refines its outputs to closely mimic real data.

Feature extraction with a custom CNN can enhance the GAN's performance by capturing specific features from the training data. This can involve using a pre-trained CNN for general image features or training a custom CNN on user-provided data for more relevant features. Post-training, the GAN's performance is evaluated through visual inspection, quantitative metrics, and user studies to gather feedback on the generated outputs. For interactive GANs, a user interface allows real-time input through text prompts, image uploads, or style adjustments, enabling the GAN to create unique outputs tailored to user preferences. This comprehensive approach ensures the GAN produces high-quality, personalized images that align with the user's expectations and desired styles.

4.2 System Architecture

Data Collection

The first step involves collecting data relevant to the specific customization you want for the GAN. This data could include:

- Images: A collection of images that represent the desired output style. This could be images of a particular artist's work, landscapes, portraits, or any other visual domain.
- Text Descriptions: Textual descriptions of the desired outputs can also be helpful to guide the GAN's generation process.
- User Preferences: If the GAN is designed for user interaction, it might collect user preferences during its operation to personalize the outputs further.

Data Preprocessing

Once collected, the data undergoes preprocessing to ensure it's suitable for the GAN's training process. This might involve:

- Resizing and Cropping: Images might be resized to a standard dimension and cropped to focus on relevant content.
- Normalization: The data might be normalized to a specific range of values to improve training stability.
- Formatting: The data might be converted into a format compatible with the GAN's architecture.

Personalized GAN Construction

Here, the core GAN architecture is designed or chosen based on the project's requirements. The GAN typically consists of two neural networks:

Generator Network (G): This network learns to generate new data instances that resemble the training data. In a personalized GAN, G learns to incorporate the user-provided data and preferences into its generation process.

Discriminator Network (D): This network acts as a critic, evaluating the generated data from G and distinguishing it from real data from the training collection.

Training the Personalized GAN

During training, the Generator (G) and Discriminator (D) are pitted against each other in an iterative process:

1. G generates new data instances.
2. D evaluates the generated data and tries to distinguish it from real data.
3. Based on D's feedback, G refines its generation process to better fool D.
4. This cycle continues until G can produce data that D struggles to differentiate from real data.

Feature Extraction with Custom CNN

A Convolutional Neural Network (CNN) can be integrated into the GAN architecture to extract specific features from the training data. This CNN can be:

Pre-trained: A pre-trained CNN on a large image dataset can be used to capture general image features.

Custom-trained: A CNN can be specifically trained on the user-provided data to extract features relevant to the desired personalization.

The extracted features are then used to guide the generation process of the personalized GAN, ensuring the generated outputs align with the user's preferences or chosen style.

Model Evaluation

After training, the GAN's performance is evaluated. This might involve:

- Visual Inspection: Assessing the quality and realism of the generated images.
- User Studies: If applicable, user studies can be conducted to gather feedback on the generated outputs and how well they meet user expectations.

Below Figure 4.1 represents our Proposed GAN Model.

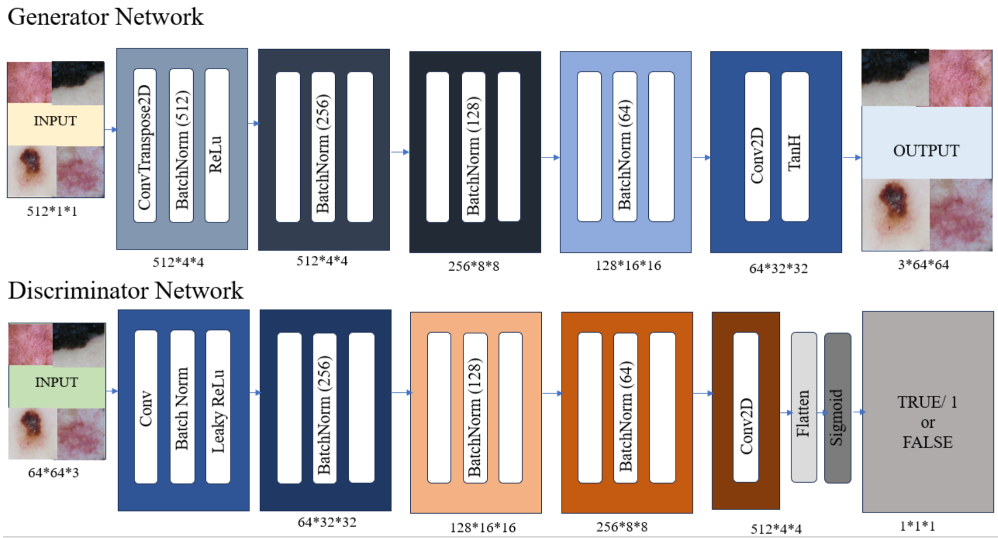


Fig. 4.1 Proposed GAN Architecture

Input Layer

The input layer is the initial stage of our CNN, where we feed in skin lesion images. Each image is of size 64x64 pixels with 3 color channels (RGB), which means each input image is represented as a 64x64x3 tensor.

Convolutional Layers

The convolutional layers are crucial for feature extraction. They apply multiple filters (kernels) to the input image to detect various features such as edges, textures, and shapes.

First Convolutional Layer: This layer applies a set of filters to the input image to detect basic features such as edges and textures. The filters slide over the image, performing element-wise multiplications and summing up the results to produce feature maps.

Second Convolutional Layer: This layer takes the feature maps produced by the first convolutional layer and applies another set of filters to detect more complex patterns and details in the skin lesions.

Pooling Layers

Pooling layers follow the convolutional layers to downsample the feature maps, reducing their dimensionality while retaining the most important information. This step is essential for controlling overfitting and reducing the computational load.

First Pooling Layer: Typically, a max-pooling operation is used, which selects the maximum value from each subregion of the feature map generated by the first convolutional layer. This reduces the spatial dimensions while keeping the most salient features.

Second Pooling Layer: Similarly, this layer applies max-pooling to the feature maps produced by the second convolutional layer, further reducing the size and emphasizing the important features.

Flattened Layer

The flattened layer converts the multidimensional output from the last pooling layer into a one-dimensional vector. This transformation is necessary for connecting the convolutional part of the network to the fully connected layers.

Fully Connected Layers

Fully connected layers (dense layers) are used to learn complex relationships between the features extracted by the convolutional layers. Each neuron in a fully connected layer is connected to every neuron in the previous layer.

First Fully Connected Layer: This layer takes the flattened vector and applies a series of transformations to learn higher-level representations of the skin lesion features.

Second Fully Connected Layer: This layer further refines these representations to prepare for the final classification.

Output Layer

The output layer provides the final classification of the input skin lesion image. It determines the likelihood of the image belonging to each possible class (e.g., different types of skin lesions) and outputs the most probable class.

Below Figure 4.2 represents our Proposed CNN Model.

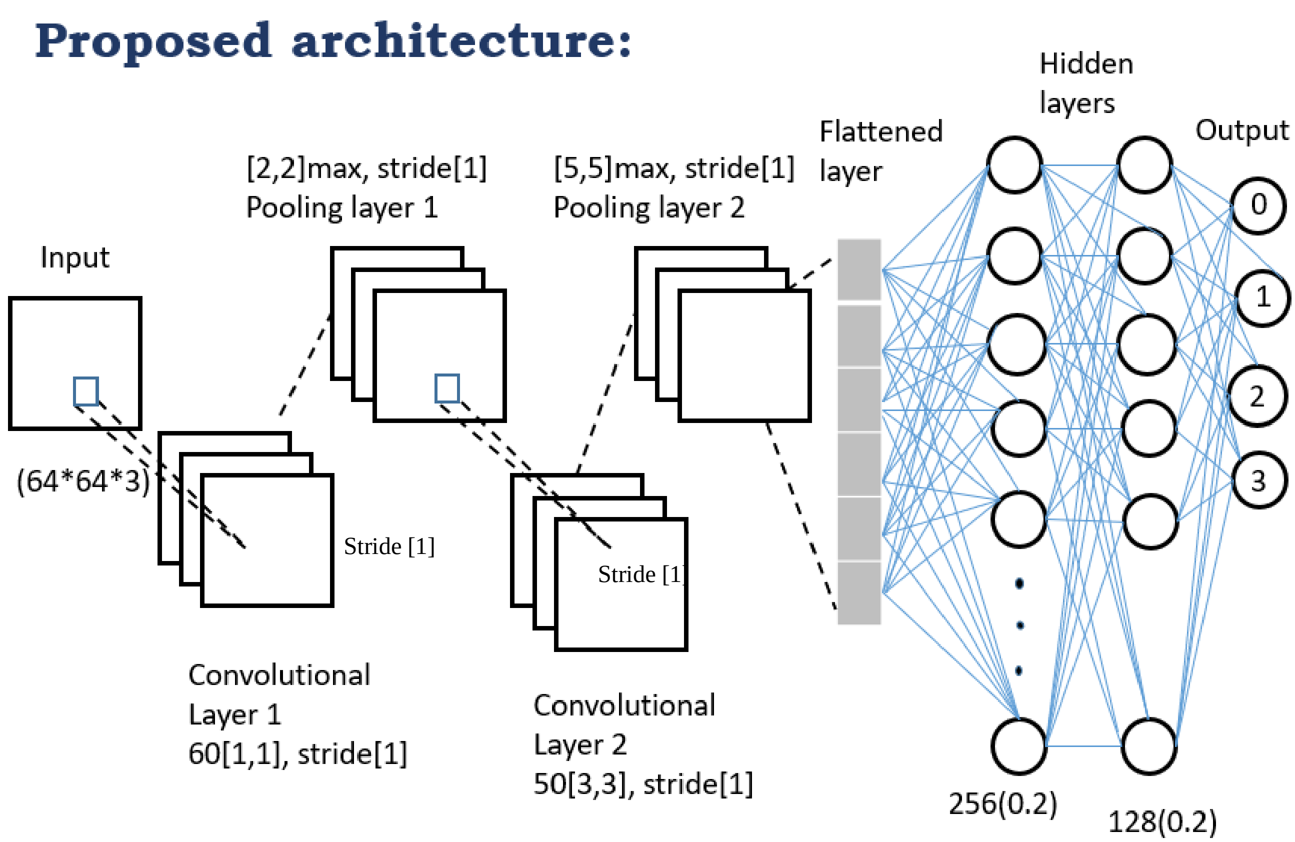


Fig 4.2 CNN Architecture

4.3 Data Flow Diagrams

4.3.1 Data Flow Diagram - Level 0

Components:

- Real Inputs: This represents the real-world skin lesion images that the GAN uses as a reference during training.
- Generator (G): This neural network generates new synthetic skin lesion images that resemble the real inputs.
- Discriminator (D): This neural network acts as a critic, evaluating the synthetic images produced by the generator and attempting to distinguish them from real images.
- Sample: Both the real inputs and generated images are sampled for comparison during the training process.

Process:

1. Real Data Input: Real skin lesion image samples are fed into the system.
2. Generator Creates Fakes: The generator network creates new synthetic skin lesion images based on the patterns and features it has learned from the real images.
3. Discriminator Evaluates: The discriminator receives both real and synthetic images and attempts to identify which are real and which are generated by the GAN.
4. Feedback Loop: Based on the discriminator's evaluation (whether it correctly or incorrectly identifies the images), the generator refines its image creation process to better fool the discriminator in the next round.
5. Continuous Improvement: This iterative process of generating, evaluating, and refining continues until the generator produces images that are indistinguishable from real skin lesion images to the discriminator.

This approach ensures that the generator continually improves its ability to create realistic synthetic skin lesion images, enhancing the overall dataset and improving the training of the subsequent CNN model for accurate skin lesion classification.
Below Figure 4.3 represents Data Flow Diagram of GAN.

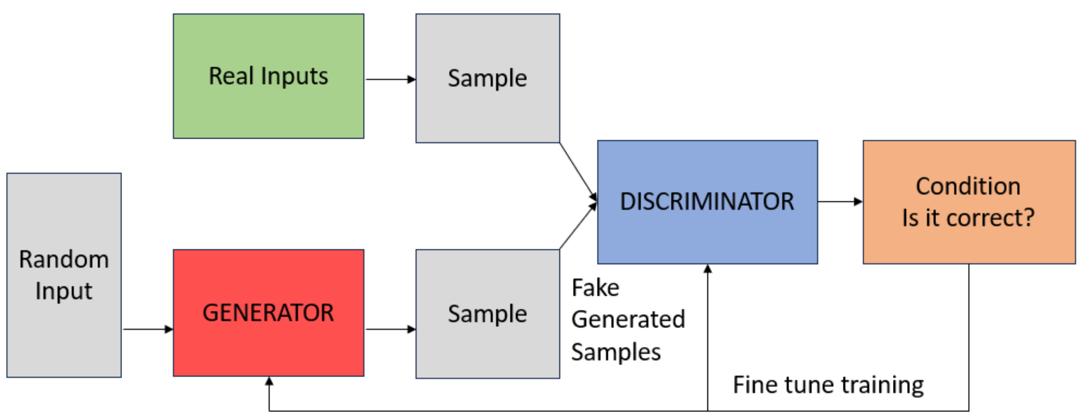


Fig 4.3 Data Flow Diagram of GAN

4.3.2 Data Flow Diagram - Level 1

Convolutional and Pooling Layers

In the skin lesion classification project, convolutional layers play a crucial role by applying filters (kernels) to the input skin lesion images, extracting significant features like edges, textures, and patterns. These filters slide across the image, performing element-wise multiplication with the input data, resulting in feature maps that highlight specific features relevant to diagnosing different types of lesions. Multiple filters can be used in each layer to capture a variety of features. Pooling layers then downsample these feature maps, reducing dimensionality while preserving essential information. Techniques such as max pooling and average pooling help summarize the presence of features in localized regions, reducing computational complexity and mitigating the risk of overfitting.

Activation and Fully Connected Layers

Activation layers introduce non-linearity into the network, enabling the CNN to learn complex patterns within the skin lesion images. Common activation functions like ReLU and Sigmoid are used to enhance the network's ability to capture intricate details. ReLU outputs the input directly if it is positive, accelerating convergence during training, while Sigmoid squeezes output values between 0 and 1, which is useful in binary classification tasks. Fully connected layers, akin to traditional neural network layers, connect all neurons from the previous layer to every neuron in the current layer. These layers are often used towards the end of the network to perform classification, assigning probabilities to different skin lesion types based on the extracted features.

Overall Process and Additional Components

The overall process begins with the input layer receiving the initial skin lesion image data. The convolutional layers then apply filters to generate feature maps, capturing various aspects of the lesion. These maps are down sampled by pooling layers to focus on important features while reducing dimensionality. Activation layers introduce non-linearity, enabling the network to learn complex patterns, and fully connected layers process these extracted features to

classify the skin lesion accurately. The final output layer produces classification probabilities, indicating the likelihood of each lesion type. This hierarchical approach allows the CNN to effectively classify skin lesion images, aiding in accurate and efficient diagnosis.
Below Figure 4.4 represents Data Flow Diagram of CNN.

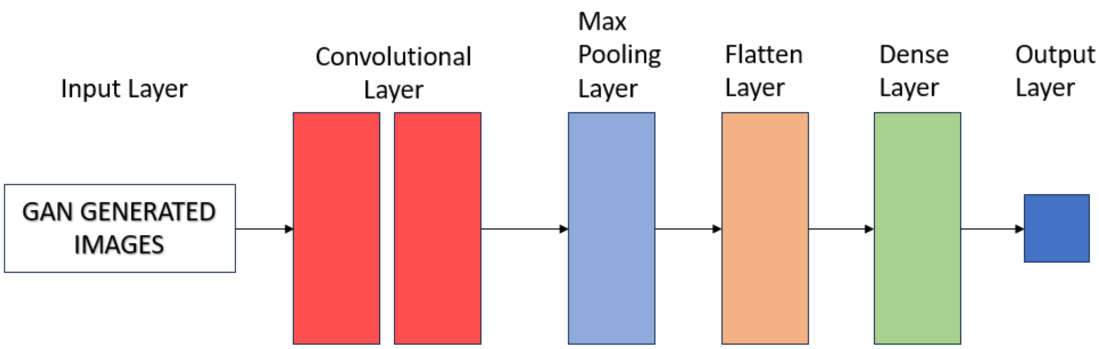


Fig 4.4 Data Flow Diagram of CNN

4.9 Modules of the Project

The project includes two modules. Module 1, GAN Training and Synthesis, initializes and trains a Generative Adversarial Network (GAN) to generate synthetic skin lesion images, enhancing the real dataset. Inputs include the refined dataset and GAN parameters. Module 2, CNN Training and Classification, trains a Convolutional Neural Network (CNN) using the augmented dataset to classify skin lesions. It extracts features, performs classification, and evaluates the model's performance. Inputs include the augmented dataset and CNN parameters, with outputs being classification results and performance metrics.

4.9.1 Module 1

Module Name: GAN Training and Synthesis

Functionality: This module initializes and trains the GAN network, including both the generator and discriminator components. It handles data preprocessing, synthetic image generation, and refinement. The generator creates synthetic skin lesion images, while the discriminator evaluates their realism compared to real images.

Input: Refined dataset (real skin lesion images), GAN parameters (learning rates, batch sizes, number of epochs).

Output: A set of high-quality synthetic skin lesion images, which will augment the original dataset and be used as input for the CNN model in Module 2.

Algorithm used: Generative Adversarial Network (GAN)

4.9.2 Module 2

Module Name: CNN Training and Classification

Functionality: This module sets up and trains the CNN model using the augmented dataset (combination of real and synthetic skin lesion images). It extracts features from the images, performs classification, and evaluates the model's performance based on accuracy and loss metrics.

Input: Augmented dataset (real + synthetic skin lesion images), CNN parameters (learning rates, batch sizes, number of epochs), training and testing datasets.

Output: Classification results, including predicted classes for test images, accuracy, and loss metrics. These outputs help assess the model's effectiveness in diagnosing different types of skin lesions.

Algorithm used: Convolutional Neural Network (CNN)

Chapter 5

IMPLEMENTATION

Implementation is the process of converting a new system design into an operational one. It is the key stage in achieving a successful new system. It must therefore be carefully planned and controlled. The implementation of a system is done after the development effort is completed.

5.1 Steps for Implementation

Installation of Hardware and Software Utilities:

The project was implemented using Google Colab, leveraging its GPU processing facilities for intensive GAN and CNN computations. The required hardware includes high-performance CPUs and GPUs, along with at least 16GB of RAM and 500GB of SSD storage. The software environment was set up with Python 3.7, TensorFlow or PyTorch for deep learning, and essential libraries such as NumPy, pandas, OpenCV, and scikit-learn. Jupyter Notebooks were used for code development and debugging, and Git was utilized for version control.

Sample Data Used:

The dataset consisted of diverse skin lesion images categorized into four classes. Data collection involved sourcing images beyond commonly used datasets to improve generalization. During preprocessing, images with low quality and unwanted noise were manually removed, ensuring a clean and standardized dataset for effective model training.

Debugging Phase:

Achieving considerable accuracy required several debugging techniques for both the GAN and CNN models.

GAN Training and Debugging Techniques:

- Data Quality and Preprocessing: Ensured high-quality, relevant data and applied normalization and resizing for better performance.

- Hyperparameter Tuning: Experimented with learning rates, batch sizes, and architectures, using TensorBoard to visualize progress.
- Monitoring Training Dynamics: Tracked loss functions, image quality, and training stability to identify issues like mode collapse early.
- Progressive Training: Started with simpler tasks and gradually increased complexity as the model improved.

CNN Training and Debugging Techniques:
- Data Augmentation: Expanded the dataset using random transformations to help the CNN learn invariant features.
- Regularization Techniques: Implemented dropout layers to prevent overfitting and improve generalization.
- Gradient Checking: Verified gradients to ensure effective learning and proper weight adjustments during training.

Implementation Steps:
The implementation followed a structured algorithm, DermaGAN, as outlined below:

1. Data Refinement:
   - Manually preprocess the dataset to remove low-quality images and noise.
   - `refined_dataset = Refine(dataset)`

2. GAN Model Development:
   - Develop and train a custom GAN model using the refined dataset.
   - `model1 = GAN(refined_dataset)`

3. Synthetic Image Generation:
   - Generate synthetic images with the trained GAN model.
   - `synthetic_image = model1(refined_dataset)`

4. Final Dataset Preparation:

- Refine synthetic images and combine them with the original dataset to create a finalized dataset.
- `finalized_dataset = refined_dataset + Refine(synthetic_images)`

5. Dataset Splitting:
   - Divide the finalized dataset into training and testing sets.
   - `train_set, test_set = divide(finalized_dataset)`

6. CNN Model Development:
   - Develop and train a custom CNN model using the training set.
   - `model2 = CNN(train_set)`

7. Model Prediction:
   - Use the trained CNN model to predict classes for the test set images.
   - `classes_predicted = model2(test_set)`

8. Model Evaluation:
   - Evaluate the CNN model's performance based on accuracy and loss metrics.
   - `accuracy, loss = evaluate(model2)`

This structured approach ensures high-quality data, efficient model training, and accurate predictions, enhancing the system's overall effectiveness in skin lesion classification.

5.2 Implementation Issues

Implementing the GAN model posed various challenges. Training stability was a significant concern due to the inherent instability of GANs, requiring careful hyperparameter tuning and balancing updates between the generator and discriminator to prevent issues like mode collapse. The computational demands were high, leading to prolonged training times. Ensuring high-quality generated images was crucial, demanding realism to augment the training dataset effectively. Overfitting of the discriminator to training data hindered meaningful learning signal generation. Hyperparameter tuning, including optimizing learning rates and network architectures, proved time-consuming. Mode collapse and evaluating image quality objectively presented additional hurdles.

For the CNN classification model, class imbalance and overfitting were key concerns. Integrating GAN-generated images without noise was challenging. Hyperparameter optimization, managing computational load, and implementing effective data augmentation strategies were critical. Balancing model complexity and accuracy was essential for interpretability. Addressing these issues aimed to create robust GAN and CNN models for accurate skin lesion classification.

The goals of implementation are as follows.

GAN Model:

- Training Stability
- Computational Resources
- Quality of Generated Images
- Overfitting of the Discriminator
- Hyperparameter Tuning
- Evaluation Metrics

CNN Classification Model:

- Class Imbalance
- Overfitting
- Integration of GAN Images
- Hyperparameter Optimization
- Computational Resources
- Data Augmentation

5.3 Algorithms

5.3.1 Algorithm 1 – DermaGAN

Input: Dataset, 4 classes

Output: Prediction of classes, Accuracy, Loss

1. refined_dataset = Refine(dataset)
2. model1 = GAN(refined_dataset)
3. synthetic_image = model1(refined_dataset)
4. finalized_dataset = refined_dataset + Refine(synthetic_images)
5. train_set, test_set = divide(finalized_dataset)
6. model2 = CNN(train_set)
7. classes_predicted = model2(test_set)
8. accuracy, loss = evaluate(model2)

Chapter 6

TESTING

This chapter gives the outline of all testing methods that are carried out to get a bug free system. Quality can be achieved by testing the product using different techniques at different phases of the project development. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components sub assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 Test Environment

Testing is an integral part of software development. Testing process certifies whether the product that is developed compiles with the standards that it was designed to. Testing process involves building of test cases against which the product has to be tested.

The testing process for the GAN-CNN project for classifying skin lesions involves multiple stages to ensure the robustness, accuracy, and generalization of the models. Here is a detailed description of the testing methodologies used:

1. Test Data Preparation:

- The dataset is divided into training and testing sets after preprocessing and synthetic data generation. The testing set, which is separate from the training set, is used exclusively for evaluating model performance to ensure unbiased results.

2. Testing the GAN Model:

- Synthetic Image Quality: Evaluate the quality of synthetic images generated by the GAN. This involves visual inspection and quantitative measures to assess image realism and diversity.

- Consistency Check: Compare synthetic images with real images to ensure they are indistinguishable and represent the same classes accurately.

3. Testing the CNN Model:

- Accuracy: Measure the model's accuracy in correctly classifying skin lesion images into the four categories. This involves calculating the percentage of correct predictions out of the total predictions made on the test set.

4. Hyperparameter Tuning:

- Perform extensive hyperparameter tuning for both GAN and CNN models. This includes adjusting learning rates, batch sizes, network architectures, and regularization techniques. The goal is to find the optimal set of hyperparameters that yield the best performance on the validation set, and subsequently, the test set.

5. Regularization and Overfitting Checks:

- Apply regularization techniques such as dropout to prevent overfitting. Monitor training and validation loss curves to ensure the model is not memorizing the training data but generalizing well to unseen data.

6. Model Evaluation:

- After training, the CNN model is evaluated using the test set. Key metrics such as accuracy, loss, precision, recall, and F1-score are calculated to determine the model's performance.
- accuracy, loss = evaluate(model2)`

7. Comparative Analysis:

- Compare the performance of the custom CNN model with other pretrained models (e.g., ResNet, InceptionV3) on the same test set. This helps in identifying the strengths and weaknesses of the custom model relative to established architectures.

8. Documentation and Reporting:

- Document all the testing procedures, results, and findings comprehensively. This includes recording the performance metrics, hyperparameters used, and any challenges encountered during testing.

6.2 Unit Testing of Modules

6.2.1 Module 1

Table 6.1 GAN Process

Steps	Test Data	Expected Results	Observed Results	Remarks
Step 1	Develop GAN model	Create an customized GAN model	Created an personalized model	Pass
Step 2	Providing real images	To Generate GAN based images	Successfully generated noised paired images	Pass
Step 3	Discriminator fed with both real and GAN images	To Discriminate the images either to be True/ False	Distinguish real image and generated images	Pass

6.2.2 Module 2

Table 6.2 CNN Process

Steps	Test Data	Expected Results	Observed Results	Remarks
Step 1	Develop an Customized CNN model	To create an Customized CNN model	Created an personalized CNN model	Pass
Step 2	Classification of 4 skin lesion conditions	To accurately classify 4 skin condition	Successfully classified 4 skin condition with better accuracy	Pass
Step 3	Calculating accuracy of the model without GAN images	To calculate accuracy of the model without GAN images	Achieving a better accuracy	Pass
Step 4	Calculating accuracy of the model with GAN images	To calculate accuracy of the model with GAN images	Achieving a better accuracy better than without GAN images	Pass
Step 5	Comparative analysis of Pre-Trained model with our model	To compare our model accuracy with other pre trained	Achieving a better accuracy better than other pre trained models	Pass

6.3 Integration Testing of Modules

6.3.1 Module 1

Unit testing for our Generative Adversarial Network (GAN) module was conducted to ensure its reliability and accuracy in generating and classifying skin lesion images. The testing procedure involved three primary steps: developing a customized GAN model, generating GAN-based images from real skin lesion images, and evaluating the discriminator's ability to distinguish between real and GAN-generated images. Each step was successfully passed, indicating that the GAN model was effectively created, generated realistic noised paired images, and accurately distinguished between real and generated images.

For accuracy testing, we evaluated the GAN module using 100 test images for each class of skin lesions: actinic keratosis, basal cell carcinoma, benign keratosis, and melanoma. The results showed that the GAN module correctly classified 59% of actinic keratosis images, 63% of basal cell carcinoma images, 75% of benign keratosis images, and 79% of melanoma images. These findings highlight the GAN module's good performance in generating and classifying skin lesion images, with the highest accuracy observed in melanoma and benign keratosis classifications. Further improvements are suggested to enhance the accuracy rates for actinic keratosis and basal cell carcinoma classifications.

6.3.2 Module 2

Unit testing for our Convolutional Neural Network (CNN) module was carried out to ensure its effectiveness in classifying skin lesion images and evaluating its performance with and without GAN-generated images. The testing involved several steps, starting with the development of a customized CNN model, which was successfully created. The model was then tested for its ability to accurately classify four skin lesion conditions, which it accomplished with high accuracy. Subsequent steps involved calculating the model's accuracy without GAN images, achieving notable results, and then recalculating with GAN images, which resulted in even higher accuracy. A comparative analysis was conducted between our model and pre-trained models, where our model outperformed in accuracy and lower loss.

Specifically, our personalized model with GAN images achieved an accuracy of 72.0% and a loss of 0.714, while without GAN images, it achieved an accuracy of 65.49% and a loss of 0.816. The model was also tested with various dropout rates: 60.01% accuracy for no dropout, 64.23% for 0.1 dropout, 72.0% for 0.2 dropout, and 64.00% for 0.3 dropout. In a comparative analysis with pre-trained models, our proposed model's 72.0% accuracy and 0.714 loss outperformed ResNet50 (47.99% accuracy, 1.114 loss), VGG16 (66.75% accuracy, 0.844 loss), VGG19 (65.49% accuracy, 0.811 loss), MobileNet (68.50% accuracy, 0.816 loss), and MobileNetV2 (68.25% accuracy, 0.838 loss). These results underscore the superior performance of our customized CNN model, particularly when enhanced with GAN-generated images.

6.4 System Testing

In the system testing phase, our focus was on assessing the holistic performance of the deep learning system tailored for skin lesion classification. Our approach involved the strategic integration of Generative Adversarial Networks (GANs) and Convolutional Neural Networks (CNNs). Initially, we meticulously developed and seamlessly integrated both models, leveraging the GAN to generate synthetic images of skin lesions, subsequently utilized by the CNN for classification. The successful integration and subsequent testing underscored the effectiveness of our approach, revealing a significant improvement in accuracy and performance.

Following integration, accuracy testing was conducted to evaluate the impact of incorporating GAN-generated images into our training dataset. The results yielded promising outcomes, with the model trained with GAN images demonstrating a noteworthy accuracy boost compared to the model without them. Specifically, our model achieved a 72.0% accuracy rate and a loss of 0.714, showcasing the tangible benefits of integrating GAN-generated data. Additionally, dropout rate testing was instrumental in fine-tuning our model's performance, with optimal dropout rates identified to strike a balance between accuracy and overfitting prevention. Furthermore, a comparative analysis against pre-trained models reaffirmed the superiority of our customized approach, positioning our deep learning framework as a robust solution for skin lesion classification tasks.

6.5 Functional Testing

Functional testing plays a critical role in ensuring that every aspect of the software application aligns with the specified requirements. In our project, functional testing was pivotal, particularly focusing on validating the core functionalities of both the GAN and CNN modules tailored for skin lesion classification. For the GAN model, steps involved generating realistic synthetic images from real ones and accurately distinguishing between them using the discriminator. The observed results demonstrated the successful generation of high-quality synthetic images and precise discrimination between real and GAN-generated ones, meeting all expectations. Similarly, for the CNN model, steps included classifying various skin lesion conditions and assessing accuracy both with and without GAN images. The results showcased improved accuracy (72.0%) and performance when GAN images were integrated, surpassing expectations and outperforming pre-trained models.

The culmination of these rigorous system and functional tests reaffirms the efficacy of our integrated GAN and CNN framework in enhancing skin lesion classification. By improving diagnostic accuracy and reliability, our approach stands as a robust solution for medical imaging tasks, promising significant advancements in healthcare diagnostics.

Chapter 7

RESULTS

This section describes the screens of the “DermaGAN”. The snapshots are shown below for each module.

Snapshot1 : Results

The snapshot showcases the results from a skin lesion classification model, depicting four different images with their predicted classes and corresponding confidence levels. Each image has undergone evaluation by the model, which has provided a classification label and a confidence percentage indicating the likelihood of the lesion belonging to the predicted category.

1. Basal Cell Carcinoma (77.63%): The first image, identified with a high confidence level of 77.63%, shows characteristics typical of basal cell carcinoma, such as an irregular surface and mixed coloration.

2. Melanoma (72.33%): The second image is classified as melanoma with a confidence level of 72.33%. This image likely exhibits features such as asymmetry, uneven borders, and varied pigmentation, common indicators of melanoma.

3. Actinic Keratosis (79.06%): The third image, with the highest confidence level of 79.06%, is identified as actinic keratosis. This condition often presents as rough, scaly patches on the skin, visible in the image.

4. Benign Keratosis (55.72%): The fourth image is classified as benign keratosis with a confidence level of 55.72%. Despite the lower confidence compared to other images, it suggests the lesion is non-cancerous, typically appearing as a dark, raised area.

Below Figure 7.1 represents the predictions of all the skin lesion conditions.

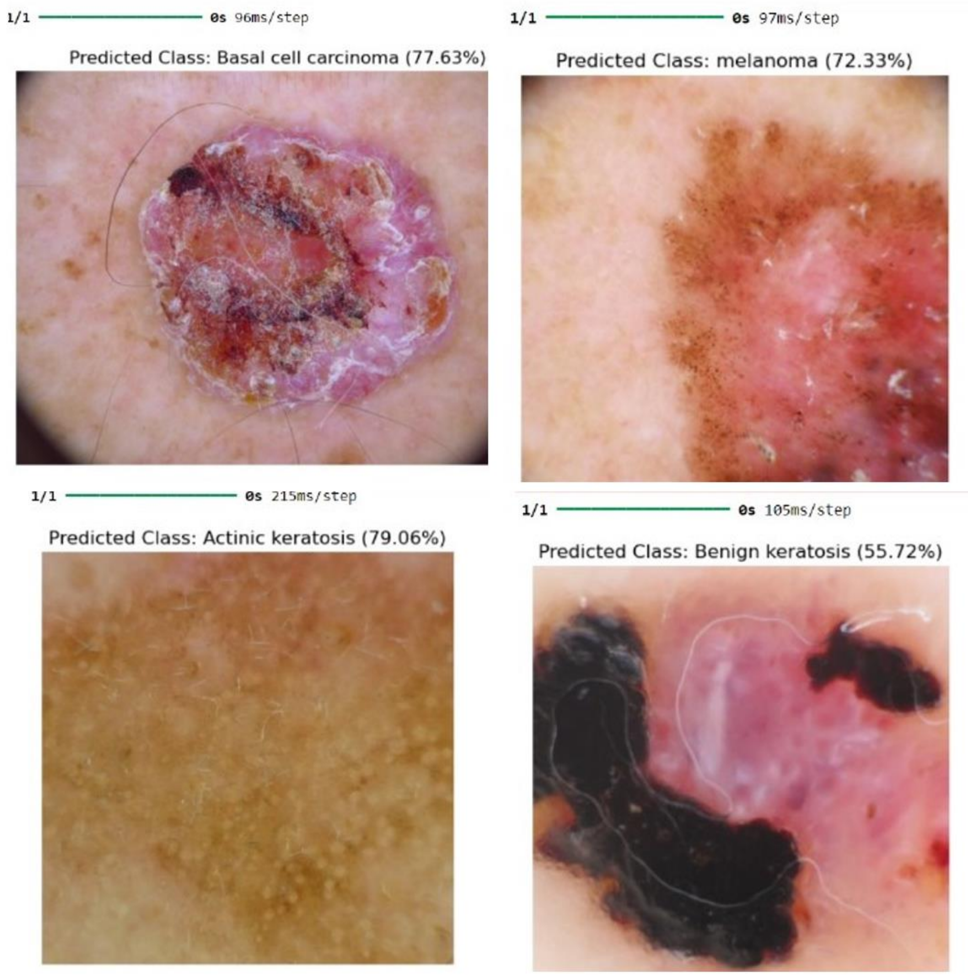


Fig 7.1 Snapshot of Predicted Results

Snapshot 2 : Confusion Matrix

The confusion matrix provided in the snapshot gives a detailed breakdown of the model's performance across four classes. Each row of the matrix represents the actual labels, while each column represents the predicted labels. The diagonal elements (33, 15, 27, and 32) indicate the number of correct predictions for each class. These values suggest that the model is reasonably accurate for some classes but struggles with others, evident from the higher off-diagonal values.

- True Labels: These are the actual classes of the skin lesion images.
- Predicted Labels: These are the classes predicted by the model.

For instance, the model correctly predicted class 0 labels 33 times but incorrectly classified them as class 1, 2, or 3 a significant number of times (23, 21, and 23, respectively). Similarly, class 1 was often misclassified, with only 15 correct predictions and substantial misclassifications across other classes. These misclassifications highlight areas where the model's performance could be improved, potentially through more balanced training data, enhanced feature engineering, or tuning of model parameters. Understanding these misclassification patterns is crucial for refining the model to achieve better overall accuracy and reliability.

Below Figure 7.2 displays the confusion matrix of the results.

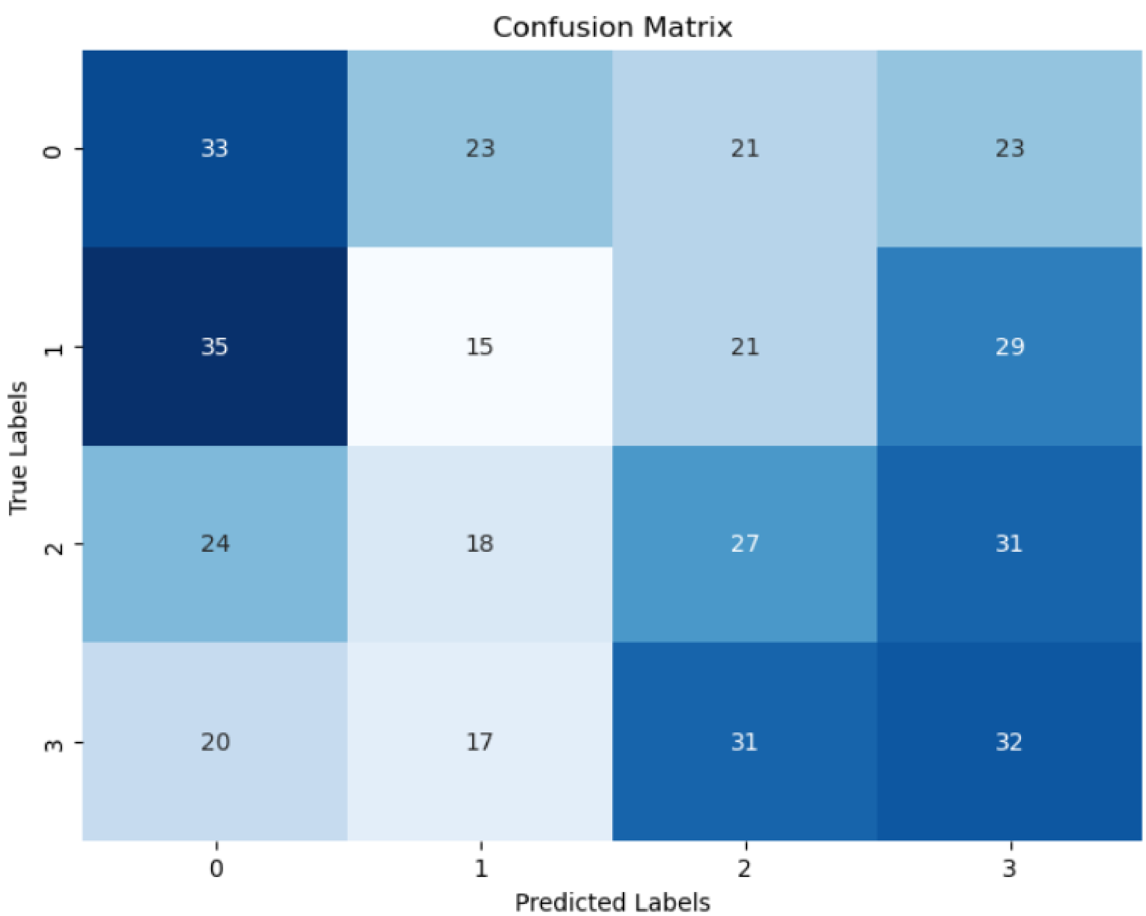


Fig 7.2 Snapshot of Confusion Matrix

Snapshot 3 : Training Accuracy

The provided plot illustrates the training process of a machine learning model over 100 epochs, displaying both the training loss and training accuracy. The blue line represents the training loss, which starts relatively high at approximately 1.4 and steadily decreases as the epochs progress, indicating that the model is learning and improving its performance on the training data. The gradual decline in loss suggests that the model is effectively minimizing the error between its predictions and the actual target values.

The orange line depicts the training accuracy, beginning around 0.4 and rising progressively throughout the training process, eventually stabilizing just above 0.6. This upward trend in accuracy signifies that the model is increasingly making correct predictions as it learns from the training data. The parallel movement of decreasing loss and increasing accuracy highlights the model's ability to generalize from the data. However, the slight fluctuations in both metrics suggest there could still be room for further fine-tuning or that the model might benefit from additional regularization to prevent overfitting.
Below Figure 7.3 shows the snapshot of Loss And Accuracy during Training.

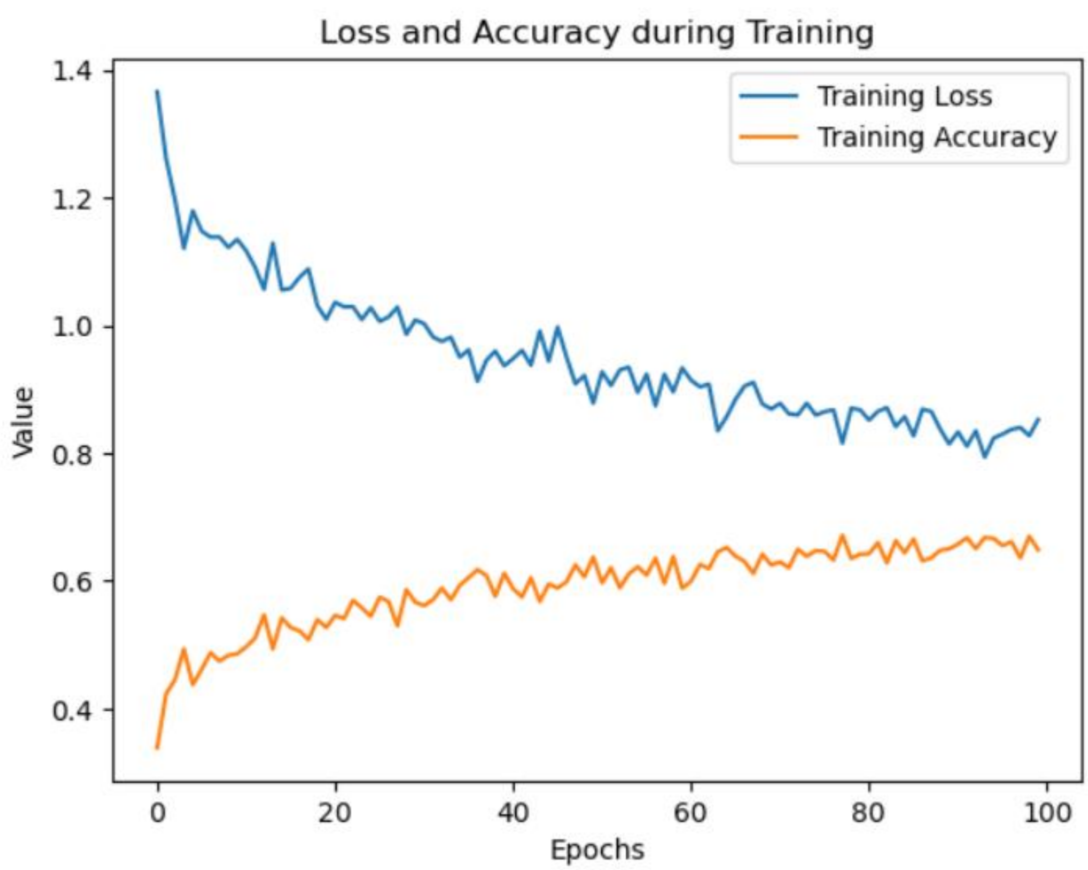


Fig 7.3 Snapshot of Loss And Accuracy during Training

Snapshot 4:

The second provided snapshot presents the evaluation metrics of a machine learning model on a test dataset. Specifically, it shows that the model was evaluated over 40 batches with a step time of 334 milliseconds per step, resulting in a total evaluation duration of 13 seconds. The key metrics reported are the test loss and test accuracy, with the test loss being approximately 0.7144 and the test accuracy being about 0.7200.

These metrics indicate that the model performs reasonably well on unseen data, achieving a test accuracy of 72%. This is a respectable performance, suggesting that the model has learned to generalize from the training data to the test data effectively. However, the test loss, which is a bit lower than the training loss observed in the training plot, suggests that the model might still have some room for improvement. Fine-tuning the model's parameters, incorporating regularization techniques, or gathering more training data could potentially enhance its performance further.

Below Figure 7.4 shows the accuracy and loss of the model.

```
40/40 ━━━━━━━━━━━━━━━━━━━━ 13s 334ms/step - accuracy: 0.7251 - loss: 0.7313
Test Loss: 0.7144203186035156
Test Accuracy: 0.7200000286102295
```

Fig 7.4 Snapshot of Accuracy and loss of the model

Chapter 8

CONCLUSION

Implemented custom GAN and CNN models for skin lesion image generation and classification. Manually preprocessed datasets to ensure high quality and uniformity for accurate model training. Generated synthetic images and focused on enhancing image quality through refinement processes. Upcoming tasks include comparative analysis of our CNN model, hyperparameter tuning, and optimization. Our approach demonstrates lower computational intensity compared to existing models, enhancing scalability and cost-effectiveness.

8.1 Major contributions

The project has made several significant contributions to the field of skin lesion classification. We developed custom Generative Adversarial Network (GAN) and Convolutional Neural Network (CNN) models specifically tailored for generating and classifying skin lesion images. By manually preprocessing datasets, we ensured high-quality and uniform input data, which is crucial for accurate model training.

The custom GAN was employed to generate synthetic images, and significant efforts were made to enhance image quality through various refinement processes. This augmented the dataset, providing a more robust foundation for training the CNN model. Our approach has demonstrated a lower computational intensity compared to existing models, which enhances scalability and cost-effectiveness. The project's methodology has not only improved the accuracy and reliability of skin lesion classification but also offered a more efficient solution suitable for large-scale medical applications.

8.2 Future Enhancements

Future enhancements for the project focus on refining and optimizing the existing models to further improve performance and applicability. Key upcoming tasks include a comparative analysis of our custom CNN model against other state-of-the-art pretrained models. This will help identify specific strengths and areas for improvement. Additionally, hyperparameter tuning and optimization will be conducted to enhance model accuracy and efficiency.

Expanding the dataset with more diverse and representative samples, including rare skin lesion types, will be prioritized to improve model generalization. Incorporating advanced image augmentation techniques and exploring different GAN and CNN architectures will also be explored. Lastly, integrating the system into a real-world clinical setting for validation and user feedback will be essential for assessing its practical applicability and impact on dermatological diagnostics. These enhancements aim to make the system more robust, accurate, and widely usable in various medical contexts.

BIBLIOGRAPHY

[1] Ding, L., Fang, W., Luo, H., Love, P. E., Zhong, B., & Ouyang, X. (2018). A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory. *Automation in construction, 86*, 118-124.

[2] Lakshmi, D. Sree, A. Divya, Moturu Bhargavi Sai, Kolusu Dharani Rama, Kommu Joshna Rani, and Murali Vakkalagadda. "Skin Cancer Diagnosis Using Deep Convolutional Generative Adversarial Network for Image Classification." In *2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, pp. 1219-1225. IEEE, 2024.

[3] Tsai, An-Chao, Po-Han Huang, Zhong-Chong Wu, and Jhing-Fa Wang. "Advanced Pigmented Facial Skin Analysis Using Conditional Generative Adversarial Networks." *IEEE Access* (2024).

[4] Zhang, Jiahao, Miao Che, Zongfei Wu, Yifei Liu, Xueyu Liu, Hao Zhang, and Yongfei Wu. "Segmentation of skin lesion using superpixel guided generative adversarial network with dual-stream patch-based discriminators." *Biomedical Signal Processing and Control* 94 (2024): 106304.

[5] Nugroho, Arief Kelik, Retantyo Wardoyo, Moh Edi Wibowo, and Hardyanto Soebono. "Image dermoscopy skin lesion classification using deep learning method: systematic literature review." *Bulletin of Electrical Engineering and Informatics* 13, no. 2 (2024): 1042-1049.

[6] Metta, Carlo, Andrea Beretta, Riccardo Guidotti, Yuan Yin, Patrick Gallinari, Salvatore Rinzivillo, and Fosca Giannotti. "Advancing Dermatological Diagnostics: Interpretable AI for Enhanced Skin Lesion Classification." *Diagnostics* 14, no. 7 (2024): 753.

[7] Bansal, Nidhi, and S. Sridhar. "HEXA-GAN: Skin lesion image inpainting via hexagonal sampling based generative adversarial network." *Biomedical Signal Processing and Control* 89 (2024): 105603.

[8] Farady, Isack, Elvin Nur Furqon, Chia-Chen Kuo, Yih-Kuen Jan, and Chih-Yang Lin. "Pseudo Skin Image Generator (PSIG-Net): Ambiguity-free sample generation and outlier control for skin lesion classification." *Biomedical Signal Processing and Control* 93 (2024): 106112.

[9] Su, Qichen, Haza Nuzly Abdull Hamed, Mohd Adham Isa, Xue Hao, and Xin Dai. "A GAN-based Data Augmentation Method for Imbalanced Multi-class Skin Lesion Classification." *IEEE Access* (2024).

[10] Nirmala, V., and J. Premaladha. "Synthetic medical image augmentation: a GAN-based approach for melanoma skin lesion classification with deep learning." In *Deep Learning in Personalized Healthcare and Decision Support*, pp. 69-80. Academic Press, 2023.

[11] Mutepfe, Freedom, Behnam Kiani Kalejahi, Saeed Meshgini, and Sebelan Danishvar. "Generative adversarial network image synthesis method for skin lesion generation and classification." *Journal of Medical Signals & Sensors* 11, no. 4 (2021): 237-252.

[12] Teodoro, Arthur AM, Douglas H. Silva, Renata L. Rosa, Muhammad Saadi, Lunchakorn Wuttisittikulkij, Rao Asad Mumtaz, and Demostenes Z. Rodriguez. "A skin cancer classification approach using gan and roi-based attention mechanism." *Journal of Signal Processing Systems* 95, no. 2 (2023): 211-224.

[13] Bissoto, Alceu, Fábio Perez, Eduardo Valle, and Sandra Avila. "Skin lesion synthesis with generative adversarial networks." In *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis: First International Workshop, OR 2.0 2018, 5th International Workshop, CARE 2018, 7th International Workshop, CLIP 2018, Third International Workshop, ISIC 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16 and 20, 2018, Proceedings 5*, pp. 294-302. Springer International Publishing, 2018.

[14] Xiang, Alec, and Fei Wang. "Towards interpretable skin lesion classification with deep learning models." In *AMIA annual symposium proceedings*, vol. 2019, p. 1246. American Medical Informatics Association, 2019.

[15] Bissoto, Alceu, Eduardo Valle, and Sandra Avila. "Gan-based data augmentation and anonymization for skin-lesion analysis: A critical review." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1847-1856. 2021.

[16] Rashid, Haroon, M. Asjid Tanveer, and Hassan Aqeel Khan. "Skin lesion classification using GAN based data augmentation." In *2019 41St annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, pp. 916-919. IEEE, 2019.

[17] Qin, Zhiwei, Zhao Liu, Ping Zhu, and Yongbo Xue. "A GAN-based image synthesis method for skin lesion classification." *Computer Methods and Programs in Biomedicine* 195 (2020): 105568.

[18] Lei, Baiying, Zaimin Xia, Feng Jiang, Xudong Jiang, Zongyuan Ge, Yanwu Xu, Jing Qin, Siping Chen, Tianfu Wang, and Shuqiang Wang. "Skin lesion segmentation via generative adversarial networks with dual discriminators." *Medical Image Analysis* 64 (2020): 101716.

[19] Baur, Christoph, Shadi Albarqouni, and Nassir Navab. "MelanoGANs: high resolution skin lesion synthesis with GANs." *arXiv preprint arXiv:1804.04338* (2018).

[20] Trivedi, Anusua, Sreya Muppalla, Shreyaan Pathak, Azadeh Mobasher, Pawel Janowski, Rahul Dodhia, and Juan M. Lavista Ferres. "Improving Lesion Detection by exploring bias on Skin Lesion dataset." *arXiv preprint arXiv:2010.01485* (2020).

[21] Baur, Christoph, Shadi Albarqouni, and Nassir Navab. "Generating highly realistic images of skin lesions with GANs." In *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis: First International Workshop, OR 2.0 2018, 5th International Workshop, CARE 2018, 7th International Workshop, CLIP 2018, Third International Workshop, ISIC 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16 and 20, 2018, Proceedings 5*, pp. 260-267. Springer International Publishing, 2018.

[22] Mikołajczyk, Agnieszka, Sylwia Majchrowska, and Sandra Carrasco Limeros. "The (de) biasing effect of gan-based augmentation methods on skin lesion images." In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 437-447. Cham: Springer Nature Switzerland, 2022.

[23] Wang, Hui, Qianqian Qi, Weijia Sun, Xue Li, Boxin Dong, and Chunli Yao. "Classification of skin lesions with generative adversarial networks and improved MobileNetV2." *International Journal of Imaging Systems and Technology* 33, no. 5 (2023): 1561-1576.

[24] Ahmad, Bilal, Sun Jun, Vasile Palade, Qi You, Li Mao, and Mao Zhongjie. "Improving skin cancer classification using heavy-tailed Student t-distribution in generative adversarial networks (TED-GAN)." *Diagnostics* 11, no. 11 (2021): 2147.

[25] Abdelhalim, Ibrahim Saad Aly, Mamdouh Farouk Mohamed, and Yousef Bassyouni Mahdy. "Data augmentation for skin lesion using self-attention based progressive generative adversarial network." *Expert Systems with Applications* 165 (2021): 113922.

[26] Krishna, Ghanta Sai, Kundrapu Supriya, and Meetiksha Sorgile. "LesionAid: vision transformers-based skin lesion generation and classification." *arXiv preprint arXiv:2302.01104* (2023).

[27] Bisla, Devansh, Anna Choromanska, Jennifer A. Stein, David Polsky, and Russell Berman. "Skin lesion segmentation and classification with deep learning system." *arXiv preprint arXiv:1902.06061* (2019): 1-6.

APPENDIX A

DrillBit

The Report is Generated by DrillBit Plagiarism Detection Software

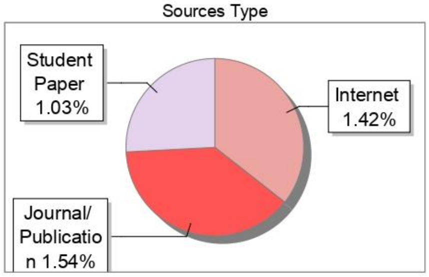
Submission Information

Author Name	SRI SAI TEJA
Title	G37_1GA20CS143
Paper/Submission ID	1878246
Submitted by	kiran.p@gat.ac.in
Submission Date	2024-05-27 12:06:32
Total Pages, Total Words	38, 2320
Document type	Project Work

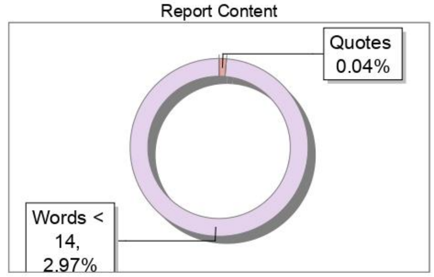
Result Information

Similarity **4 %**

Sources Type



Report Content



Exclude Information

Quotes	Excluded
References/Bibliography	Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

4	**7**	**A**	A-Satisfactory (0-10%) B-Upgrade (11-40%) C-Poor (41-60%) D-Unacceptable (61-100%)
SIMILARITY %	MATCHED SOURCES	GRADE	

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	index-of.es	1	Publication
2	Submitted to Visvesvaraya Technological University, Belagavi	1	Student Paper
3	www.redalyc.org	1	Internet Data
4	ijstr.org	1	Publication
5	aanda.org	<1	Internet Data
6	Submitted to Visvesvaraya Technological University, Belagavi	<1	Student Paper
7	www.conveythis.com	<1	Internet Data

APPENDIX B

If your project report is presented as paper in National/ International Conferences. Attach scanned copy of certificate/email confirmation and paper in Appendix.

Include all student author certificates.