caption

M.Tech Dissertation Report

titled

ENHANCED CONDITIONAL GAN-AUGMENTED DEEP NEURAL NETWORK WITH SELF-ATTENTION FOR DIABETIC FOOT ULCER CLASSIFICATION

Submitted in partial fulfilment towards the award of the degree of

MASTERS OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

by

Ms. Tejal Khade

P23CS019

Supervisor

Dr. Chandra Prakash, SVNIT, Surat

2024 – 2025

Department of Computer Science and Engineering

SVNIT, SURAT.

DECLARATION

I hereby declare that the work being presented in this dissertation report entitled "Enhanced Conditional GAN-Augmented Deep Neural Network with Self-Attention for Diabetic Foot Ulcer Classification" by me i.e. Ms. Tejal Khade, bearing Roll No: P23CS019 and submitted to the Department of Computer Science and Engineering Sardar Vallabhbhai National Institute of Technology, Surat; is an authentic record of my own work carried out during the period of July 2024 to June 2025 under the supervision of Dr. Chandra Prakash. The matter presented in this report has not been submitted by me to any other University/Institute for any cause.

Neither the source code there in, nor the content of the project report have been copied or downloaded from any other source. I understand that my result grades would be revoked if later it is found to be so.

_____

(Tejal Khade)

C E R T I F I C A T E

This is to certify that the dissertation report entitled " Enhanced Conditional GAN-Augmented Deep Neural Network with Self-Attention for Diabetic Foot Ulcer Classification ", prepared and presented by Ms. Tejal Khade, bearing Admn. No: P23CS019 of MTech.- II, Semester - IV in Computer Science And Engineering, at Department of Computer Science and Engineering of the Sardar Vallabhbhai National Institute of Technology, Surat is satisfactory.

Certified By

Dr. Chandra Prakash
Assistant Professor,
Department of Computer
Science and Engineering,
Sardar Vallabhbhai National
Institute of Technology,
Surat - 395007,
India

Jury's Signature

PG Incharge,
M.Tech in CSE
SVNIT, Surat

Head,
Department of Computer
Science and Engineering,
SVNIT, Surat

SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY, SURAT

Department of Computer Science and Engineering

(2024-25)

Approval Sheet

This is to state that the Dissertation Report entitled Enhanced Conditional GAN-Augmented Deep Neural Network with Self-Attention for Diabetic Foot Ulcer Classification submitted by Ms. Tejal Khade (Admission No: P23CS019) is approved for the award of the degree of Masters of Technology in CSE.

Board of Examiners

Examiners

_____

_____

Supervisor(s)

_____

_____

Head, Department of Computer Science and Engineering

_____

Date:_____

Place:_____

Acknowledgments

I am grateful to Dr. Chandra Prakash, who has been a great advisor from the very beginning. I am thankful to him for his valuable discussions and the numerous contributions that he has provided to this work. Without his support and guidance, this work could not have been accomplished. Besides my advisor, I would like to thank my research progress committee members for their encouragement, insightful comments, and suggestions.

I want to thank Dr. Sankita J. Patel, Head of Computer Science and Engineering, SVNIT, Surat, for allowing me to explore research aspects of security and providing infrastructural facilities for my work. I want to thank all the faculties and staff members of the Computer Science and Engineering Department, SVNIT, Surat.

Tejal Khade

P23CS019

Abstract

Generative Adversarial Networks (GANs) have emerged as powerful tools in medical imaging, demonstrating significant potential in tasks such as image classification, segmentation, detection, denoising, and reconstruction. Their ability to generate high-quality synthetic data makes them particularly valuable in addressing common challenges in medical datasets, such as limited data availability and severe class imbalance. One such application is the classification of Diabetic Foot Ulcers (DFUs) which is a serious complication of diabetes that, if undetected or untreated, can lead to infection, amputation, or even death. Early and accurate classification of DFUs is crucial for timely intervention; however, deep learning models often struggle with poor generalization due to the lack of diverse, balanced datasets. To overcome these limitations, we propose a novel classification framework that integrates data augmentation through an enhanced Conditional GAN (cGAN) with a self-attention-based convolutional neural network. The enhanced cGAN generates realistic and diverse synthetic images for underrepresented classes in the DFUC2021 dataset, thereby mitigating class imbalance and improving training diversity. Our cGAN demonstrates superior performance in image generation, achieving a Fréchet Inception Distance (FID) of 32.14, Peak Signal-to-Noise Ratio (PSNR) of 18.32, and Structural Similarity Index Measure (SSIM) of 0.451. When this augmented data is combined with the proposed classification model, the framework achieves a classification accuracy of 92.57%, outperforming several state-of-the-art architectures. This work highlights the effectiveness of GAN-based augmentation in enhancing model robustness and classification performance of DFU.

Keywords: *Generative Adversarial Networks, Diabetic Foot Ulcer, Data Augmentation, Class Imbalance, Conditional GAN, Medical Image Classification, Self-Attention, Convolutional Neural Network, Deep Learning.*

Table of Contents

List of Figures

List of Tables

List of Acronyms

DFU Diabetic Foot Ulcer

GAN Generative Adversarial Network

cGAN Conditional Generative Adversarial Network

CNN Convolutional Neural Network

FID Fréchet Inception Distance

PSNR Peak Signal-to-Noise Ratio

SSIM Structural Similarity Index Measure

GAP Global Average Pooling

ReLU Rectified Linear Unit

BN Batch Normalization

Chapter 1

Introduction

The advancement of artificial intelligence in medical studies has created a growing need for Generative AI (GenAI) tools that can synthesize high quality, diverse, and contextually meaningful data. In domains like medical imaging, collecting large, balanced, and annotated datasets is often difficult due to security concerns, expert dependency, and the rarity of certain conditions. GenAI offers powerful solutions by addressing class imbalance, enhancing data diversity, and supporting model generalization, all of which are critical for improving the performance of deep learning systems. Among various GenAI techniques, Generative Adversarial Networks (GANs) introduced by Ian Goodfellow in 2014 have emerged as a transformative approach for data augmentation[1].GANs consist of two neural networks: a Generator, which creates synthetic data, and a Discriminator, which evaluates the authenticity of that data against real samples. These networks are trained in opposition while the generator tries to fool the discriminator, the discriminator learns to distinguish real from fake. This adversarial process enables the generator to learn the data distribution and produce realistic synthetic outputs. GANs are especially effective for generating high-quality visual data, making them valuable in domains like medical imaging and computer vision.

In medical imaging, GANs have shown effectiveness in image synthesis, segmentation, denoising, and disease classification. Specifically, for tasks such as Diabetic Foot Ulcer (DFU) classification, where dataset scarcity and class imbalance are major challenges, GANs play a pivotal role. By generating synthetic DFU patches particularly for underrepresented classes they help models learn subtle and complex ulcer patterns, ultimately improving classification performance and robustness. Despite their potential, GANs still face challenges in balancing quality and diversity in generated outputs. However, their application in medical image augmentation represents a crucial step toward building more data-efficient, accurate, and clinically reliable AI models.

1.1 Need of Generative Adversarial Networks

GANs have transformed deep learning by empowering models to produce remarkably realistic synthetic data. GANs consist of two neural networks a generator and a discriminator that compete in a game-theoretic framework to improve each other's performance. The generator attempts to produce convincing fake data, while the discriminator learns to differentiate between real and generated samples. Through this adversarial training process, GANs can create data distributions that closely mimic real world datasets. Beyond their role in image synthesis, GANs have become essential tools in areas such as image restoration, super resolution, domain translation, data anonymization, and even simulation of rare scenarios where real data is scarce. In medical imaging, their potential is particularly significant. Traditional augmentation techniques like flipping, rotating, and adding noise are often insufficient for capturing the complex and diverse characteristics of clinical data. GANs overcome this limitation by generating entirely new samples that not only expand the dataset but also preserve its clinical relevance.

Real-world applications have demonstrated the value of GANs in augmenting medical datasets, including generating synthetic chest X-rays for COVID-19 diagnosis, enhancing brain tumor detection, and supporting lung disease classification. Notably, during the COVID-19 pandemic, GANs were used to generate synthetic chest X-ray images to overcome the scarcity of annotated COVID-positive cases, enabling researchers to improve the performance of diagnostic models without exposing sensitive patient data. Similarly, GAN-based augmentation has been employed for brain tumor segmentation in MRI scans, where collecting labeled data is both expensive and time consuming. Other applications include lung disease detection, skin lesion synthesis, and retinal disease classification, all of which benefit from the increased data diversity and balance provided by GANs.

1.2 Applications of GAN

Generative Adversarial Networks (GANs) have been widely adopted across domains such as medical imaging, data generation, multimedia, and education. In healthcare, GANs address challenges like data scarcity and low quality images by generating realistic synthetic data. Applications include super-resolution, denoising, and cross modality translation (e.g., CT to MRI), which enhance diagnostic accuracy. GANs also aid classification and segmentation by producing diverse training samples, as demonstrated by CovidGAN, which generated synthetic X-ray images during the early COVID-19 outbreak.

Beyond healthcare, GANs are used for image synthesis, generating training datasets for object detection, facial recognition, and scene understanding. In multimedia, they support video super-resolution, face animation, lip-syncing, and speech enhancement. Text-to-image mod-

els such as DALL-E and AttnGAN enable content creation from natural language, benefiting fields like marketing and education. GANs also power image augmentation, creating varied and realistic samples to boost model generalization, especially in low-data scenarios.

In gaming and virtual reality, GANs automate the generation of textures, characters, and environments. For security and surveillance, they provide synthetic data for facial recognition and restore degraded images. In education, GANs simulate realistic visuals for training in medical and technical domains. Some of the key applications of GANs discussed above are visually illustrated in Figure 1.1.

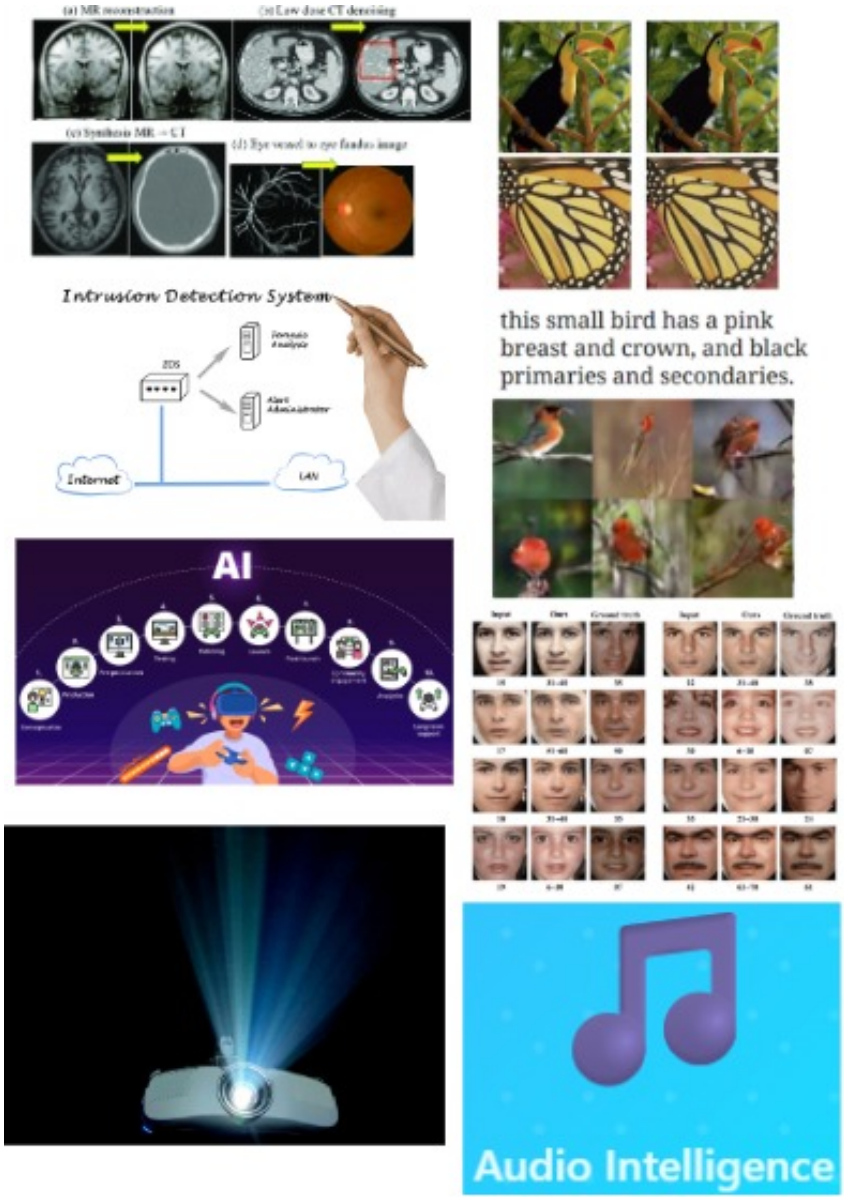

Figure 1.1: Applications of GAN

1.3 Types of Generative Adversarial Networks(GANs)

A hierarchical categorization of various types of Generative Adversarial Networks (GANs) is shown in Figure 1.2 showcasing the diversity of GAN architectures designed for medical imaging tasks.

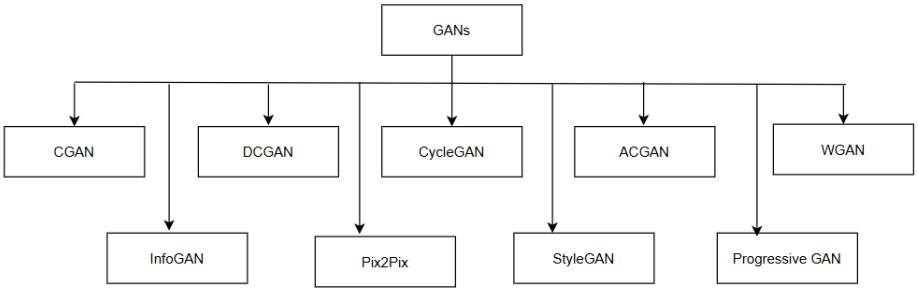

Figure 1.2: Types of GAN[1]

1. **CGAN (Conditional GAN)[1]**: Conditional GANs (cGANs) enhance the original GAN framework by introducing auxiliary information such as class labels or specific attributes into both the generator and discriminator, allowing the model to generate data conditioned on that information. This guidance helps the generator produce more class specific and controlled outputs, making cGANs particularly useful for tasks where labeled data is available. In medical imaging, they are effective in generating targeted synthetic images to address class imbalance by producing minority class samples with high relevance and quality.

2. **DCGAN (Deep Convolutional GAN)[1]**: DCGANs replace traditional fully connected layers with convolutional and transposed convolutional layers, allowing the model to better capture spatial hierarchies in images. This architecture enables the generation of high-quality, realistic images and forms the foundation for many advanced GAN variants. Its stability and performance have made it widely used in computer vision applications.

3. **CycleGAN[1]:** CycleGANs facilitate image to image translation between two domains without requiring paired examples. By introducing a consistency loss, they ensure that translating an image to another domain and back results in the original image. This architecture proves valuable in applications such as translating images between modalities (e.g., MRI to CT), particularly in situations where perfectly aligned training pairs are not available.

4. **ACGAN (Auxiliary Classifier GAN):[1]** ACGAN enhances the cGAN architecture by enabling the discriminator to not only distinguish real from fake images but also classify

them into predefined categories. This variant of GAN was used in [3]. This dual-task discriminator improves the quality of generated images and ensures that they carry semantically meaningful features related to their class labels.

5. **WGAN (Wasserstein GAN):[1]** WGAN introduces the Wasserstein (Earth Mover's) distance as a new loss function to overcome the instability and mode collapse often seen in traditional GANs. This metric provides smoother gradients and more stable training, enabling the generation of higher-quality images even with challenging datasets.

6. **InfoGAN[1]:** InfoGAN is an unsupervised variant of GANs that encourages the discovery of disentangled and interpretable latent representations. It introduces additional latent codes and a mutual information term in the loss function to ensure the generator captures distinct, meaningful variations in the output (e.g., rotation, size, or color).

7. **Pix2Pix [1]:** Pix2Pix is a supervised image-to-image translation model based on cGANs, requiring paired training data. It is designed to generate an output image that closely corresponds to a given input image. Applications include semantic segmentation, sketch-to-photo generation, and medical image translation where input-output pairs are known.

8. **StyleGAN [1]:** StyleGAN introduces a novel architecture that allows control over the style of different levels of image features from coarse structures to fine textures. By mapping a latent vector into an intermediate space and using style modulation, StyleGAN produces highly realistic and diverse images, especially useful in applications requiring fine-grained control such as face synthesis.

9. **Progressive GAN (Progressive Growing GAN) [1]:** This architecture gradually increases the resolution of both the generator and discriminator during training, starting from very low resolutions. This progressive training approach allows the network to first learn coarse features and then refine them, resulting in high-resolution and detailed image generation with improved stability.

1.4 Use Case in Medical Imaging: Diabetic Foot Ulcer Classification

Diabetes, currently impacting over 425 million people globally and projected to rise to 629 million by 2045, significantly increases the risk of neuropathy and peripheral arterial disease—two primary contributors to DFU development[4]. A Diabetic Foot Ulcer (DFU) is an open wound or sore as shown in Figure 1.3 that typically forms on the foot of a person with diabetes, often as a result of prolonged high blood sugar levels that damage nerves (neuropathy) and reduce

blood circulation (peripheral arterial disease). These conditions impair the body's ability to sense pain and heal injuries, making even minor cuts or pressure points susceptible to serious infections.

DFUs commonly develop on weight-bearing areas like the soles and can go unnoticed due to a lack of sensation. If not identified and treated promptly, DFUs can lead to severe complications such as infection, gangrene, and ultimately, lower-limb amputation. Early detection, accurate classification, and timely intervention are critical for improving patient outcomes and preventing long-term disability. The growing prevalence of DFUs presents a significant burden on healthcare systems worldwide, both in terms of resources and patient outcomes. Consequently, early detection and regular monitoring potentially through assistive technologies like mobile applications are becoming increasingly critical. Recent studies have explored automated DFU detection algorithms that could empower patients to participate actively in wound monitoring, facilitating timely medical intervention and reducing complications.

Figure 1.3: Diabetic Foot Ulcer(DFU) [2]

1.5 Motivation

Diabetic Foot Ulcer (DFU) is a serious diabetes complication that can lead to infection, amputation, or death if not diagnosed early. Accurate DFU classification is essential for timely treatment, but automated systems often struggle due to limited, imbalanced, and less diverse medical image datasets, affecting model performance and generalization.

Traditional data augmentation methods often fall short in capturing the complex visual patterns of medical conditions like DFUs. This highlights the need for advanced techniques that can generate realistic and diverse samples to enrich training data. Generative Adversarial Networks (GANs) address this gap by producing high-quality synthetic images, making them a powerful tool for medical image enhancement.

GANs have emerged as a powerful tool in medical image synthesis, offering the ability to generate high-quality, diverse, and lifelike images. Their application in DFU classification not only enhances the quantity and variety of training data but also addresses the imbalance between normal and abnormal classes more effectively than traditional augmentation methods.

The motivation behind this research stems from the critical need to develop robust and accurate classification systems for DFUs using state-of-the-art techniques. By integrating GAN-based data augmentation with deep neural networks, this work aims to improve model performance, enable early diagnosis, and ultimately contribute to reducing the burden of diabetic foot complications on both patients and healthcare systems.

1.6 Problem Statement

Diabetic Foot Ulcer (DFU) poses a significant health threat to diabetic patients, often leading to severe outcomes such as infection, amputation, or death if not detected and treated promptly. While automated DFU classification systems have the potential to assist in early diagnosis, their effectiveness is hindered by the scarcity, imbalance, and limited diversity of available medical image datasets. To overcome these limitations, there is a need for advanced data generation approaches that can enrich training datasets with realistic and diverse samples. Generative Adversarial Networks (GANs), known for their ability to produce high quality synthetic images, offer a promising solution. However, the integration of GAN generated data into deep learning pipelines for DFU classification remains an underexplored area. This research addresses the problem of improving DFU classification accuracy by leveraging GAN-based data augmentation to generate high-quality synthetic images.

1.7 Objective

The main objective of this research is to overcome the limitations posed by data scarcity and class imbalance in Diabetic Foot Ulcer (DFU) classification using Generative Adversarial Networks (GANs). By generating high-quality and diverse synthetic images, GANs enrich the dataset, thereby enhancing the training process of deep learning models. This study aims to integrate GAN-augmented data with a custom deep neural network to significantly improve classification performance. Furthermore, it addresses broader challenges like detecting rare conditions, ensuring better generalization across unseen data and improving model robustness. Through comprehensive experimentation and analysis, this study demonstrates that GAN powered augmentation serves as a reliable and effective strategy for enhancing the accuracy, consistency, and robustness of diabetic foot ulcer (DFU) classification systems.

1.8 Report Outline

The structure of the report is organized as follows: Chapter 2 presents the theoretical background and literature survey, highlighting related work in diabetic foot ulcer (DFU) classification and identifying the limitations of existing methods. Chapter 3 describes the proposed implementation methodology, detailing the architecture. Chapter 4 provides a in depth analysis of the experimental results, evaluating the performance of the proposed model using standard classification metrics and comparing it against state-of-the-art methods. Finally, Chapter 5 concludes report by the key findings and powerful future directions to further enhance DFU classification using advanced deep learning techniques.

Chapter 2

Theoretical Background & Literature Survey

2.1 Theoretical Background

The reviewed literature, highlights current approaches to Diabetic Foot Ulcer (DFU) classification facing several critical limitations that hinder their clinical effectiveness and scalability shown in Table 2.5. The foremost challenges is the scarcity of large, diverse, and well labelled datasets. Most existing datasets are relatively small and lack representation across different stages of ulcers, skin tones, and imaging conditions, making it difficult for models to generalize effectively in real-world scenarios. Additionally, persistent class imbalance where abnormal or minority classes are significantly underrepresented remains inadequately addressed by conventional data augmentation techniques such as rotation, flipping, or translation. While some studies have adopted synthetic data generation methods using GANs or similar techniques, over reliance on synthetic images without proper validation can lead to models that perform well during training but fail to generalize on unseen clinical data.

This chapter presents the theoretical foundation and relevant prior research essential for understanding the proposed methodology. It begins by discussing the core principles of deep learning, focusing on Convolutional Neural Networks (CNNs), which serve as the backbone for image classification tasks. The chapter then introduces Generative Adversarial Networks (GANs), explaining their architecture, working mechanism, and role in data augmentation. These concepts are particularly relevant in addressing common challenges in medical imaging, such as class imbalance and data scarcity. Finally, a literature survey is provided to review existing works in diabetic foot ulcer (DFU) classification and highlight the research gaps that motivate this study.

Deep learning has revolutionized medical image analysis by enabling automated feature extraction and classification. It allows models to learn complex visual patterns from large datasets, improving diagnostic accuracy and efficiency. In medical imaging applications, such as diabetic foot ulcer (DFU) classification, deep learning facilitates early detection, severity assessment,

and categorization of pathological conditions with high precision. As a result, deep learning has become a cornerstone of modern computer aided diagnosis systems, contributing significantly to improved healthcare outcomes, especially in scenarios where rapid, reliable, and repeatable diagnostic support is essential. Among deep learning architectures, Convolutional Neural Networks (CNNs) have been particularly impactful due to their ability to effectively capture spatial hierarchies and local patterns in image data. CNNs consist of convolutional layers that learn to detect features such as edges, textures, and shapes, making them well-suited for identifying complex pathological patterns in medical images. In applications such as diabetic foot ulcer (DFU) classification, CNNs can facilitate early detection, severity assessment, and accurate categorization of lesions with high precision and minimal manual intervention.

Data augmentation is a critical strategy in deep learning, especially when working with limited or imbalanced datasets, as commonly encountered in medical imaging. Traditional augmentation techniques include geometric transformations such as rotation, flipping, scaling, and cropping, as well as photometric alterations like brightness adjustment, contrast variation, and noise injection[5]. These methods artificially expand the training dataset by introducing minor variations, helping models generalize better and reduce overfitting. However, in complex medical tasks such as diabetic foot ulcer (DFU) classification these basic transformations may not sufficiently capture the wide range of pathological variations present in real world cases. To address this, more advanced augmentation methods such as Generative Adversarial Networks (GANs) have been introduced. GANs generate entirely new, realistic synthetic samples that resemble true medical images, thereby enriching the diversity and quality of the dataset.

GANs are composed of two networks: a generator and a discriminator. The generator creates synthetic data resembling real samples, while the discriminator evaluates their authenticity. This adversarial training results in highly realistic data generation, useful for augmenting medical datasets suffering from scarcity or imbalance. Data augmentation enhances training datasets by applying transformations like rotation, flipping, and noise injection. Though helpful, these conventional methods often fall short in representing complex variations found in medical images. GANs provide a more advanced form of augmentation by generating entirely new but realistic samples.

2.2 Literature Survey

Research on Diabetic Foot Ulcer (DFU) classification has achieved significant momentum in recent years due to its critical role in early diagnosis and improved patient care. The classification of Diabetic Foot Ulcers (DFUs) has become an increasingly important area of research due to its significant implications for early diagnosis, personalized treatment planning, and the

overall reduction of diabetes-related complications. This section reviews notable developments in the application of image processing, classical machine learning, and modern deep learning algorithms to DFU classification, highlighting their contributions, limitations, and scope for future improvements.

N. Bansal and A. Vidyarthi proposed a deep neural architecture for diabetic foot ulcer (DFU) classification, combining CNNs with residual blocks and feature fusion layers for improved feature extraction[5]. Data augmentation (rotation, flipping, Gaussian noise, shearing, translation) enhanced model robustness. The model achieved 98.87% accuracy on DFU2020 and MICCAI datasets. Limitations include the need for more diverse data, hyperparameter tuning, and broader clinical validation.

Hamghalam and Simpson introduced two cGAN-based models—EnhGAN and ESGAN—for brain tumor segmentation in MRI scans [6]. Using PDF transformation blocks, these models enhance intensity distribution to improve class separability. Tested on BraTS'13 and BraTS'18 datasets, ESGAN performed well on small datasets, while EnhGAN improved segmentation in complex tumor regions. Despite their effectiveness, the models face limitations in generalizing across diverse clinical data and require further validation for broader applicability.

A. Waheed et al., explores the use of deep learning, particularly Convolutional Neural Networks (CNNs), for detecting COVID-19 using chest X-rays (CXR) in[3]. The researchers employed the VGG16 CNN model, which was limited by the availability of a small dataset. To overcome this challenge, they introduced an Auxiliary Classifier Generative Adversarial Network (ACGAN)-based approach called CovidGAN, designed to generate synthetic CXR images. Incorporating these synthetic images into the training process significantly improved the performance of CNNs for COVID-19 detection, increasing classification accuracy from 85% to 95%.

J. Amin et al. combined Quantum Machine Learning (QML) and Classical Machine Learning (CML) for COVID-19 classification using CT scans[7] . They highlighted key diagnostic features like ground-glass opacity and pulmonary consolidation. To overcome limited data, a Conditional GAN (CGAN) generated synthetic CT images for training. Both quantum and classical models were evaluated, achieving accuracy, precision, recall, and F1-scores up to 0.96 on UCSD-AI4H and 0.94 on POF Hospital datasets. This approach demonstrates the effectiveness of synthetic data augmentation combined with advanced ML techniques for COVID-19 diagnosis.

M. S. A. Toofanee et al. proposed an ensemble of CNNs and Vision Transformers (ViT) for diabetic foot ulcer (DFU) classification, integrating a Siamese Neural Network (SNN) with a k-Nearest Neighbors (kNN) classifier for improved performance [8]. K-Fold cross-validation ensured robustness, while data augmentation addressed class imbalance. The DFUC2021 dataset included classes: None, Infection, Ischemia, and Both. Despite these improvements, the study

was limited by the inability to explore more complex ensemble models or computationally intensive approaches.

A. Qayyum et al. in addresses the growing challenge of Diabetic Foot Ulcers (DFUs), particularly those with ischemia and infection, emphasizing the need for early detection[9]. The authors proposed using pre-trained transformer models, fine tuned on the DFUC-21 dataset, for multiclass DFU classification.A Multi-Model approach was proposed, where features from parallel-trained transformers were fused from the last layers, achieving a macro-average F1-Score of 0.569. Weighted cross-entropy optimization and pairwise feature fusion addressed class imbalance. The results highlight the potential of combining CNNs with transformer architectures for future improvements in DFU classification.

M. H. Yap et al. addressed diabetic foot ulcer (DFU) classification challenges using the DFUC 2021 dataset with significant class imbalance[4]. Evaluation metrics included per-class F1-Score, micro-average F1, and macro-averages of Precision, Recall, F1-Score, and AUC. Pretrained ImageNet models and data augmentation were applied—eight augmentations for ischaemia images, three for infection and ischaemia classes—to increase sample size. DenseNet121 achieved the highest macro-average AUC (0.88), while EfficientNetB0 excelled in macro-average Precision, Recall, and F1-Score, especially improving infection detection. UMAP analysis showed EfficientNetB0 improved intra-class clustering, but inter-class separation remained challenging. Detecting infection and co-occurrence of ischaemia and infection ("both" category) remains difficult.

L. Alzubaidi and A. A. Abbood et al. designed four hybrid CNN models for DFU classification, comparing architectures with varying branch numbers [10]. They used feature aggregation with Global Average Pooling and dropout-enhanced fully connected layers, and a Softmax layer for output. The dataset had 754 foot images labeled as abnormal (DFU) or normal (healthy skin). Limitations included no performance gain from increased network width, small dataset size, and focus on only two classes. Future work aims to apply transfer learning.

L. Alzubaidi, M. A. Fadhel et al. proposed a novel CNN model, DFU_QUTNet, for diabetic foot ulcer classification, comparing it to GoogleNet, AlexNet, and VGG16 after fine-tuning[11]. Paired with an SVM classifier, DFU_QUTNet achieved a higher F1-Score of 94.5%. Features extracted were used to train SVM and KNN classifiers, with SVM yielding the best precision, recall, and F1-Score. The dataset included 754 resized foot images (224x224). Limitations were the small dataset size, limited generalization to other tasks, and lack of clinical validation or expert comparison.

M. Goyal et al. compared an ensemble CNN model with traditional machine learning algorithms for binary DFU classification (Ischaemia vs. Non-Ischaemia and Infection vs. Non-Infection) using 1459 images from Lancashire Teaching Hospitals [2]. Despite using natural data augmentation, challenges such as class imbalance, visual similarity between classes, and

poor image quality impacted performance. The study noted that infections often lack clear visual indicators, making classification difficult.

M. Goyal et al. proposed a hybrid approach combining Conventional Machine Learning (CML) techniques and Convolutional Neural Networks (CNNs) for effective classification of diabetic foot ulcers (DFUs) into ulcer and non-ulcer categories[12]. At the core of their methodology was the development of a custom-designed CNN architecture named DFUNet, tailored specifically for analyzing complex and variable DFU image data. DFUNet was designed to improve input preprocessing and feature extraction efficiency, incorporating architectural modifications that make it more suitable for medical image analysis compared to standard deep learning models. The study employed 10-fold cross-validation to ensure robust and unbiased performance evaluation. DFUNet demonstrated a high Area Under the Curve (AUC) of 0.961, outperforming several well-established deep learning architectures such as LeNet, AlexNet, and GoogLeNet. The authors also applied a comprehensive set of data augmentation techniques—including rotation, flipping, contrast enhancement, color perturbations, and scaling—to increase dataset variability and reduce overfitting. However, it was noted that these augmentation strategies did not significantly enhance the model's performance, indicating the strength of the base DFUNet architecture. Despite this, DFUNet achieved better classification performance in terms of accuracy, sensitivity, and specificity compared to deeper and more complex networks like GoogLeNet and AlexNet, highlighting its effectiveness and computational efficiency. This work illustrated the importance of architecture customization for domain-specific tasks and set a benchmark for DFU classification using lightweight and optimized deep learning models.

Ahmed Makhlouf et al. presented a in-depth systematic analysis of the applications of Generative Adversarial Networks (GANs) for medical image augmentation, addressing one of the most pressing issues in the domain limited and imbalanced datasets [1]. These challenges often hinder the development of accurate and generalizable deep learning models in healthcare, where data acquisition is expensive, time-consuming, and constrained by privacy concerns. The review focuses on how GANs, by generating high-quality synthetic medical images, serve as a powerful augmentation tool to improve model robustness, especially for underrepresented classes or rare pathologies. The authors examined 52 peer-reviewed publications from 2018 to 2022, categorizing the literature based on several important dimensions. First, they analyzed popular GAN architectures used in medical imaging, such as DCGAN, Pix2Pix, CycleGAN, and StyleGAN, identifying their suitability for different tasks. They also mapped the studies to common medical imaging modalities, including MRI, CT, ultrasound, and X-ray, and to target organs and anatomical regions such as the brain, lungs, breast, and retina. Furthermore, the paper explored the downstream tasks for which GAN-generated data was applied, with a focus on classification, segmentation, detection, and image reconstruction. The review also

categorized evaluation metrics used to assess GAN effectiveness. These included qualitative visual assessments by experts, quantitative direct metrics such as Structural Similarity Index Measure (SSIM), Peak Signal-to-Noise Ratio (PSNR), and Fréchet Inception Distance (FID), as well as indirect quantitative methods, which involve measuring performance improvements in downstream tasks (e.g., classification accuracy or segmentation Dice score) when trained with GAN-augmented datasets. Overall, the review emphasized that GANs not only help in mitigating class imbalance and improving generalization but also accelerate AI adoption in medical imaging by enabling data efficient learning.

N. Al-Garaawi et al. proposed a novel method combining mapped binary patterns with convolutional neural networks for diabetic foot ulcer classification [13]. Their approach leverages handcrafted texture features alongside deep learning to enhance the discrimination of ulcer tissue types. The model was evaluated on a clinical dataset, showing improved classification accuracy and robustness compared to traditional methods. This hybrid strategy highlights the benefit of combining domain-specific features with CNN architectures for medical image analysis.

N. Tajbakhsh et al. investigated the trade-offs between full training and fine-tuning of convolutional neural networks (CNNs) for medical image analysis tasks [14]. Through extensive experiments on multiple datasets, they demonstrated that fine-tuning pretrained networks often outperforms training from scratch, especially with limited labeled data. Their study provided practical guidelines for CNN adaptation in healthcare imaging, emphasizing transfer learning as a powerful technique to boost performance while reducing training time and resource requirements.

F. Veredas et al. developed a binary tissue classification framework for wound images using neural networks combined with Bayesian classifiers [15]. Their method integrates probabilistic modeling with learned features to achieve robust segmentation of wound tissues. Applied to medical wound datasets, this approach enhanced classification accuracy and enabled better wound characterization, contributing valuable tools for automated wound assessment and monitoring.

C. Liu et al. presented an automatic detection system for diabetic foot complications based on infrared thermography using asymmetric analysis [16]. Their method captures thermal anomalies indicative of tissue damage by analyzing temperature differences between symmetrical foot regions. Validated on clinical thermographic images, the system demonstrated promising accuracy in early detection of diabetic foot pathologies, highlighting the potential of non-invasive imaging techniques for timely diagnosis.

L. Wang et al. introduced a smartphone-based wound assessment system tailored for diabetic patients [17]. This system combines mobile imaging with automated analysis algorithms to quantify wound characteristics remotely. Through clinical trials, the approach showed re-

liable wound measurement and classification capabilities, offering a practical tool for patient self-monitoring and telemedicine applications, thus improving diabetic wound care accessibility.

M. H. Yap et al. developed an innovative mobile application designed to standardize the acquisition process of diabetic foot images, thereby enhancing diagnostic consistency and facilitating automated analysis workflows [18]. The core functionality of the application lies in its ability to guide users both patients and clinicians through a structured image capture protocol that includes controlled positioning, distance, angle, and lighting conditions. By enforcing these constraints, the app ensures that the collected images maintain a consistent format, which is crucial for both manual evaluation by healthcare professionals and input into AI based diagnostic systems. The research emphasizes that inconsistency in image acquisition remains a significant barrier in diabetic foot ulcer (DFU) assessment, often leading to unreliable interpretations. Field testing of the application demonstrated substantial improvements in image quality, uniformity, and reproducibility across different users and environments.

H.-C. Shin et al. conducted a comprehensive investigation into the application of deep convolutional neural networks (CNNs) for computer-aided detection (CAD) systems in medical imaging, with a particular focus on the role of network architecture, dataset characteristics, and the efficacy of transfer learning [19]. The authors benchmarked several CNN models both shallow and deep across a range of clinical imaging modalities, including CT, MRI, and X-ray. Their analysis revealed that the performance of CNNs is highly influenced by the nature and size of the dataset; specifically, larger and more domain-relevant datasets tend to produce better generalization and classification accuracy. In scenarios where annotated medical data is scarce, transfer learning from large-scale natural image datasets (e.g., ImageNet) proved to be a viable and efficient alternative. The study highlights that fine-tuning pretrained CNNs often outperforms models trained from scratch, especially when the medical imaging task shares structural similarities with natural image recognition.

The current research focuses primarily on binary classification (e.g., normal vs. abnormal), overlooking the clinical necessity of multiclass classification for more nuanced diagnosis and treatment planning. There is also a notable lack of rigorous clinical validation, with many models tested only in controlled environments rather than real-world healthcare settings. Furthermore, high visual similarity between classes and variable data quality due to differences in lighting, imaging devices, and foot conditions pose additional barriers to achieving high diagnostic accuracy. In light of these challenges, our work specifically addresses the issues of class imbalance and limited image quality by leveraging enhanced Conditional GAN-based data augmentation and a robust CNN architecture, aiming to improve both the reliability and generalizability of DFU classification systems.

Table 2.1: Summary of Related Work

Year	Ref	Approaches Used	Evaluation Parameters	Dataset and Limitations
2024	[5]	Model trained with CNN Residual Blocks and Feature Fusion Layers; used traditional data augmentation (rotation, flipping, shearing)	Accuracy, Precision, Recall, F1-Score, AUC-ROC	DFU2020, MICCAI DFU dataset. Limited dataset diversity, weak hyperparameter tuning, and lacks clinical deployment validation.
2024	[6]	cGAN-based models (EnhGAN and ESGAN) with PDF transformation blocks to enhance MRI intensity distribution for class separability	Probability Density Function (PDF), AUC-ROC Curve	BraTS'13 and BraTS'18 datasets. Limited generalizability across clinical datasets; requires broader validation.
2022	[9]	Fine-tuned pre-trained transformers for multiclass DFU classification; feature fusion from parallel transformers	Macro-average F1-Score, Weighted Cross-Entropy	DFUC2021 Challenge dataset. Class imbalance remains; future improvement requires integrating CNNs with transformers.
2020	[3]	VGG16 CNN for COVID-19 detection on chest X-rays; enhanced data using ACGAN-generated images	Accuracy, Precision, Recall, F1-score, AUC	Small chest X-ray (CXR) dataset. Performance limited by small dataset size and class imbalance.
2023	[8]	Ensemble of CNN and ViT for DFU classification, combined with SNN and kNN	Accuracy, Precision, Recall, F1-score, AUC	DFUC2021 Challenge dataset. Faced class imbalance; failed to explore ensemble modeling for improved accuracy.
2021	[20]	Advanced models like EfficientDet, Cascade R-CNN, and Faster R-CNN with deformable convolutions for the detection and classification of DFUs.	Accuracy, Precision, Recall	MS-COCO and DFUC2020 datasets. CNNs struggled with remote monitoring due to high false positives.
2022	[7]	Combined Quantum and Classical ML for COVID-19 classification using CGAN to generate high-quality synthetic CT images	Accuracy, Precision, Recall, F1-score	UCSD-AI4H and POF Hospital CT scan datasets. Relied on synthetic data; limited real-world generalization.

Table 2.3: Summary of Related Work (continued)

Year	Ref	Approaches Used	Evaluation Parameters	Dataset and Limitations
2021	[10]	Four hybrid CNN models with multi-branch architectures and feature aggregation using Global Average Pooling	Per-class F1-Score, Macro-average Precision, Recall, F1-Score, AUC	754 foot images (normal vs. abnormal). Performance limited by network width, small dataset size, and binary class focus; transfer learning suggested for future work.
2021	[4]	Pretrained ImageNet models with data augmentation; DenseNet121 and EfficientNetB0 outperformed others	Per-class F1-score, Micro-avg F1, Macro-avg Precision, Recall, AUC	DFUC2021 Challenge dataset. Faced class imbalance, especially poor performance in "both" (infection + ischaemia) category.
2020	[2]	Ensemble CNN compared with traditional ML for binary DFU classification	Accuracy, Precision, Recall	1459 DFU images from Lancashire Teaching Hospitals. Limitations: class imbalance, visual similarity, poor image quality, subtle infection indicators
2020	[11]	Proposed DFU_QUTNet and compared it with GoogleNet, AlexNet, and VGG16 with SVM and KNN classifiers	Precision, Recall, F1-score	754 DFU images. Limitations: small dataset, no clinical validation, limited generalization to other conditions.
2017	[12]	CNN model DFUNet compared against CML models and CNNs like AlexNet, GoogLeNet	Accuracy, Precision, Recall, F1-Score, AUC	Ulcer and non-ulcer image dataset. Limited use of augmentation; performance could be improved with more variability.
2023	[1]	Review of GAN applications for classification, segmentation; analysis of GAN architectures	–	General medical imaging datasets. Addressed data insufficiency and class imbalance; highlights importance of synthetic data in training robust models.

Table 2.5: Summary of Related Work (continued)

Year	Ref	Approaches Used	Evaluation Parameters	Dataset and Limitations
2022	[13]	Hybrid method combining mapped binary patterns with CNNs for DFU classification	Accuracy, Robustness	Clinical DFU dataset; demonstrated improved classification accuracy by combining handcrafted and deep features.
2016	[14]	Comparative study on CNN full training vs fine tuning for medical imaging	Accuracy, Training time, Transfer learning efficacy	Multiple medical image datasets; showed fine tuning outperforms training from scratch with limited data.
2010	[15]	Neural networks combined with Bayesian classifiers for binary tissue classification in wound images	Classification accuracy, Segmentation quality	Medical wound image datasets; probabilistic integration improved tissue segmentation.
2015	[16]	Infrared thermography analysis with asymmetric temperature difference for diabetic foot complications detection	accuracy,precision,recall	Clinical thermographic images; non-invasive early detection with promising accuracy but limited dataset size.
2015	[17]	Smartphone-based automated wound assessment system for diabetic patients	accuracy, precision,recall	Clinical wound images captured via smartphones; practical for telemedicine but affected by image quality variability.
2018	[18]	Mobile app for standardized diabetic foot image acquisition ensuring uniform lighting and positioning	Image quality, Reproducibility	Field-tested diabetic foot images; improved diagnostic consistency, limited by user adherence to protocol.
2016	[19]	Analysis of CNN architectures and transfer learning impact on medical CAD tasks	accuracy, Training efficiency	Various medical imaging datasets; emphasized effectiveness of transfer learning on limited data.

2.3 Research Challenges

The existing research shows that medical image analysis especially for classifying Diabetic Foot Ulcers (DFUs) still faces many challenges. Even though deep learning has improved significantly, there are still key problems that make it hard to apply DFU classification systems effectively in real-world settings. These challenges include:

- **Limited and non-diverse datasets:** A significant challenge in medical image analysis, particularly for tasks like diabetic foot ulcer classification, is the scarcity of large, well-annotated datasets. Most publicly available datasets tend to be relatively small in size and often lack sufficient diversity across key factors such as patient demographics (age, gender, ethnicity), ulcer stages (early to advanced), and imaging conditions (lighting, resolution, device types). This limited diversity restricts the model's exposure to the full spectrum of variations that occur in real-world clinical settings. As a result, models trained on such constrained datasets tend to overfit to the specific characteristics of the training data, which impairs their ability to generalize to new patients or different clinical environments. This issue underscores the need for comprehensive datasets that better represent the heterogeneity of the target population and clinical scenarios to develop robust and reliable AI systems.

- **Class imbalance:** Another critical limitation in medical datasets is the uneven distribution of samples across different classes. Certain categories such as severe ulcer stages or rare pathological findings are often underrepresented compared to more common classes. This class imbalance can bias the learning process, causing models to disproportionately favor the dominant classes during training. Consequently, the model's predictive performance on minority or rare classes deteriorates, which is problematic since accurate identification of these less frequent but clinically important cases is essential for effective diagnosis and treatment planning. Addressing class imbalance requires the adoption of specialized strategies such as weighted loss functions, synthetic data generation, oversampling minority classes, or employing advanced algorithms designed to mitigate bias and enhance sensitivity to minority class instances.

- **Insufficient augmentation techniques:** Conventional augmentation techniques—such as rotation, flipping, scaling, cropping, and noise addition—have been widely used to artificially enlarge training datasets and improve model robustness. However, these basic transformations primarily create variations of existing samples rather than generating fundamentally new and diverse examples. As a result, they are often inadequate for capturing the complex, multi-dimensional variability present in real-world medical images, which include diverse anatomical structures, varying imaging modalities, acqui-

sition conditions, and pathological presentations. This limitation is especially critical in medical image classification tasks, where subtle differences in texture, shape, and intensity can dramatically affect diagnosis. Consequently, reliance on conventional augmentation methods may not sufficiently address data scarcity or the class imbalance issues typical of medical datasets, ultimately limiting model performance and generalizability.

- **Poor generalization:** Models trained exclusively on specific, often curated datasets frequently exhibit limited ability to generalize to unseen data collected in real-world clinical environments. This challenge arises due to domain shifts caused by differences in patient demographics, imaging devices, protocols, and noise characteristics. Such models may perform well on validation sets but falter when exposed to novel cases, reducing their clinical utility and reliability. Poor generalization undermines the robustness of AI systems, leading to potential misdiagnoses or missed detections when deployed in diverse healthcare settings. Addressing this issue requires not only more diverse and representative training data but also advanced techniques such as domain adaptation, transfer learning, and incorporation of synthetic data that can bridge the gap between training and real-world scenarios.

- **Binary Classification Focus:** Many existing approaches to diabetic foot ulcer (DFU) analysis simplify the problem by treating it as a binary classification task—distinguishing merely between the presence or absence of ulcers. While this simplification can be useful for initial screening, it neglects the clinically important nuances associated with different ulcer types, severities, and stages. Multi-class or stage-wise classification provides richer diagnostic information by categorizing ulcers according to their progression, infection status, or tissue involvement, which directly informs personalized treatment planning and prognosis. Overlooking these distinctions limits the practical utility of models in guiding effective clinical decision-making and may result in suboptimal patient management.

- **Lack of clinical validation:** A common limitation of many AI based DFU classification studies is their reliance on experimental datasets and controlled validation protocols without rigorous testing in real-world clinical environments. Such studies often lack prospective clinical trials or validation across diverse healthcare settings, which are essential to assess the models' true performance, robustness, and safety. The absence of clinical validation restricts the models' credibility and acceptance among healthcare professionals, impeding their integration into routine practice. To bridge this gap, future research must prioritize clinical evaluation, including prospective studies, multi center trials, and collaboration with clinicians to ensure models deliver meaningful and reliable support in everyday healthcare scenarios.

To address critical challenges such as limited dataset size, severe class imbalance, and poor generalization of deep learning models, this study adopts a comprehensive approach that integrates GAN-based data augmentation with a convolutional neural network (CNN) architecture enhanced by self-attention mechanisms. The use of a Generative Adversarial Network (GAN) enables the generation of realistic, high-quality synthetic images that closely mimic the appearance of actual diabetic foot ulcers. These synthetic images significantly enrich the training set, helping to alleviate data scarcity and mitigate class distribution disparities, particularly for underrepresented ulcer categories. On the other hand, the incorporation of self-attention modules within the CNN framework empowers the model to learn long range spatial dependencies and inter-region contextual cues that are often crucial for accurately identifying complex pathological features. This design not only ensures a more balanced and diverse dataset but also enhances the network's representational capacity and robustness. As a result, the proposed method exhibits improved generalization performance and diagnostic accuracy when applied to challenging real-world DFU images exhibiting variability in ulcer severity, appearance, and imaging conditions.

Based on the insights from the literature and theoretical foundations discussed, it is evident that a combination of advanced data augmentation techniques and deep learning architectures is essential for effective DFU classification. In the next chapter, we present the proposed implementation methodology, detailing the model architecture, data preprocessing strategies, training procedures, and evaluation pipeline adopted in this study.

Chapter 3

Proposed Framework

3.1 Workflow Overview

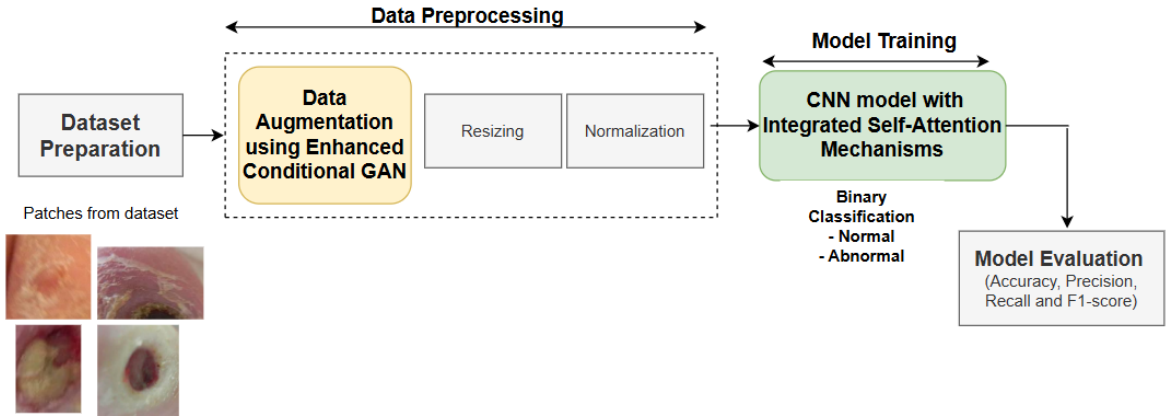

Figure 3.1: Proposed Methodology Pipeline for DFU Classification

The proposed methodology for Diabetic Foot Ulcer (DFU) classification is designed as a comprehensive, systematic pipeline aimed at tackling key challenges in medical image analysis, particularly class imbalance and limited data diversity. The pipeline shown in Figure 3.1 is structured into four major stages: dataset preparation, data preprocessing, model training, and performance evaluation. Each stage plays a crucial role in ensuring the robustness and reliability of the classification system. The proposed methodology for Diabetic Foot Ulcer (DFU) classification is structured into five major stages:

1. **Dataset Preparation:** This study utilizes the DFUC2021 dataset, which comprises of DFU images. The dataset includes two annotated classes: abnormal (1,038 images with ulcers) and normal (641 images without ulcers). Due to the evident class imbalance, augmentation techniques are employed to improve dataset diversity and model performance.

2. **Data Preprocessing:** All images are resized to 224x224 pixels to emphasize the region of interest and reduce background noise. Pixel intensities are normalized from the range

[0, 255] to [-1, 1] to stabilize training. The images are then converted into tensors compatible with deep learning frameworks such as PyTorch or TensorFlow.

3. **Data Augmentation:** An enhanced Conditional Generative Adversarial Network (cGAN) is employed to generate high quality, class conditioned synthetic images. This augmentation particularly targets the underrepresented normal class, enriching the dataset and promoting better generalization during training.

4. **Model Training with Proposed Framework:** A deep learning architecture combining a Convolutional Neural Network (CNN) with self-attention mechanisms is developed for binary classification. The CNN extracts spatial features, while the self-attention modules enhance focus on ulcer-specific regions. The model is trained on a balanced dataset consisting of both real and cGAN generated images, with optimized hyperparameters such as learning rate, batch size, and epochs to improve training efficiency and accuracy.

5. **Image Quality Analysis and Model Evaluation:** The quality of synthetic images generated by the cGAN is evaluated using Fréchet Inception Distance (FID), Structural Similarity Index Measure (SSIM), and Peak Signal-to-Noise Ratio (PSNR). These metrics confirm the realism and structural fidelity of the generated images. The classification model is then assessed using accuracy, precision, recall, and F1 score, demonstrating enhanced robustness and generalization for DFU detection, especially in the presence of class imbalance.

3.2 Dataset

This research utilizes the DFUC2021 dataset, an extensive compilation of diabetic foot ulcer (DFU) images. The dataset includes clinical photographs taken from patients at Lancashire Teaching Hospitals during routine visits. These images capture a wide range of real-world conditions such as differing lighting environments, skin tones, ulcer severities, and camera perspectives. Such diversity enhances the dataset's value for developing and training reliable machine learning models. It serves as the foundation of this study, supporting deep learning-based processes including feature extraction, classification, and performance evaluation. The DFUC2021 dataset is organized into two primary subsets: Part A and Part B.

- Part A is designed for binary classification and includes image patches labeled as Abnormal and Normal. Specifically, it consists of 1,038 DFU patches and 641 normal patches.

- Part B focuses on the recognition of infection and ischaemia within the ulcer regions.

For this study, we focus on Part A to train and evaluate models for the binary classification task distinguishing DFU-affected skin patches from normal skin patches. The DFUC2021 dataset is shown in Table 3.1 contains 1,038 abnormal(ulcer) and 641 normal samples(non-ulcer), reflecting a significant class imbalance.

Table 3.1: DFUC2021 Dataset Description

Data Type	Internal Sub Division	Sample Count
PartA_DFU_Dataset	Abnormal	1038
	Normal	641

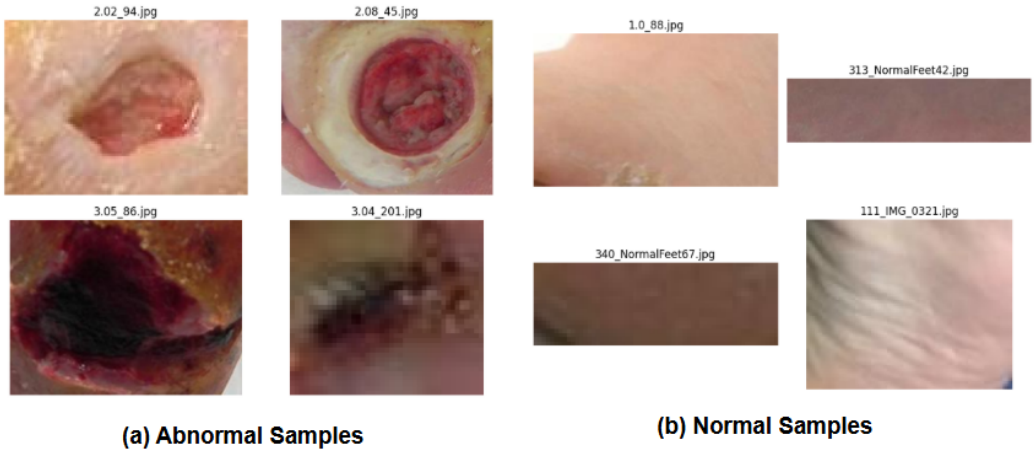

Figure 3.2: Example of Abnormal samples and normal samples from PartA_DFU Dataset

Figures 3.2 display representative sample images from the PartA_DFU dataset, illustrating the visual differences between abnormal and normal foot conditions. left part shows abnormal cases, typically characterized by visible ulceration, discoloration, inflammation, or tissue damage hallmarks of Diabetic Foot Ulcers (DFUs). These images vary in severity and appearance, capturing the heterogeneity of ulcer presentations across patients. In contrast, right part displays normal foot images, which exhibit healthy skin texture without signs of infection or ischemia. Presenting these samples provides insight into the visual cues leveraged by the classification model and underscores the complexity involved in accurately distinguishing between the two classes, especially given the subtle differences and visual noise present in real-world clinical images.

3.3 Data Pre-processing

- **Resizing:** The DFUC2021 dataset comprises images with varying resolutions and aspect ratios. To maintain uniformity and ensure compatibility with the deep neural network architecture, all input images were resized to a fixed dimension of 224×224 pixels. This standardization ensures consistent input dimensions across the dataset, facilitating efficient batch processing and model convergence during training. For example, an original DFU image of size 200×300 pixels would be rescaled proportionally to fit within the target dimensions while preserving the essential structural content. This resizing process is crucial for enabling the model to process inputs of consistent size regardless of their original resolution.

- **Normalization:** To enhance training stability and improve convergence speed, pixel intensity values were normalized from their original [0, 255] range to a standardized range of [-1, 1]. Normalization mitigates internal covariate shift and supports more efficient gradient descent optimization. The pixel values are transformed using the following formula:

$$\text{Normalized Value} = \frac{\text{Pixel Value}}{127.5} - 1$$

  For example, a pixel with value 255 becomes 1, and a value of 0 becomes -1 after normalization. This symmetric scaling is especially suitable when using activation functions like Tanh in the model.

3.3.1 Data Augmentation

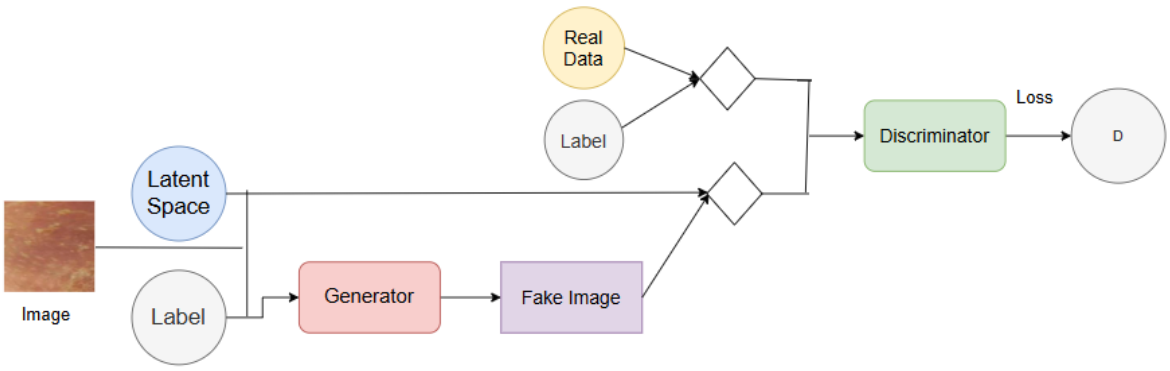

Figure 3.3: CGAN Architecture

Working of Conditional cGAN

A Generative Adversarial Network (GAN) is a deep learning framework consisting of two competing neural networks: the Generator and the Discriminator, trained in a game-theoretic manner. One uses the noise as input and generates samples. The second model which is known as discriminator receives samples from the generator and the training data. According to game theory, the generator is trained to produce an image that looks like a real image, whereas the discriminator is learning to discriminate perfectly from generated data to actual data[8]. A Conditional Generative Adversarial Network (cGAN) shown in Figure 3.3 is a type of Generative Adversarial Network (GAN) architecture that introduces label information into both the generator and discriminator, enabling the controlled generation of data based on specific conditions. Unlike traditional GANs, where the generator produces samples solely from random noise, cGANs incorporate an additional input class labels allowing for more targeted and meaningful generation.

The generator in a cGAN accepts two inputs: a random noise vector from a latent space and a class label representing the desired condition (e.g., "normal" or "abnormal" in DFU classification). It learns to generate realistic synthetic data conditioned on this label. Simultaneously, the discriminator also receives two inputs: an image (either real from the dataset or fake from the generator) and the corresponding label. It evaluates whether the image is real or synthetic, given the provided label. During training, the discriminator learns to correctly classify real versus fake images in the context of the provided labels, while the generator attempts to produce images that not only appear realistic but also align with the given condition. This adversarial training process leads to a generator capable of producing high-quality, label consistent synthetic images.

In the context of medical imaging, particularly for datasets suffering from class imbalance, cGANs offer a powerful solution for data augmentation. Traditional augmentation methods such as rotation, flipping, and scaling merely apply transformations to existing images, which may not sufficiently capture the diversity and complexity of real pathological variations. In contrast, cGANs can generate entirely new, realistic images that mimic the underlying distribution of the data while being conditioned on specific class labels. This label-driven synthetic data generation makes cGANs particularly effective for augmenting datasets in medical applications where data collection is expensive, privacy-sensitive, or inherently imbalanced.

Data augmentation plays a crucial role in mitigating class imbalance by increasing the diversity and quantity of training samples for underrepresented classes. Severe class imbalance, as observed in the DFUC2021 dataset, poses a major challenge to deep learning models, often resulting in overfitting and poor generalization due to limited training samples for the minority class[4]. Traditional methods like rotation and flipping, shearing[5] often fall short. Therefore, we use an enhanced cGAN to generate high-quality synthetic minority class images (see

Figure 3.4).

3.3.2 Block-wise Description of Enhanced CGAN Architecture

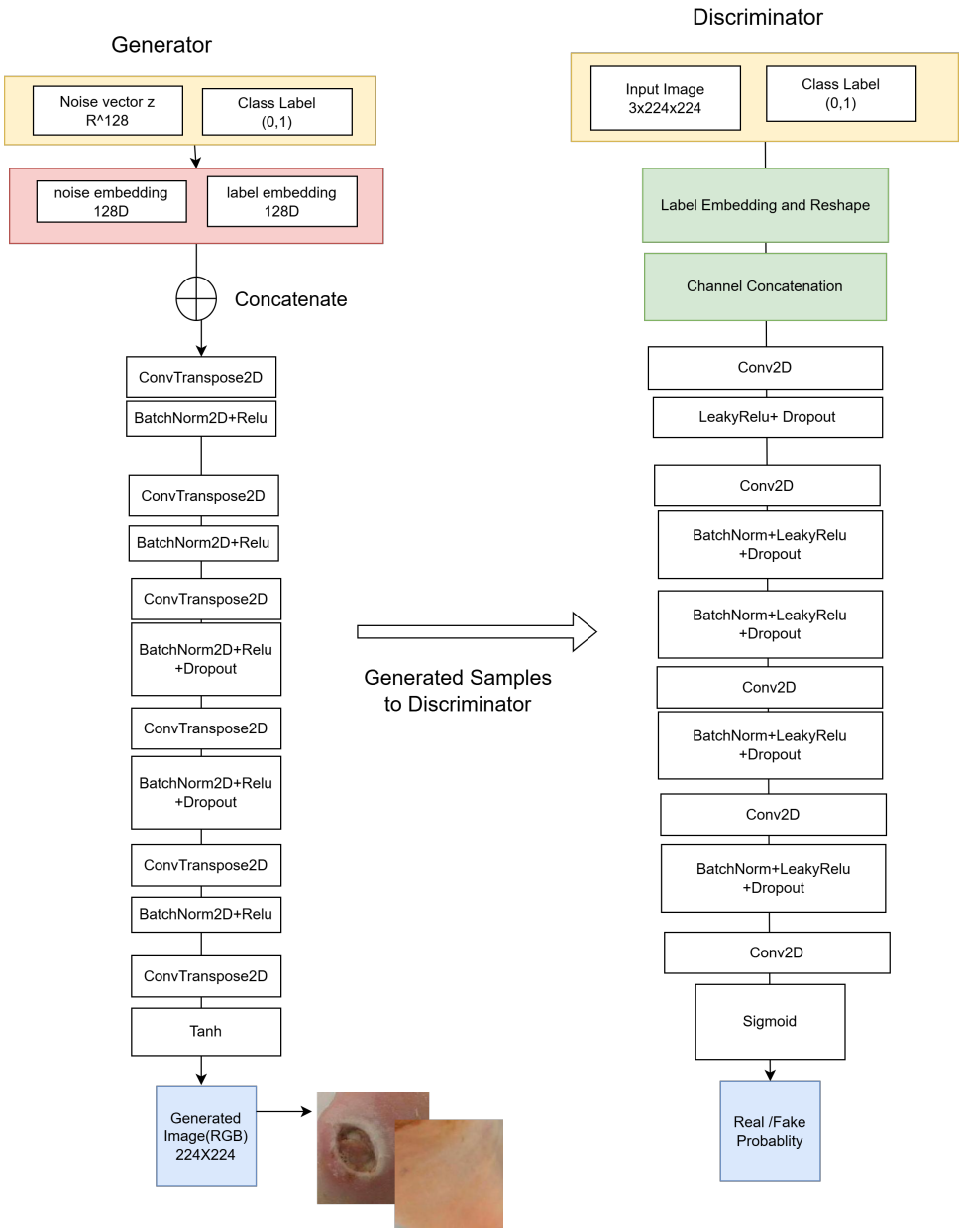

Figure 3.4: Proposed Enhanced Conditional GAN Framework for Data Augmentation

An Enhanced Conditional GAN (cGAN) architecture in Figure 3.4 is designed for generating synthetic medical images with class conditioning.

- **Input Processing and Embedding:** The random noise vector $z \in \mathbb{R}^{128}$ serves as the stochastic foundation of the generation process. This vector is sampled from a standard normal distribution N(0,1), ensuring that each generation instance produces unique

outputs even with identical class labels. The 128 dimensional space provides sufficient entropy to capture the natural variations present in real medical images, enabling the generator to produce diverse synthetic samples rather than memorizing specific patterns. Let the input to the generator be a random noise vector:

$$z \sim \mathcal{N}(0, I_{128}), \quad \text{where } z \in \mathbb{R}^{128}$$

The input to the generator typically begins with a latent variable denoted as z. This latent vector z is sampled from a multivariate normal distribution, specifically $\mathcal{N}(0, I_{128})$, where the mean vector is zero and the covariance matrix is the identity matrix of size 128×128 (I_{128}). This indicates that each component of z is independently drawn from a standard normal distribution with a variance of $\sigma^2 = 1$. As a result, z lies in the 128-dimensional real number space, $\mathbb{R}^{128}$, which serves as the latent space from which the generator learns to map noise into meaningful, structured, and label-conditioned image outputs. The choice of this distribution ensures a smooth and continuous latent space, allowing the generator to produce diverse yet coherent variations of synthetic images during training.

The noise vector undergoes a linear embedding transformation:

$$z_{\text{embedded}} = \text{Linear}(z) \in \mathbb{R}^{128}$$

$$z_{\text{embedded}} = W_z \cdot z + b_z$$

The latent vector $z \in \mathbb{R}^{128}$, sampled from a normal distribution, is first transformed using a learnable weight matrix $W_z \in \mathbb{R}^{128\times128}$ and bias vector $b_z \in \mathbb{R}^{128\times1}$. This linear transformation, given by $z' = W_z z + b_z$, adapts the noise input into a more meaningful representation for the generator to begin the image synthesis process.

- **Class Label Conditioning (Binary Classification):**
  The class label $c \in \{0, 1\}$ represents the target medical condition where $c = 0$: Abnormal (presence of diabetic foot ulcer) , $c = 1$: Normal (no diabetic foot ulcer)

  This discrete label is transformed into a learnable continuous representation through an embedding layer:

  The class label $c \in \{0, 1\}$ is embedded into a continuous vector space as follows:

$$c_{\text{embedded}} = \text{Embedding}(c) \in \mathbb{R}^{128}$$

$$c_{\text{embedded}} = E[c], \quad \text{where } E \text{ is the embedding matrix of shape } 2 \times 128$$

The embedding layer learns class-specific feature representations during training, capturing the semantic differences between normal and abnormal conditions. This allows the generator to understand what visual characteristics distinguish healthy from pathological diabetic foot images.

- **Feature Fusion and Concatenation:** The latent vector embedding $z_{\text{embedded}} \in \mathbb{R}^{128}$ and the class label embedding $c_{\text{embedded}} \in \mathbb{R}^{128}$ are concatenated to form a unified 256-dimensional representation. This is expressed as:

$$h_0 = \text{Concat}(z_{\text{embedded}}, c_{\text{embedded}}) \in \mathbb{R}^{256}$$

Here, h_0 denotes the initial hidden state vector fed into the generator network, and the `Concat` operation refers to the concatenation of both embeddings along the feature dimension. This fused vector serves as a rich, condition-aware input that guides the generator in producing class-specific synthetic images.

This 256-dimensional hybrid vector encapsulates two key components essential for conditional image generation: **stochastic information**, derived from the noise vector z, which introduces diversity into the generated samples; and **semantic information**, derived from the class label c, which ensures that the generated image aligns with the specified condition. By combining randomness and label-driven semantics, this fused representation allows the generator to produce realistic and class-consistent synthetic images.

- **Spatial Upsampling(ConvTranspose2D)**
  This layer performs the critical transition from a vectorial representation to spatial feature maps. The transposed convolution creates the first spatial structure (4×4 grid) while expanding the feature depth to 512 channels. Each channel learns to represent different aspects of the target medical condition.
  The input tensor of shape $256 \times 1 \times 1$ is transformed into an output tensor of shape $512 \times 4 \times 4$ through a transposed convolutional layer. This operation is expressed as:

$$h_1 = \text{ReLU}(\text{BatchNorm}(\text{ConvTranspose2D}(h_0))),$$

where the **ConvTranspose2D** (also known as deconvolution) uses a 4×4 kernel with stride 2 for upsampling, and padding of 1 to preserve spatial dimensions appropriately. The **BatchNorm** layer normalizes feature maps with learnable parameters γ and β, which stabilizes training by reducing internal covariate shift. Finally, the **ReLU** activation

function, defined as $\max(0, x)$, introduces non-linearity and ensures positive activations, which is important for generating images. The resulting hidden state h_1 captures the learned features after this transformation.

- **Feature Refinement (ConvTranspose2D(512 → 1024, kernel=4, stride=2, padding=1): Layer Transformation:**

  This layer doubles the spatial resolution from 4×4 to 8×8 while increasing the feature depth to 1024 channels. At this stage, the generator begins to learn low-level visual patterns such as basic textures and surface characteristics, edge orientations and contours, as well as color gradients and intensity distributions.

- **Final Image Generation ConvTranspose2D**

  The final layer reduces the feature maps to 3 RGB channels at the target resolution of 224×224 pixels. This output layer synthesizes all previously learned spatial and semantic features into a coherent and realistic medical image that resembles true DFU data.

- **Tanh Activation Function**

  The final layer of the generator applies the `Tanh` activation function to normalize the output pixel values to the range $[-1, 1]$. This normalization helps in stabilizing the training process and ensures that the generated images have pixel intensities compatible with the scaled input data. Mathematically, the Tanh function is defined as

  $$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}},$$

  which smoothly maps any real-valued input to the bounded interval $[-1, 1]$, enabling the network to generate images with balanced color and intensity distributions.

- **Generator and Discriminator Summary**

  The Conditional GAN (cGAN) architecture consists of two adversarial components: the generator and the discriminator. The generator takes as input a random noise vector ($z \in \mathbb{R}^{128}$) and a class label ($c \in \{0, 1\}$), which are embedded and concatenated into a 256-dimensional vector. This vector is progressively transformed through a series of transposed convolutional layers that upsample the spatial resolution while integrating class-specific semantic information. As the resolution increases from 1×1 to 224×224, the generator captures and refines features from global structure to fine-grained details, ultimately producing a 224×224 RGB image scaled to the $[-1, 1]$ range using a `Tanh` activation.

The discriminator, on the other hand, evaluates whether an input image is real or generated, conditioned on the corresponding class label. The label is spatially embedded and concatenated with the input image, forming a multi-channel tensor processed through successive convolutional layers. These layers reduce spatial dimensions and increase depth while employing batch normalization, ReLU activations, and dropout to maintain training stability. The discriminator outputs a single scalar indicating the probability that the input image is authentic, guiding the generator through adversarial feedback to synthesize realistic, class-consistent medical images.

- **Adversarial Loss Functions**

  The training of the Conditional GAN (cGAN) involves optimizing two competing loss functions—one for the generator (G) and one for the discriminator (D). The goal of D is to correctly classify real and generated images conditioned on the class label c, while G aims to generate realistic images that fool D. The generator loss indicates how well the synthetic DFU images can fool the discriminator. A lower loss means the generator is producing more realistic images. It is optimized to ensure high-quality, diverse outputs that the discriminator classifies as real. On the other hand, the discriminator loss measures discriminator's ability to distinguish real DFU images from generated ones. It includes errors from misclassifying both real and fake samples. A stable, decreasing loss suggests the discriminator is effectively guiding the generator toward better image synthesis.

  The loss for the generator (*G_Loss*) and discriminator (*D_Loss*) are defined as:

$$\text{G\_Loss}(z) = -\log(D(G(z))) \tag{3.1}$$

$$\text{D\_Loss}(x, z) = -\log(D(x)) - \log(1 - D(G(z))) \tag{3.2}$$

  where x is a real data sample and z is a random input noise vector.

  Here as the generator G aims to fool the discriminator D by classifying fake data as real, its objective is to achieve $D(G(z)) = 1$, which is equivalent to minimizing the generator loss defined in (3.1). The discriminator loss consists of two parts. Minimizing the first part corresponds to the goal $D(x) = 1$, meaning it correctly identifies real data x as real. Minimizing the second part corresponds to the goal $D(G(z)) = 0$, i.e., correctly identifying fake data $G(z)$ as fake, as illustrated in (3.2).

  These loss functions are optimized alternately during training using an optimizer such as Adam, enabling the generator to produce increasingly realistic class-conditioned images while the discriminator becomes more proficient at distinguishing real from fake inputs.

This adversarial training process shown in Figure 3.4 enables the generation of high-quality synthetic samples that preserve the underlying characteristics of medical images while effectively balancing the dataset by generating normal samples through controlled class-conditional synthesis. This study prioritizes data quality and data balancing, investigating whether carefully balanced synthetic augmentation can achieve competitive performance. This methodology allows us to investigate the impact of synthetic data augmentation on addressing class imbalance in medical image classification tasks while preserving model generalization capabilities to real-world scenarios.

3.3.3 Image Quality analysis

The evaluation of synthetic image quality in medical imaging applications requires rigorous quantitative assessment to ensure that generated images maintain clinical relevance and diagnostic value. In the context of Conditional Generative Adversarial Networks (cGANs) for Diabetic Foot Ulcer (DFU) image synthesis, comprehensive quality analysis serves as a critical validation step that determines whether the generated samples can effectively augment training datasets without introducing artifacts or losing essential pathological features.

Fréchet Inception Distance (FID)

The Fréchet Inception Distance represents one of the most robust metrics for evaluating the quality of generated images by measuring the distance between feature distributions of real and synthetic images in a high-dimensional feature space. FID evaluates the similarity between the real and generated images using feature representations extracted from a pre-trained Inception-v3 network. These features are assumed to follow multivariate Gaussian distributions. The FID[1] is computed as the Fréchet distance between these two distributions:

$\text{FID} = \|\mu_1 - \mu_2\|^2 + \text{Tr}(\Sigma_1 + \Sigma_2 - 2(\Sigma_1\Sigma_2)^{1/2})$

where μ_1, μ_2 are the means of real and generated image features, Σ_1, Σ_2 are the covariance matrices, Tr denotes the trace operator. Lower FID values indicate that the generated images are more similar to the real images in terms of distribution.

Peak Signal-to-Noise Ratio (PSNR)

PSNR provides a pixel-level assessment of image fidelity by quantifying the ratio between the maximum possible signal power and the power of corrupting noise[1]. PSNR measures the reconstruction quality of generated images compared to original images by evaluating pixel-wise differences. It is particularly useful for assessing how well fine details are preserved.

The PSNR is calculated as:

$$\text{PSNR} = 20 \cdot \log_{10}\left(\frac{\text{MAX}_I}{\sqrt{\text{MSE}}}\right)$$

where MAX_I is the maximum possible pixel value of the image (typically 255 for 8-bit images), MSE is the Mean Squared Error between the original and generated images.

Higher PSNR values indicate better image quality and preservation of pixel-level information.

Structural Similarity Index Measure (SSIM)

SSIM evaluates image quality based on the degradation of structural information, considering luminance, contrast, and structural components that align with human visual perception[1]. SSIM evaluates the perceived visual similarity between two images by comparing luminance, contrast, and structural information. It is a perceptual metric that reflects image quality as perceived by the human visual system.

The SSIM is computed using the formula:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

The SSIM index ranges from -1 to 1, where a value of 1 indicates perfect structural similarity between the two images.

3.4 Proposed Deep Neural Network Architecture with Self Attention mechanism

A novel integration of a Convolutional Neural Network (CNN) with a Generative Adversarial Network (GAN), specifically designed to enhance the classification performance for Diabetic Foot Ulcer (DFU) classification is shown in Figure 3.5. This hybrid framework is strategically developed to address two of the most persistent challenges in medical image analysis: limitations in feature extraction due to deep network complexity, and severe class imbalance in clinical datasets. Firstly, while CNNs are highly effective for extracting spatial features from medical images, increasing their depth to improve representational power often introduces complications such as vanishing gradients and overfitting. These issues can degrade model performance, especially when training on small or noisy datasets common in medical imaging. To counter this, the CNN component in the proposed model is carefully optimized with architectural enhancements and regularization techniques to ensure robust and stable feature learning, even in deeper layers. Secondly, real-world DFU datasets, such as those used

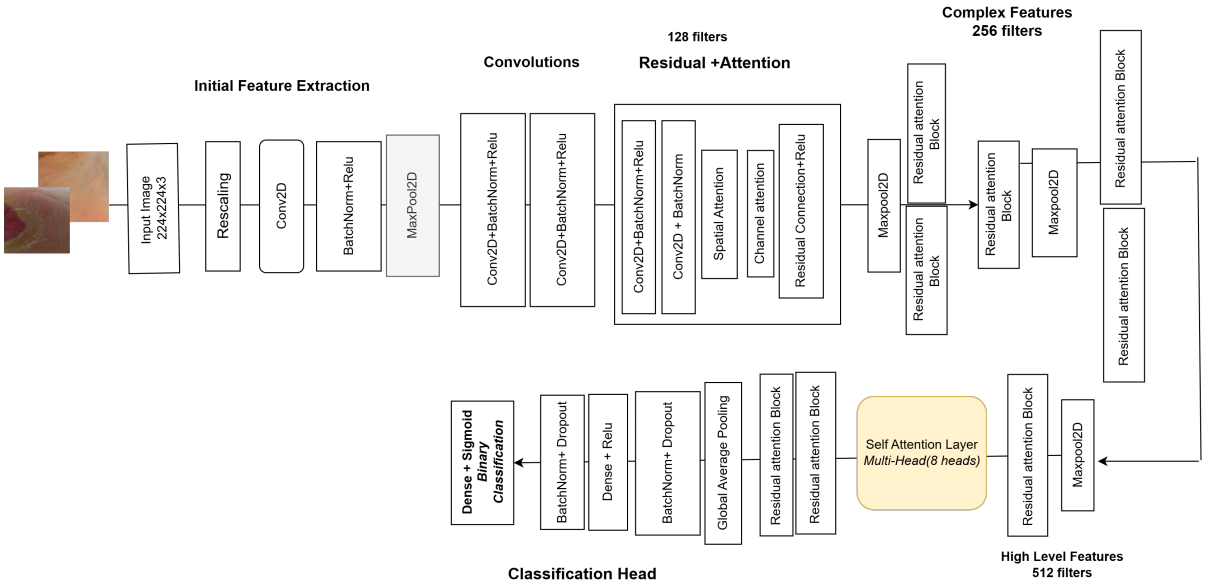

Figure 3.5: Proposed CNN model with Integrated Self-Attention Mechanisms

in clinical research, frequently suffer from a disproportionate number of abnormal cases compared to normal ones. This class imbalance skews the model's learning process, causing it to favor the majority class and reducing its ability to correctly classify minority class instances potentially leading to missed diagnoses. To mitigate this, the GAN component is employed to generate high-quality synthetic images, particularly for the underrepresented normal class. These generated images not only augment the dataset but also introduce new, realistic variations that improve the generalization capabilities of the classifier.

The proposed neural network, is a deep neural network architecture designed to efficiently process DFU images for classification tasks. It emphasizes deep convolutional layers, skip connections, bottleneck layers, and dense connections to optimize feature extraction while maintaining computational efficiency. Skip connections are skillfully integrated to allow gradients to travel efficiently across layers, mitigating the vanishing gradient problem and enhancing the network's ability to capture long-range dependencies. Dense connections further facilitate seamless information flow between layers, improving gradient propagation and feature reuse.By combining advanced architectural components such as feature extraction blocks, residual layers, dense connections, and multi-filter utilization, our proposed model is capable of capturing a rich spectrum of features across varying spatial resolutions. This multi-scale feature representation allows the model to effectively recognize both fine-grained details (such as subtle tissue textures or lesion boundaries) and broader contextual patterns (like spatial distribution and shape of ulcers).

The integration of residual layers not only facilitates deeper network training by mitigating the vanishing gradient problem but also enhances the model's capacity to extract complex

features without loss of important information. These residual connections, placed at critical depths of the network, significantly contribute to feature refinement and gradient flow, enabling more stable convergence during training. Additionally, they improve model transparency and interpretability, which is especially valuable in medical imaging applications where understanding the reasoning behind predictions is essential for clinical trust. Overall, the design encourages a robust and explainable learning process, making the model well-suited for real-world DFU classification tasks.By combining the strengths of CNNs in feature extraction with the data generation capabilities of GANs, the hybrid model improves both the diversity of training data and the accuracy of classification results.

Table 3.2: Proposed CNN-Attention Architecture Overview

Component	**Layers**	**Key Features**
Input Processing	3	Conv2D, Batch Normalization, Resize operations
Feature Extraction	4	Stacked Conv2D blocks with MaxPooling for down-sampling
Attention Modules	3	Includes Spatial Attention, Channel Attention, and Residual connections
Self-Attention	1	Multi-head (8-head) self-attention layer for contextual feature learning
Complex Features	2	Residual connections combined with attention mechanisms
Classification	4	Fully Connected layers, Dropout for regularization, and Global Average Pooling

Table 3.2 presents an overview of the proposed CNN-Attention architecture, detailing its main components, the number of layers in each component, and their key features.

- **Input Processing:** Comprises 3 layers responsible for initial data preparation, including convolutional operations (Conv2D), batch normalization to stabilize learning, and resizing to standardize input dimensions.
- **Feature Extraction:** Contains 4 layers organized as stacked Conv2D blocks, which progressively extract hierarchical features from the input. MaxPooling operations are applied to reduce spatial dimensions and focus on salient information.
- **Attention Modules:** Includes 3 layers incorporating different attention mechanisms—spatial attention to focus on relevant regions, channel attention to weigh feature maps, and residual connections to facilitate gradient flow and improve learning.

- **Self-Attention:** A single multi-head self-attention layer with 8 attention heads, enabling the model to capture long-range dependencies and contextual relationships within the feature maps.
- **Complex Features:** Consists of 2 layers combining residual connections with attention mechanisms to refine and enhance complex, high-level features before classification.
- **Classification:** Composed of 4 layers including fully connected (dense) layers for decision making, dropout layers for regularization to prevent overfitting, and a global average pooling layer to aggregate spatial information into feature vectors.

Description of CNN Architecture for DFU Classification

The transition from data augmentation to CNN-based classification represents a critical phase in the DFU analysis pipeline. After generating high-quality synthetic images through cGAN-based augmentation, the enriched dataset serves as the foundation for training robust and generalizable classification models. This process transforms the challenge of limited medical data into an opportunity for enhanced diagnostic accuracy through comprehensive feature learning.

Input Image

The input data consists of RGB color images representing raw DFU (Diabetic Foot Ulcer) samples. These images are preprocessed to ensure a standardized input size across the entire dataset, which facilitates consistent feature extraction and promotes stable model training. Additionally, the pixel values are normalized to either the $[0, 1]$ or $[-1, 1]$ range, depending on the activation function used in the final layer of the model. This normalization step is essential for maintaining numerical stability during the training process.

Initial Feature Extraction Block

This block standardizes and extracts foundational features from the input DFU images through a sequence of preprocessing and convolutional operations.

- **Layer 1: Resizing**
  All input images, originally of variable sizes, undergo a preprocessing step that includes resizing to a fixed dimension of $224 \times 224 \times 3$. This operation ensures uniformity across the dataset, which is critical for consistent model training and evaluation. By standardizing the input size, the network can effectively learn spatial hierarchies without being affected by irregular input dimensions.

- **Layer 2: Convolution (Conv2D)**
  The first convolutional layer uses a large 7×7 kernel with a stride of 2 and same padding, applied to input images of size $224 \times 224 \times 3$. This layer consists of 64 filters and is designed to capture low-level visual features such as edges, textures, and shape contours. The stride of 2 effectively reduces the spatial resolution while preserving important pattern information, thereby allowing the network to process input efficiently in the subsequent layers.

- **Layer 3: Batch Normalization + ReLU Activation**
  Following the convolutional layer, batch normalization is applied to normalize the activations across the batch, which mitigates internal covariate shift and facilitates faster convergence during training. Subsequently, the ReLU activation function, defined as $f(x) = \max(0, x)$, is used to introduce non-linearity into the model. This non-linear transformation enables the network to learn complex representations and helps in preventing issues such as vanishing gradients.

- **Layer 4: MaxPool2D**
  A max pooling operation with a pooling size of 3×3, stride of 2, and same padding is applied to downsample the feature maps. This process reduces the spatial dimensions while retaining the most prominent features, thereby introducing translational invariance. Additionally, it lowers the computational complexity of the model, which is particularly beneficial when processing high-resolution medical images such as those used in diabetic foot ulcer analysis.

Multi-Scale Convolution Block (Layers 5–7)

This block employs parallel convolutional layers with varying kernel sizes -3×3, 5×5, and 7×7 - to capture features at multiple spatial scales.

- **3×3 Convolution:** Extracts fine-grained local features such as micro-lesions, sharp boundaries, and subtle texture variations, which are critical for identifying early signs of DFU.

- **5×5 Convolution:** Detects medium-scale patterns including tissue transitions, wound morphology, and regional irregularities. Provides a balance between detail and context.

- **7×7 Convolution:** Captures large-scale structural information such as the overall foot shape, widespread ulceration areas, and contextual tissue relationships. Useful for holistic analysis.

All three branches receive the same input of size $56 \times 56 \times 64$ and produce outputs of size $56 \times 56 \times 128$. The resulting multi-scale feature maps are later fused to enhance the network's ability to analyze DFU images at various levels of granularity.

Residual Attention and Feature Integration Module (Layers 8–13)

This module enhances the network's ability to focus on clinically significant regions of Diabetic Foot Ulcers (DFUs) through a sequence of residual attention blocks and downsampling layers:

The network architecture continues with a series of attention-enhanced residual blocks and pooling layers that progressively abstract and refine features. **Layer 8** introduces a Residual Attention Block that integrates residual connections with both spatial and channel attention mechanisms. This design allows the model to selectively focus on the most relevant regions of the input while preserving the overall global context. In **Layer 9**, a combination of Batch Normalization and ReLU activation is applied to the attention-refined features, facilitating stable gradient propagation and efficient learning in deep architectures.

Layer 10 performs max pooling, reducing the spatial dimensions of the feature maps to $28 \times 28 \times 128$, which not only improves computational efficiency but also expands the receptive field of subsequent layers. **Layer 11** employs another Residual Attention Block, this time with 256 filters, to enhance complex feature extraction at a reduced spatial resolution. This layer is particularly effective at capturing inter-class variations and localized pathology patterns relevant to diabetic foot ulcers (DFUs).

Following this, **Layer 12** applies another max pooling operation to further downsample the feature maps, concentrating the information for higher-level abstraction. Finally, **Layer 13** consists of an additional Residual Attention Block, maintaining 256 filters to refine previously learned features. This layer reinforces attention on clinically significant structures, such as ulcer boundaries and infected regions, while suppressing background noise and less informative details.

This module strengthens the network's ability to integrate multi-scale, context aware, and attention modulated features vital for accurate and robust DFU classification.

Self-Attention Mechanism

Self-attention is a powerful mechanism that enables a neural network to dynamically assign weights to different parts of the input based on their relevance to each other. Unlike traditional convolutional operations that process information locally within a limited receptive field, self-attention allows each position in the input sequence or image to attend to all other positions, regardless of spatial proximity.

This mechanism is particularly effective in capturing global contextual information and long-range dependencies, which are often critical for understanding complex spatial relationships in image data. By computing attention scores between all pairs of positions, the model can learn how features at distant locations interact and contribute to the overall representation.

In the context of image analysis, and specifically for Diabetic Foot Ulcer (DFU), self-attention helps in modeling spatial dependencies across the entire foot image. For instance, pathological features near the toes may have diagnostic relevance when analyzed in conjunction with features around the heel. Traditional CNNs may struggle to capture such non-local interactions due to limited receptive fields, but self-attention directly addresses this limitation.

In summary, self-attention significantly enhances a model's capability to understand and reason over global feature interactions, making it particularly valuable in medical image analysis tasks requiring high precision and interpretability.

The Multi-Head Self-Attention (MHSA) module is applied to capture long-range dependencies and enhance contextual understanding within the feature maps. It utilizes 8 attention heads and operates on an embedding dimension of 512. The input to this module is a feature map of size $7 \times 7 \times 512$, which is obtained by applying pooling on the output of preceding convolutional blocks. The MHSA mechanism processes this input to produce an output of the same shape, $7 \times 7 \times 512$, but enriched with globally contextualized feature representations. The Multi-Head Attention mechanism enhances the model's ability to understand complex spatial relationships in medical images. In this module, the input feature map is divided into 8 distinct attention heads, with each head learning a unique type of interaction between different regions of the image. This architectural design enables diverse pattern recognition and enriches feature representation. Unlike traditional convolutions that capture only local context, multi-head attention facilitates global context modeling by allowing each spatial position to attend to every other position across the image. This is achieved through the self-attention mechanism, where the same input is used as the query, key, and value, enabling computation of attention scores that reflect the importance of each location relative to the rest. From a medical perspective, this capability is particularly significant for diabetic foot ulcer (DFU) analysis, where pathological features may be spatially distant. By capturing long-range dependencies, the model is better equipped to perform holistic assessments and detect clinically relevant relationships across the wound region.

High-Level Feature Processing

Feature Extraction and Refinement (512 filters): At this stage, the model operates on 512 high-dimensional feature maps that encapsulate rich and abstract representations of the input data. These features capture complex patterns, including subtle textural cues and semantic structures

critical for accurate classification of diabetic foot ulcer conditions.

Final Pooling and Normalization: To prepare the extracted features for the final classification layer, global average pooling is applied, effectively reducing the spatial dimensions and converting the feature maps into compact feature vectors. This is followed by batch normalization, which stabilizes the final feature representations, improves generalization, and ensures that the inputs to the classifier are well-conditioned for learning.

Classification Head

- **Multi-Layer Classification Network:** The classification head of the model consists of a series of fully connected (dense) layers designed to progressively distill and interpret the high-level features extracted from previous layers. It begins with a dense layer of 256 units for initial classification processing, followed by batch normalization and dropout for regularization and training stability. Subsequent dense layers with 128 and 64 units serve to further compress and refine the learned representations. A global average pooling layer is then applied to spatially average the feature maps, reducing them to a format suitable for final classification. This is followed by a dense layer of 32 units, which performs pre-classification transformations to prepare the features for the output layer.

- **Classification Strategy:** The architecture employs progressive dimensionality reduction through successive dense layers, allowing the network to efficiently compress feature information while retaining the most discriminative characteristics. Dropout layers are incorporated as a form of regularization, randomly deactivating neurons during training to mitigate overfitting and promote better generalization to unseen data. By structuring the classifier as a multi-stage decision-making process, the network is capable of learning complex, non-linear boundaries in the feature space. This layered transformation enhances the model's ability to accurately distinguish between subtle variations in diabetic foot ulcer images.

The classification head employs a multi-layer dense network designed for progressive feature compression and robust decision-making. It begins with a 256-unit dense layer followed by batch normalization and dropout for regularization. Subsequent dense layers with 128 and 64 units further refine the features, while a global average pooling layer aggregates spatial information. A final 32-unit dense layer prepares the features for classification. This architecture balances dimensionality reduction with regularization, enabling complex and accurate classification boundaries. The model offers robust diagnostic capabilities by performing multi-class classification to distinguish different diabetic foot ulcer (DFU) stages and types, assessing wound severity and healing progress, and supporting treatment planning through objective clinical assessments. Technically, it achieves high accuracy by leveraging multi-scale features and

attention mechanisms, ensuring robustness against variations in image quality and providing interpretability for model decisions. Practically, the architecture is computationally efficient, balancing accuracy with speed, scalable to various medical imaging tasks, and designed for seamless integration into clinical workflows following proper validation.

This CNN architecture represents a unique approach to DFU classification, combining multiple advanced techniques to achieve robust and accurate performance. The integration of multi-scale convolutions, attention mechanisms, and residual learning creates a powerful tool for automated DFU analysis, potentially improving accuracy and workflow efficiency.

Model Training Hyperparameters:

Table 3.3 lists the key hyperparameters along with the selected values used during training and testing of both the generator and discriminator networks.

Table 3.3: Hyperparameters Used for CNN based self-attention Model Training

Hyperparameter	**Value**
Batch Size	32
Image Size (Height × Width)	224×224
Number of Classes	2
Epochs	200
Dropout	0.3
No. of layers	17
Learning Rate	0.0001
Loss Function	Binary Cross Entropy (BCELoss)
Optimizer (Generator & Discriminator)	Adam ($\beta_1 = 0.5, \beta_2 = 0.999$)

In the experimentation phase, several hyperparameters were fine-tuned to optimize the training performance and ensure stable convergence of the generative adversarial network. The batch size of 32 determines the number of samples processed before the model's parameters are updated, balancing training speed and gradient estimation accuracy. The image size was fixed at 224×224 pixels to standardize input dimensions for the networks, facilitating consistent feature extraction. With two classes defined, the model performs binary classification, distinguishing between real and synthetic images. Training was conducted over 200 epochs, allowing sufficient iterations for the generator and discriminator to improve iteratively. A dropout rate

of 0.3 was applied to reduce overfitting by randomly deactivating neurons during training. The architecture consists of 17 layers, providing sufficient depth to learn complex representations without excessive computational cost. The learning rate of 0.0001 controls the step size during optimization, enabling gradual and stable updates to the model weights. For the loss function, Binary Cross Entropy (BCELoss) was used to measure the discrepancy between predicted and true labels, a standard choice for binary classification tasks. Both the generator and discriminator were optimized using the Adam optimizer with parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$, which adaptively adjusts learning rates and helps stabilize GAN training by smoothing the parameter updates.

Having detailed the architecture design, data preprocessing, augmentation strategy, and training process in this chapter, the next chapter presents the performance results and analysis of the proposed model. It evaluates the classification effectiveness using standard metrics and compares the model against existing state-of-the-art approaches.

Chapter 4

Performance Results and Analysis

4.1 Experimental Setup

4.1.1 Hardware Configuration

The experiments were conducted on both local and cloud-based hardware platforms:

- **Local Machine (Dell G15 5530):**
  - CPU: 13th Gen Intel Core i5-13450HX (16 threads, 2.4 GHz)
  - GPU: NVIDIA GeForce RTX 3050 (4 GB VRAM)
  - RAM: 16 GB
  - Storage: 512 GB SSD
- **Cloud Environment (Kaggle GPU):**
  - GPU: NVIDIA Tesla P100 (16 GB VRAM)
  - RAM: 13 GB
  - Temporary Storage: 20 GB

4.1.2 Software Environment

- **Operating System:** Windows 11 Home 64-bit (Build 26100)
- **Programming Language:** Python 3.13.2
- **Development Platform:** Kaggle Jupyter Notebook
- **Libraries Used:** TensorFlow, Keras, PyTorch, NumPy, Matplotlib, OpenCV, Scikit-learn

4.2 Class Distribution in Part A Dataset

This work utilizes the DFUC2021 dataset, which contains clinical image samples of Diabetic Foot Ulcers (DFUs) collected from multiple patients. The dataset is divided into two parts: Part A and Part B. For this study, we focus on Part A to train and evaluate models for the binary classification task distinguishing DFU-affected skin patches from normal skin patches. The DFUC2021 dataset is shown in Table 3.1 contains 1,038 abnormal(ulcer) and 641 normal samples(non-ulcer), reflecting a significant class imbalance.

4.2.1 Class Distribution in Dataset after augmentation

Table 4.1: DFUC2021 Dataset: Impact of Data Augmentation

Data Type	**Class**	**Samples Before**	**Samples After**
PartA_DFU	Abnormal	1038	1038
	Normal	641	1024

After applying data augmentation using the enhanced cGAN model, the dataset achieved a near-balanced distribution between the two classes ulcer (abnormal) and non-ulcer (normal). Originally, the dataset was significantly imbalanced, with only 641 samples in the normal class compared to 1038 in the abnormal class as shown in figure 4.1. This imbalance could lead to biased model training, where the classifier favors the majority class, resulting in poor accuracy for the minority class.

To address this, we synthetically generated additional normal class samples using our cGAN-based augmentation pipeline, increasing the count from 641 to 1024. This brought the number of samples in both classes to a comparable level as shown in Figure 4.2, minimizing class imbalance without over-relying on synthetic data.

Balancing the dataset in this manner improves the model's ability to learn discriminative features for both classes equally. As a result, the classification performance particularly in terms of recall and F1-score for the minority class shows significant improvement. Figure **??** visually demonstrates the class distribution before and after augmentation, confirming the effectiveness of this approach in achieving dataset balance and enhancing model robustness.

Train-Test Split

To evaluate the performance and generalization capability of the proposed model, the dataset from Part A of DFUC2021 containing two folders: Abnormal (ulcer) and Normal (non-ulcer) was used. Prior to splitting, data augmentation technique was applied to increase diversity and handle imbalance.

The dataset was then divided into training and validation subsets using a 70-30 split ratio. Specifically, 70% of the data was allocated for training the model, while the remaining 30% was reserved for validation. This stratified split ensures balanced representation from both classes in each subset, enabling an unbiased evaluation of the model's classification performance.

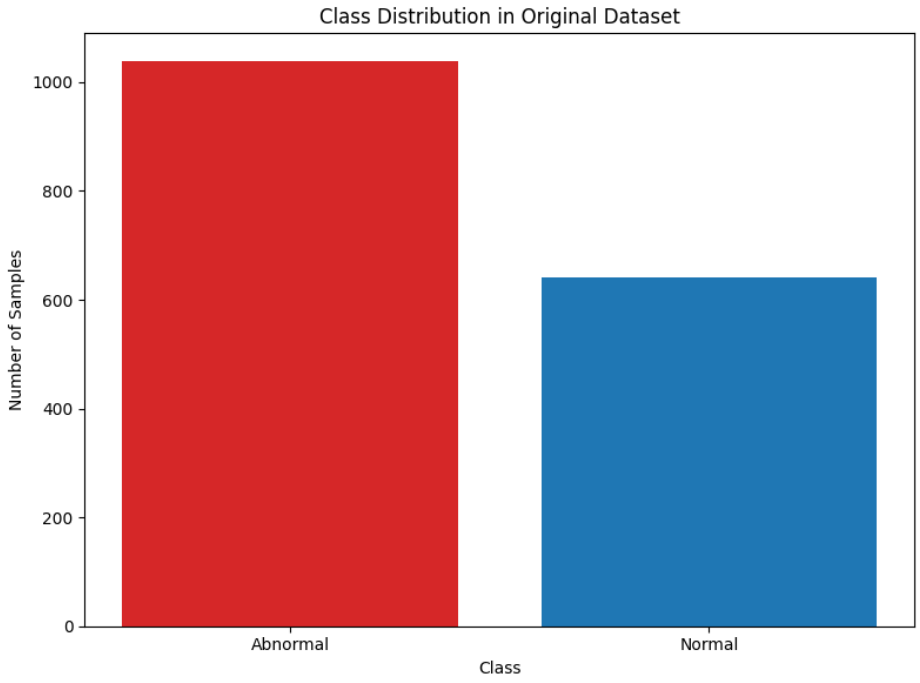

Figure 4.1: Class Distribution before augmentation

4.2.2 Visualization of Class Distribution

Figures 4.1 and 4.2 illustrate the class distribution before and after applying data augmentation to address the imbalance in the dataset.

- **Class Distribution Before Augmentation**
  The original distribution of samples across the two classes normal (non-ulcer) and abnormal (ulcer) is shown in Figure 4.1 .A noticeable class imbalance is evident, with the abnormal class (1038 samples) significantly outnumbering the normal class (641 samples). This disproportionate distribution can negatively affect the model's learning, leading to biased predictions favoring the majority class.

- **Class Distribution After Augmentation**
  The improved class distribution following data augmentation using the enhanced cGAN model is shown in Figure 4.1. The number of normal class samples is increased to 1024,

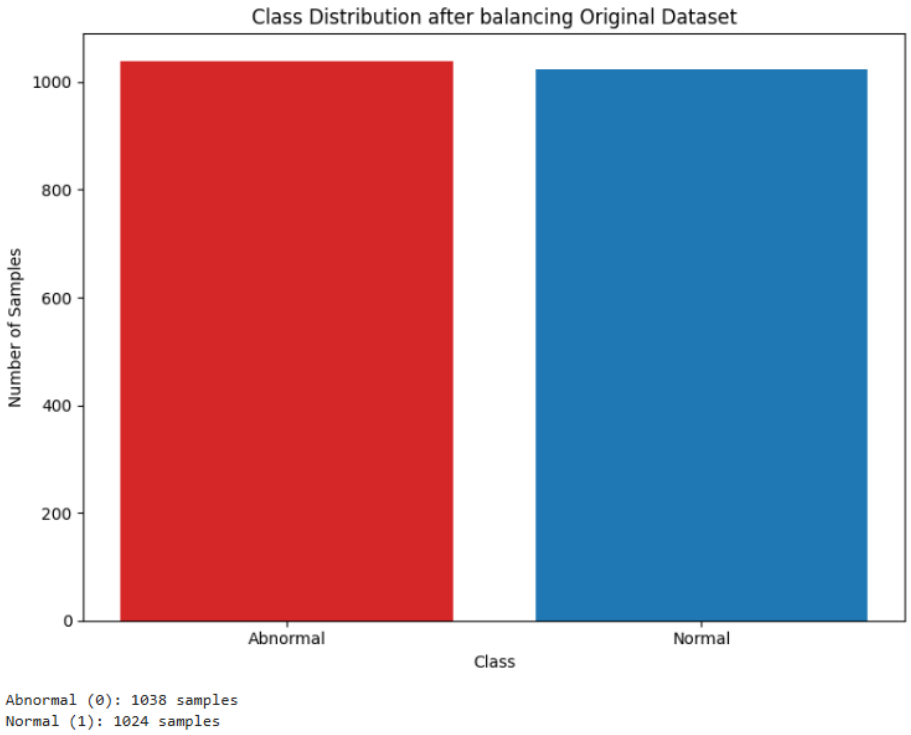

Figure 4.2: Class Distribution after augmentation

bringing it nearly equal to the abnormal class. This balanced distribution ensures that the model is trained on a more representative dataset, improving its ability to generalize and perform equally well on both classes.

Together, these visualizations clearly demonstrate the effectiveness of the augmentation process in achieving dataset balance, which is crucial for enhancing the fairness, accuracy, and robustness of the classification model.

4.3 Experimental Results

4.3.1 Discriminator Accuracy Analysis:

As shown in Figure 4.3, the discriminator's accuracy trends throughout training provide valuable insights into the evolving balance between the generator and discriminator. Initially, the discriminator achieves high accuracy as the generator produces low-quality synthetic images. However, as training progresses and the generator improves, generating more realistic outputs, the discriminator's task becomes increasingly difficult resulting in fluctuations in its accuracy. In our experiments, the discriminator stabilizes around an accuracy of approximately 76%. This stabilization suggests that the discriminator is still effectively identifying fake samples, while also being appropriately challenged by the generator. The generator, under these conditions,

learns to produce high-quality synthetic images that closely resemble real samples. However, there is still room for improvement in generating even more diverse and photorealistic synthetic images. Further enhancement could benefit downstream tasks such as classification and segmentation by providing richer training data.

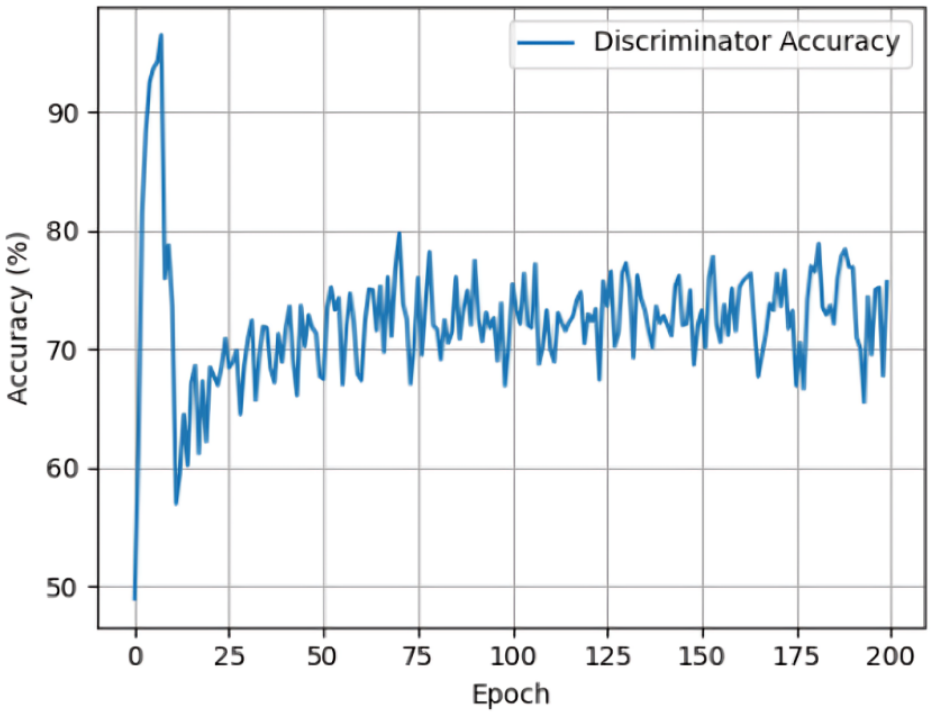

Figure 4.3: Discriminator Accuracy Curve

4.3.2 Image Quality Metrics:

Quantitative evaluation of the quality and diversity of the synthetic images generated by the enhanced cGAN model is shown in Table 4.2. The evaluation employs three widely recognized image quality metrics: Fréchet Inception Distance (FID), Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). These metrics provide complementary perspectives on image quality PSNR focusing on low-level pixel-wise accuracy, and SSIM[1, 6] capturing perceptual fidelity from a human visual system perspective.

Fréchet Inception Distance (FID): FID quantitatively measures the distance between the distributions of generated images and real images in a deep feature space, usually extracted from a pretrained Inception network[1]. Lower FID scores indicate that the generated images are more similar to the real images in terms of both quality and diversity. Unlike PSNR and SSIM, which are pairwise comparison metrics, FID evaluates the overall statistical similarity of the datasets, capturing mode collapse and diversity aspects effectively.

Peak Signal-to-Noise Ratio (PSNR): PSNR is a classical metric used to evaluate the fidelity of a reconstructed or synthesized image with respect to a reference ground truth image[1]. Higher PSNR values indicate that the generated image is very close to the original, with lower levels of noise or distortion. In the context of our experiment, consistently high PSNR values suggest that the synthetic images maintain strong pixel-level resemblance to real DFU images, confirming the generator's capacity to learn fine-grained visual structures.

Structural Similarity Index Measure (SSIM): SSIM evaluates image similarity by comparing three perceptual components: luminance, contrast, and structural information. Unlike PSNR, which considers only pixel-wise error, SSIM reflects the way humans perceive image quality. It ranges from -1 to 1, with values closer to 1 indicating greater similarity. In our analysis, high SSIM scores validate that the synthetic images preserve not only structural integrity but also the visual appearance of real medical images, which is crucial for downstream tasks such as classification and diagnosis.

Table 4.2: Image Quality Analysis

GAN Method	**FID ↓**	**PSNR ↑**	**SSIM ↑**
CGAN	42.40	13.01	0.176
Enhanced CGAN Model	32.14	18.32	0.451

Table 4.2 compares the image quality and diversity metrics between the baseline Conditional GAN (CGAN) and the proposed Enhanced CGAN model. Three key evaluation metrics are considered: Fréchet Inception Distance (FID), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM).

The FID score of the Enhanced CGAN model is significantly lower (32.14) compared to the baseline CGAN (42.40). Since a lower FID indicates a closer match between the distributions of generated and real images, this improvement demonstrates that the enhanced model produces synthetic images that are more realistic and diverse. Similarly, the PSNR value increases from 13.01 dB in the baseline to 18.32 dB in the Enhanced CGAN, indicating that the generated images have better pixel-wise reconstruction quality and less noise. SSIM score, which measures perceptual similarity based on luminance, contrast, and structure, also improves notably from 0.176 to 0.451. This suggests that the Enhanced CGAN better preserves important structural and textural information in the synthetic images, making them more visually consistent with real images.

Figure 4.4: Conditional GAN Augmented Image samples

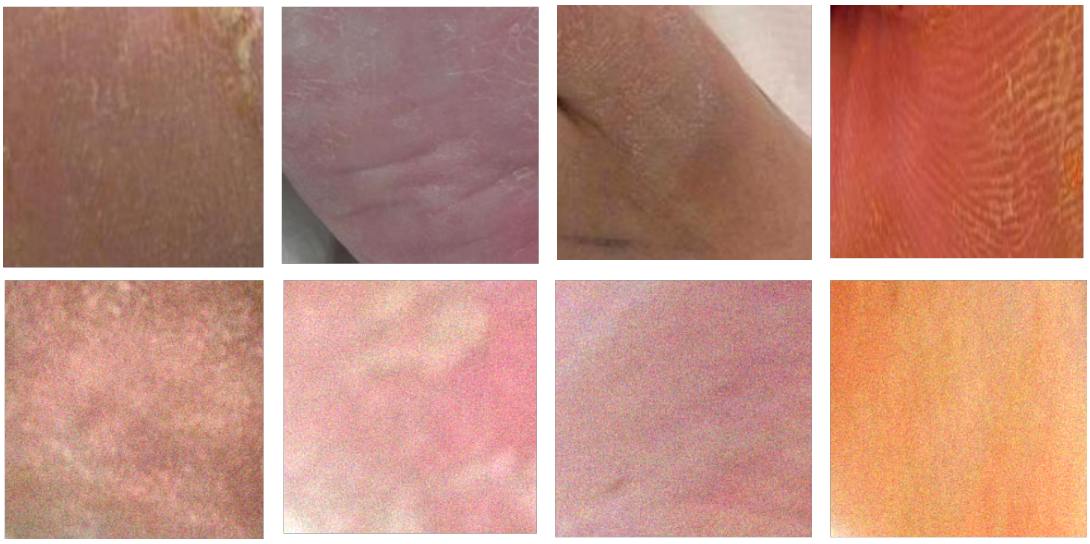

Figure 4.5: Enhanced Conditional GAN Augmented Image samples

Image Quality Improvement

Figure 4.4 shows CGAN augmented normal image samples in second row while the first row shows original images. Figure 4.5 shows Enhanced CGAN augmented normal image samples in second row while first row shows original images. There is progressive improvement in synthetic image quality generated by the Enhanced CGAN model over successive training epochs. The bottom row presents high-quality synthetic images generated after extensive training (epochs 200). These images closely resemble real skin textures and medical features observed in the original dataset, showing smoothness, fine-grain details, and improved realism. This progression confirms that the Enhanced CGAN effectively learns the data distribution and generates visually convincing samples over time.

4.3.3 Model Evaluation

To assess the performance of the proposed DFU classification model, we employed standard evaluation metrics including accuracy, precision, recall, F1-score, specificity and AUC shown in Equation (4.1) (4.2) (4.3) (4.4) (4.3) (4.4) (4.5) respectively. These metrics provide a comprehensive evaluation of the model's capability to correctly distinguish between ulcer (abnormal) and non-ulcer (normal) cases.

Accuracy measures the overall correctness of the model's predictions, reflecting the proportion of total correct predictions (both positive and negative) among all cases [1]. It is calculated as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

In this study, the proposed model achieved an accuracy score of 0.9257, indicating a high level of overall prediction correctness across both classes.

Precision evaluates the accuracy of positive predictions by measuring the proportion of correctly predicted positive cases out of all predicted positives [1]. It indicates how many of the samples labeled as positive are actually positive. It is calculated as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.2}$$

In this study, the model achieved a precision score of 0.9061, demonstrating a strong ability to minimize false positive predictions and reliably identify true positive cases.

Recall, also known as sensitivity, measures the ability of the model to identify all actual positive cases [1]. It is the proportion of correctly predicted positives out of all true positives

and false negatives. It is calculated as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

In this study, the model achieved a recall score of 0.9204, indicating a high capability to correctly detect the majority of positive cases with minimal false negatives.

F1-Score is the harmonic mean of precision and recall, providing a single metric that balances both [1]. It is especially useful in cases of class imbalance, ensuring neither precision nor recall dominates the evaluation. It is calculated as:

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

In this study, the model achieved an F1-score of 0.9373, reflecting a well-balanced performance in terms of both precision and recall, particularly in handling the positive class effectively.

Specificity (also known as the true negative rate) measures the model's ability to correctly identify negative cases [1]. It indicates the proportion of actual negatives that are correctly predicted as negative. It is calculated as:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{4.5}$$

In this study, the model achieved a specificity score of 0.911, indicating strong performance in correctly identifying negative cases and minimizing false positives.

AUC (Area Under the Curve) represents the area under the Receiver Operating Characteristic (ROC) curve [1], which plots the true positive rate (Recall) against the false positive rate (1 - Specificity) as shown in Figure 4.8. AUC provides an aggregate measure of performance across all classification thresholds. A value closer to 1 indicates better overall performance. In this study, the model achieved an AUC score of 0.9431, reflecting good discriminative ability between the positive and negative classes.

4.3.4 Confusion Matrix Analysis:

A confusion matrix is a table used to evaluate the performance of a classification model by comparing the predicted labels with the actual true labels. It provides a detailed breakdown of how well the model is performing across different classes, especially in binary or multiclass

classification problems. In a confusion matrix, True Positives (TP) refer to the number of instances that are correctly predicted as belonging to the positive class. True Negatives (TN) indicate the number of instances accurately classified as belonging to the negative class. On the other hand, False Positives (FP) are the cases where the model incorrectly predicts the positive class when the actual class is negative this is also known as a Type I error. Similarly, False Negatives (FN) represent instances where the model wrongly classifies them as negative, despite them actually being positive, which is referred to as a Type II error. These four values form the core of the confusion matrix and are essential for evaluating the performance of a classification model.

The figure 4.6 demonstrates strong performance on the validation dataset, with 287 true positives correctly identified abnormal cases constituting 56.8% of all samples. It also achieved 176 true negatives (34.9%), correctly identifying normal cases. There were 17 false positives (3.4%) where normal cases were misclassified as abnormal, and 25 false negatives (5.0%) where actual abnormal cases were missed.

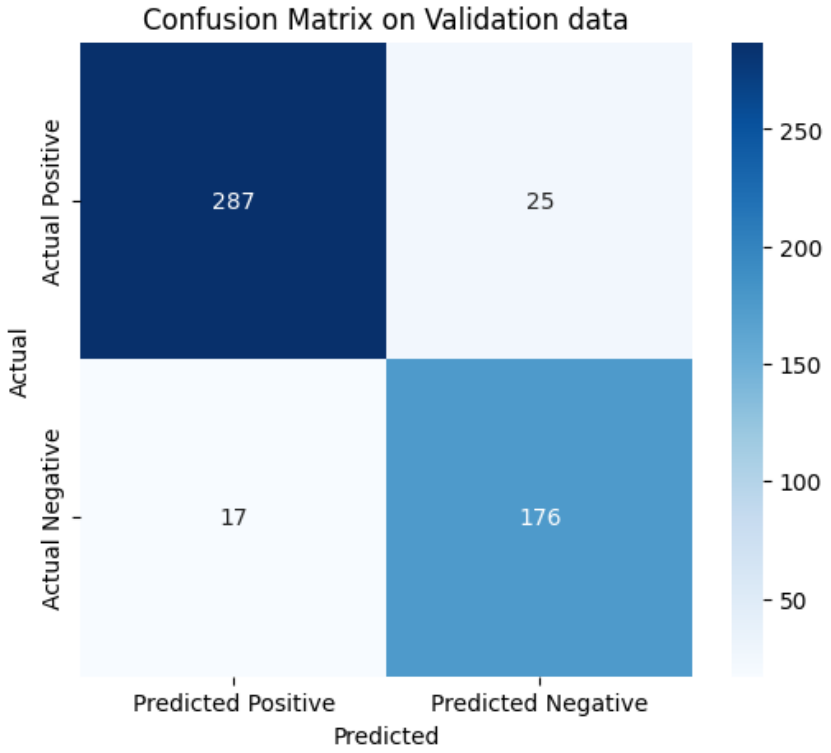

Figure 4.6: Confusion Matrix for binary classification

4.3.5 Heatmap Analysis of Classification Report:

The heatmap shown in Figure 4.7 is a visual representation of data in a matrix format, where individual values are represented by varying color intensities. In the context of a classification

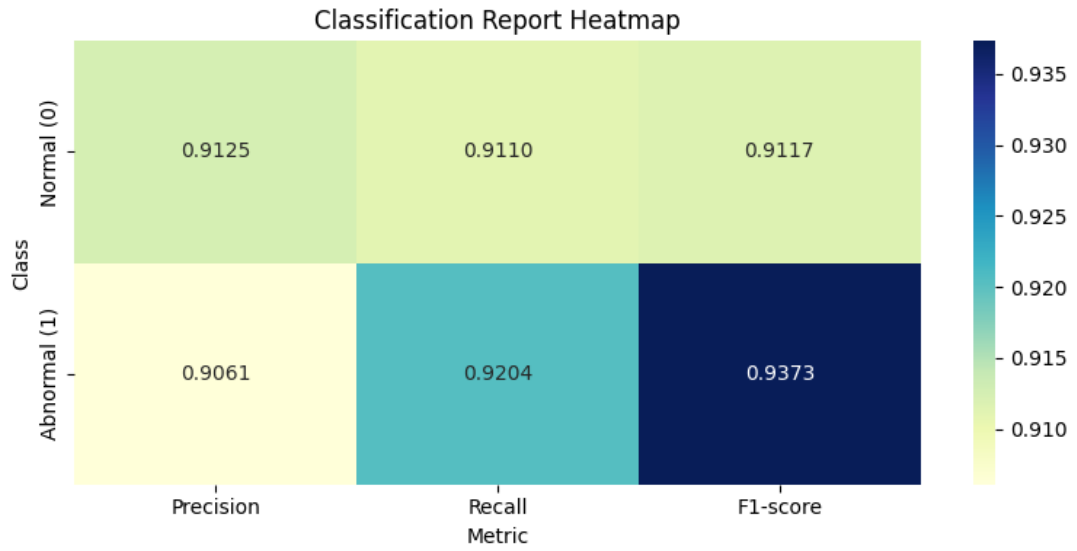

Figure 4.7: Heatmap of Classification Report

report, a heatmap is particularly useful for quickly interpreting how a model performs across different metrics (such as precision, recall, and F1-score) for each class. The heatmap analysis of the classification report offers a clear visualization of the model's performance across key metrics for both classes Normal (Class 0) and Abnormal (Class 1). The model demonstrates consistently strong results for the Normal class, with a precision of 0.9125, recall of 0.9110, and an F1-score of 0.9117, indicating well-balanced and reliable predictions. For the Abnormal class, the performance is even more impressive in certain areas, with a slightly lower precision of 0.9061 but a notably higher recall of 0.9204, suggesting the model is highly effective at correctly identifying abnormal cases. The F1-score for Abnormal (0.9373) is the highest among all metrics, reflecting a strong balance between precision and recall. Overall, the model shows robust performance across both classes, with a particular strength in detecting abnormalities, making it valuable for applications where identifying abnormal instances is critical.

4.3.6 Comparison with State-of-the-Art Models

Table 4.3: Performance Comparison with State-of-the-art models

Models	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**AUC**	**Specificity**
LeNet[13]	0.8752	0.861	0.9012	0.8806	0.9541	0.849
AlexNet[13]	0.8873	0.8924	0.9028	0.8921	0.9603	0.865
GoogleNet[13]	0.9024	0.8857	0.9101	0.9073	0.9685	0.902
EfficientNetB0[9]	0.9251	0.8923	0.9502	0.9204	0.9812	0.894
DenseNet121[9]	0.9186	0.8881	0.9445	0.9154	0.9787	0.887
ResNet101[4]	0.9202	0.8896	0.9470	0.9174	0.9793	0.889
InceptionV3[4]	0.9114	0.8807	0.9362	0.9075	0.9745	0.880
VGG16[4]	0.9069	0.8769	0.9310	0.9032	0.9723	0.874
Proposed Model	**0.9257**	**0.9061**	**0.9204**	**0.9373**	**0.9431**	**0.911**

The performance comparison (Table 4.3) indicates that the Proposed Model achieves superior results across most evaluation metrics when benchmarked against various established convolutional neural networks (CNNs), including LeNet, AlexNet, GoogleNet, EfficientNetB0, DenseNet121, ResNet101, InceptionV3, and VGG16. The models are evaluated across multiple performance metrics such as accuracy, precision, recall, F1 score, AUC (Area Under the Curve), and specificity to ensure a holistic assessment. While the proposed model stands out in performance, it shares several similarities with the state-of-the-art CNN architectures evaluated in this study. Most of the models EfficientNetB0, DenseNet121, ResNet101, InceptionV3, and VGG16 are pretrained on ImageNet, which provides a strong foundation for learning low-level features like edges, textures, and patterns. The proposed model also benefits from this technique, leveraging pretrained convolutional backbones and fine-tuning on DFU data. This similarity helps explain the relatively high accuracy and AUC across models. The other benchmark models selected for comparison include well-known architectures that have demonstrated strong performance in DFU classification. The proposed model demonstrates superior performance in most evaluation criteria. These improvements indicate a strong balance between precision and recall, making the model highly effective at correctly identifying both positive and negative cases. Furthermore, the high AUC value reflects the model's robust discriminative capability in binary classification. Overall, these results demonstrate that the proposed CNN with self-attention architecture provides superior classification performance on the DFU dataset compared to existing state-of-the-art models, making it a promising approach for accurate and reliable diabetic foot ulcer classification.

4.3.7 ROC Curve Analysis

The Receiver Operating Characteristic (ROC) curve shown in Figure 4.8 is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It visualizes the trade off between the true positive rate (sensitivity or recall) and the false positive rate across different threshold settings. This ROC analysis presents a comprehensive comparison of the proposed model with state-of-the-art pretrained models. The ROC curves visualize these trade-offs between sensitivity and specificity. Higher Area Under the Curve (AUC) values indicate superior overall classification performance, with the best-performing models achieving curves that approach the top left corner of the plot, signifying minimal false positives and maximal true positives.

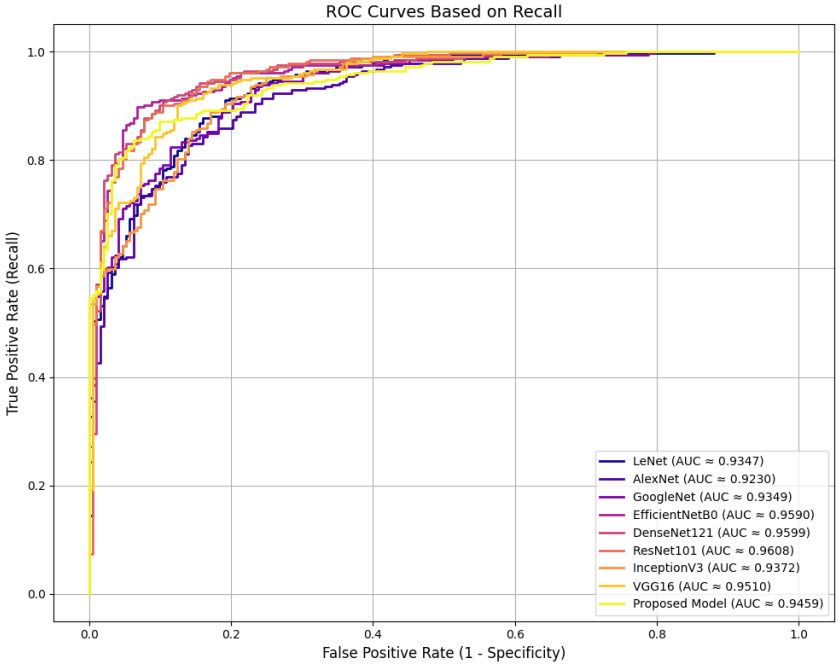

Figure 4.8: Area Under Curve for state-of-the-art models

4.3.8 Ablation Study Analysis

The table 4.4 presents a systematic evaluation of different architectural and training configurations for a binary classification model. The ablation study modifies one or more components at a time such as data augmentation (CGAN, Enhanced CGAN), dropout, self-attention, and training parameters (epochs, layers) to assess their individual impact on the model's performance.

Beginning with A1, the baseline configuration, only original data without augmentation was used with both self-attention and spatial attention enabled. This setup yielded a decent

Table 4.4: Ablation Study Results Comparing Different Model Parameters

ID	Orig.	CGAN	Enhanced	Dropout	Epochs	Layers	Self-attn	Spatial attn	Precision	Recall	Accuracy	F1	AUC
A1	✓	X	X	0.3	200	17	✓	✓	0.7991	0.9409	0.8916	0.8628	0.9116
A2	✓	✓	X	0.3	200	17	✓	✓	0.9036	0.9077	0.9142	0.9056	0.9489
A3	✓	X	✓	0.3	200	17	✓	✓	0.9061	0.9214	0.9156	0.9261	0.9213
A4	✓	X	✓	0.3	200	17	✓	X	0.901	0.9102	0.9125	0.9055	0.918
A5	✓	X	✓	0.5	200	17	X	✓	0.8852	0.915	0.908	0.8998	0.9135
A6	✓	X	✓	0.5	300	16	✓	✓	0.91	0.918	0.9165	0.914	0.9255
A7	✓	X	✓	0.3	200	16	X	X	0.875	0.901	0.896	0.884	0.9021
A8	✓	X	✓	0.3	100	20	✓	✓	0.889	0.9025	0.8995	0.8956	0.9075
A9	✓	X	✓	0.7	100	20	✓	✓	0.8761	0.8902	0.8866	0.8821	0.9001
A10	✓	X	✓	0.3	300	17	✓	✓	0.9112	0.9223	0.9187	0.9167	0.9267

recall of 0.9409, but the precision was significantly lower at 0.7991, resulting in an F1-score of 0.8628. The model was better at identifying positive cases but often misclassified negative cases as positive, indicating a relatively high false positive rate. In A2, the model incorporated CGAN-based data augmentation, while EnhancedCGAN was not used. This led to a notable improvement in both precision (0.9036) and recall (0.9077), boosting the F1-score to 0.9056. Most remarkably, the AUC reached 0.9489, the highest among all configurations, showing improved overall discrimination between the classes. However, the F1 and accuracy were still slightly lower than the best-performing configuration.

A3 introduced Enhanced CGAN while keeping CGAN disabled and retained both attention blocks. This configuration emerged as the best-performing model across most metrics, achieving the highest F1-score of 0.9261, precision of 0.9061, and a robust AUC of 0.9213. With balanced attention mechanisms and a dropout rate of 0.3 over 200 epochs and 17 layers, it demonstrated the most effective precision-recall tradeoff, indicating fewer false positives and false negatives. In A4, spatial attention was removed from the A3 configuration. This led to a slight drop in performance, with the F1-score reducing to 0.9055 and AUC to 0.918. Although still strong, the decline shows that spatial attention contributes positively to performance. A5 instead removed self-attention, while keeping spatial attention. It resulted in further reduction in precision (0.8852) and F1-score (0.8998), confirming that both attention mechanisms are important, and the model performs best when both are active. A6 increased the number of epochs to 300 and reduced the number of layers to 16, maintaining both attention types. This led to a high F1-score of 0.914 and strong precision (0.91), but it still didn't surpass A3, implying that more training doesn't always result in better generalization when the architecture is already optimal. Similarly, A7 eliminated both attention modules and resulted in lower metrics across the board, particularly an F1-score of 0.884, indicating that attention mechanisms are critical to performance. In A8, the model was trained for fewer epochs (100) and had 20 layers. It retained both attention blocks and had a dropout of 0.3. The performance remained decent (F1: 0.8956), but again, fell short of A3, showing that longer training (200 epochs) offers better convergence. A9 used a high dropout of 0.7 with the same architecture as A8. As expected, the

performance dropped further (F1: 0.8821), suggesting that excessive dropout impairs learning. Finally, A10 increased the epochs to 300 with the optimal architecture of A3. Although the results were very close to A3, there was no significant gain (F1: 0.9167), which reinforces the idea that the model in A3 strikes the best balance between complexity, regularization, and performance.

Therefore, A3 stands out as the most effective configuration. It demonstrates the highest F1-score and strong performance across all key metrics, validating the use of Enhanced CGAN augmentation combined with both self-attention and spatial attention mechanisms.

4.3.9 Summary

The comprehensive evaluation confirms that the proposed, Enhanced Conditional GAN based augmentation and self-attention mechanisms, consistently achieves robust and consistent classification across all major performance metrics. This demonstrates its strong generalization capability, robustness to class imbalance, and practical relevance for DFU classification. An extensive ablation study evaluates the impact of various configurations, such as dropout, number of layers, attention modules, and different GAN variants. To validate the design choices, an extensive ablation study was conducted, systematically modifying individual components such as dropout rates, number of layers, types of attention modules, and augmentation strategies (CGAN vs. ECGAN). The configuration with Enhanced Conditional GAN, attention blocks, 17 layers, and 200 training epochs demonstrated the best performance across key metrics including accuracy (92.57%), F1-score (93.73%), AUC (94.31%), and specificity (91.1%) outperforming baseline and state-of-the-art architectures.This architecture demonstrates strong generalization ability, consistently maintaining high accuracy, thereby ensuring dependable performance in real-world diagnostic settings. The comprehensive evaluation affirms that the proposed framework integrating Enhanced Conditional GAN (ECGAN) based data augmentation with both self-attention and spatial attention mechanisms delivers robust, reliable, and balanced classification performance across all major evaluation metrics. By generating high quality synthetic data that closely resembles real clinical samples, the ECGAN effectively addresses the challenge of class imbalance, enhancing the diversity and representativeness of the training data. The incorporation of attention mechanisms further enables the model to focus on discriminative features, improving its capacity to distinguish between normal and abnormal DFU patches. Overall, the results confirm that the proposed method not only enhances accuracy but also ensures consistent, interpretable, and efficient performance making it a strong candidate for DFU classification.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This report presents a comprehensive study on automated Diabetic Foot Ulcer (DFU) classification, addressing critical challenges such as class imbalance and limited dataset diversity. We began by highlighting the clinical importance of early DFU detection and the limitations of existing deep learning models under imbalanced data conditions. To overcome these challenges, we explored the potential of Generative Adversarial Networks (GANs), particularly Conditional GANs (cGANs), for generating synthetic medical images.

After reviewing existing approaches and their limitations, we proposed an enhanced cGAN architecture to generate high-quality synthetic images for the minority class. These synthetic images were used to augment the training data, ensuring a more balanced and diverse dataset. These augmented datasets were then used to train a custom Convolutional Neural Network (CNN) integrated with self-attention mechanisms, which enabled the model to focus on complex and clinically relevant features. The custom Convolutional Neural Network (CNN) integrated with self-attention mechanisms effectively captured complex patterns and improved classification performance.

Extensive evaluation using metrics such as FID, PSNR, and SSIM confirmed the quality of the generated images. Classification results demonstrated that our model outperformed state-of-the-art architectures such as EfficientNetB0, DenseNet121, ResNet101 and more along with significant improvements in precision, recall, and F1-score. Overall, the combination of enhanced data augmentation and attention-driven CNN modeling proved to be a robust framework for DFU classification.

5.2 Future Work

This work presents a novel deep learning-based approach that combines an enhanced conditional GAN for data augmentation and a custom CNN architecture for image classification, addressing key challenge of data imbalance. Our proposed solution was evaluated on a binary

classification task. While our current experiments focus on a binary classification problem, the framework can be adapted and extended to handle more complex scenarios such as multi-class classification and multi-label detection tasks. This flexibility positions the model as a valuable tool for a wide range of medical and non-medical image analysis applications. Thus, the methodology is generalizable and can be extended to more complex, multiclass classification problems. Looking ahead, we plan to explore several other ways for enhancement and expansion. Firstly, the generative component the enhanced cGAN will be further refined to produce higher fidelity and more diverse synthetic data, not only for diabetic foot ulcers but also for other skin lesion types and image classes. This improvement is expected to further mitigate data scarcity and imbalance issues in various domains. Secondly, we intend to fine tune and adapt the classification network for broader applications, including large scale datasets and multi-modal inputs, to enhance its clinical applicability and scalability. Moreover, integrating explainable AI techniques to visualize model decisions and attention maps will be a focus, improving the interpretability necessary for clinical acceptance.

Finally, we recognize the potential for practical deployment of this framework in real-world healthcare environments. Future work will investigate system integration, real-time inference capabilities, and user-interface design to facilitate adoption by clinicians and healthcare providers.Thus, the proposed framework has potential for integration into real world systems, and we intend to explore its deployment in practical scenarios that require automated, reliable image analysis.

5.3 Paper Publication

Tejal Khade, Dr. Chandra Prakash, "Enhanced Conditional GAN-Augmented Deep Neural Network with Self-Attention for Diabetic Foot Ulcer Classification", IEEE 4th International Conference on Technology, Engineering, Management for Societal impact using Marketing, Entrepreneurship and Talent (TEMSMET), 2025, National Institute of Technology (NIT) Delhi. Status : Under Review.

Bibliography

[1] A. Makhlouf, M. Maayah, N. Abughanam, and C. Catal, "The use of generative adversarial networks in medical image augmentation," vol. 35. Springer, 2023.

[2] M. Goyal, N. D. Reeves, S. Rajbhandari, N. Ahmad, C. Wang, and M. H. Yap, "Recognition of ischaemia and infection in diabetic foot ulcers: Dataset and techniques," *Computers in Biology and Medicine*, vol. 117, p. 103616, 2020.

[3] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman, and P. R. Pinheiro, "Covidgan: Data augmentation using auxiliary classifier gan for improved covid-19 detection," *IEEE Access*, vol. 8, pp. 91 916–91 923, 2020.

[4] M. H. Yap, B. Cassidy, J. M. Pappachan, C. O'Shea, D. Gillespie, and N. D. Reeves, "Analysis towards classification of infection and ischaemia of diabetic foot ulcers," in *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, Athens, Greece, 2021, pp. 1–4.

[5] N. Bansal and A. Vidyarthi, "Dfootnet: A domain adaptive classification framework for diabetic foot ulcers using dense neural network architecture," *Cognitive Computation*, vol. 16, no. 5, pp. 2511–2527, 2024.

[6] M. Hamghalam and A. L. Simpson, "Medical image synthesis via conditional gans: Application to segmenting brain tumours," *Computers in Biology and Medicine*, vol. 170, p. 107982, 2024.

[7] J. Amin, M. Sharif, N. Gul, S. Kadry, and C. Chakraborty, "Quantum ma-

chine learning architecture for covid-19 classification based on synthetic data generation using conditional adversarial neural network," *Cognitive Computation*, vol. 14, no. 5, pp. 1677–1688, 2022.

[8] M. S. A. Toofanee *et al.*, "Dfu-siam: A novel diabetic foot ulcer classification with deep learning," *IEEE Access*, vol. 11, pp. 98 315–98 332, 2023.

[9] A. Qayyum, A. Benzinou, M. Mazher, and F. Meriaudeau, "Efficient multi-model vision transformer based on feature fusion for classification of dfuc2021 challenge," in *Diabetic Foot Ulcers Grand Challenge*. Springer International Publishing, 2022, pp. 62–75.

[10] L. Alzubaidi, A. A. Abbood, M. A. Fadhel, O. Al-Shamma, and J. Zhang, "Comparison of hybrid convolutional neural network models for diabetic foot ulcer classification," *Journal of Engineering Science and Technology*, vol. 16, no. 3, 2021.

[11] L. Alzubaidi, M. A. Fadhel, S. R. Oleiwi, O. Al-Shamma, and J. Zhang, "Dfu_qutnet: diabetic foot ulcer classification using novel deep convolutional neural network," *Multimedia Tools and Applications*, vol. 79, no. 21, pp. 15 655–15 677, 2020.

[12] M. Goyal, N. D. Reeves, A. K. Davison, S. Rajbhandari, J. Spragg, and M. H. Yap, "Dfunet: Convolutional neural networks for diabetic foot ulcer classification," *arXiv preprint*, vol. arXiv:1711.10448v2, 2017.

[13] N. Al-Garaawi, R. Ebsim, A. F. H. Alharan, and M. H. Yap, "Diabetic foot ulcer classification using mapped binary patterns and convolutional neural networks," *Computers in Biology and Medicine*, vol. 140, p. 105055, 2022.

[14] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE Transactions on Medical*

Imaging, vol. 35, no. 5, pp. 1299–1312, May 2016, epub 2016 Mar 7.

[15] F. Veredas, H. Mesa, and L. Morente, "Binary tissue classification on wound images with neural networks and bayesian classifiers," *IEEE Transactions on Medical Imaging*, vol. 29, no. 2, pp. 410–427, Feb. 2010, epub 2009 Oct 13.

[16] C. Liu, J. J. van Netten, J. G. van Baal, S. A. Bus, and F. van der Heijden, "Automatic detection of diabetic foot complications with infrared thermography by asymmetric analysis," *Journal of Biomedical Optics*, vol. 20, no. 2, p. 026003, Feb. 2015, pMID: 25671671.

[17] L. Wang, P. C. Pedersen, D. M. Strong, B. Tulu, E. Agu, and R. Ignotz, "Smartphone-based wound assessment system for patients with diabetes," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 2, pp. 477–488, Feb. 2015, epub 2014 Sep 17.

[18] M. H. Yap, K. E. Chatwin, C. C. Ng, C. A. Abbott, F. L. Bowling, S. Rajbhandari, A. J. M. Boulton, and N. D. Reeves, "A new mobile application for standardizing diabetic foot images," *Journal of Diabetes Science and Technology*, vol. 12, no. 1, pp. 169–173, Jan. 2018, epub 2017 Jun 21, PMCID: PMC5761973.

[19] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, May 2016.

[20] M. H. Yap *et al.*, "Deep learning in diabetic foot ulcers detection: A comprehensive evaluation," *Computers in Biology and Medicine*, vol. 135, p. 104596, 2021.

Appendix A

Industrial Report

About Intel Corporation

Intel Corporation is a leading American multinational technology company renowned for its pioneering work in the semiconductor industry. Founded in 1968 by Robert Noyce and Gordon Moore, the company is headquartered in Santa Clara, California, at the heart of Silicon Valley. Intel has played a transformative role in shaping the modern computing era through continuous innovation in microprocessor and semiconductor technologies.

Intel is best known for designing and manufacturing microprocessors that serve as the central processing units (CPUs) in the majority of personal computers and servers. The company's processors, particularly the x86 architecture, have become the standard in the computing industry. Notable product lines such as the Pentium, Core i3/i5/i7/i9 series, and Xeon processors have powered everything from consumer laptops to enterprise-grade servers and cloud infrastructure.

One of Intel's most significant contributions to the industry is its advancement of Moore's Law—an observation made by co-founder Gordon Moore, stating that the number of transistors on a chip would double approximately every two years. This principle has driven exponential growth in computational power, enabling faster and more energy-efficient computing devices. Beyond CPUs, Intel has also led innovation in complementary technologies, including chipsets, integrated graphics solutions, field-programmable gate arrays (FPGAs), networking components, and high-performance memory technologies such as 3D XPoint and solid-state drives (SSDs). These technologies have not only improved the performance and reliability of personal computing devices but have also empowered modern data centers, artificial intelligence applications, and high-performance computing systems.

Today, Intel continues to invest in cutting-edge research and development across emerging fields such as quantum computing, AI accelerators, autonomous driving (via Mobileye), and advanced process nodes (e.g., Intel 7, Intel 4). With its global reach and technological leadership, Intel remains a key player in driving digital transformation and shaping the future of computing.

A.1 Internship at Intel, Bangalore

During my internship, I dedicated significant effort to rapidly ramping up my knowledge and gaining practical, hands-on experience with advanced AI technologies, with a special focus on the Intel AI ecosystem. I immersed myself in understanding and working with inference workloads, leveraging cutting-edge tools and frameworks including OpenVINO™, Intel® Neural Processing Unit (NPU), and ONNX Runtime.

A core part of my project involved working extensively with the OpenVINO Execution Provider integrated within the ONNX Runtime deep learning framework. This integration allowed me to explore the performance benefits of Intel's heterogeneous computing architecture, leveraging CPUs, integrated GPUs, and VPUs to accelerate AI inference. I focused on optimizing deep learning models to run efficiently using OpenVINO's graph transformations, quantization techniques (such as INT8 and INT4 QDQ formats), and hardware-specific execution paths. My work involved enabling support for large language models (LLMs) and computer vision models by ensuring operator compatibility and performance tuning across various Intel hardware platforms. By profiling inference pipelines, identifying bottlenecks, and utilizing OpenVINO's runtime configuration and caching mechanisms , I contributed to significantly reducing inference latency while maintaining high accuracy. This experience deepened my understanding of how low-level optimizations can enhance real-world AI deployment, especially in edge and enterprise settings.

Additionally, I actively participated in open-source development initiatives, where my enhancements helped achieve measurable performance gains across a variety of deep learning models. These improvements directly contributed to enabling more efficient deployment and scalability of AI applications on Intel's diverse hardware portfolio.

Overall, this internship provided me with invaluable exposure to Intel's state-of-the-art AI technologies and deepened my understanding of how hardware and software co-design can accelerate real-world AI workloads, ultimately equipping me with practical skills to drive innovation in AI deployment.

A.2 Motivation

The primary purpose of my internship at Intel Bangalore was to actively contribute to innovative projects that push the boundaries of artificial intelligence capabilities, while learning from and collaborating with some of the most talented experts in the industry. This internship served as an invaluable opportunity to deepen my technical knowledge, sharpen my practical skills, and make meaningful contributions toward Intel's mission of advancing AI technologies that will fundamentally shape the future of computing and society at large.

Moreover, the internship was designed to provide a comprehensive learning and development experience, enabling me to explore diverse career paths within the AI and technology domain, and to gain hands-on exposure to cutting-edge industry practices.Beyond technical tasks, the internship was structured to offer a holistic development experience. I explored various domains within the AI ecosystem—such as model quantization, inference acceleration, and system-level optimization—which gave me a broader understanding of how AI is built, tested, and delivered at scale. I actively participated in team discussions, sprint planning, and review. This experience has significantly enriched my understanding of AI's real-world applications and prepared me for a successful professional journey in the rapidly evolving tech landscape.

A.3 Layout

The structure of this report is as follows: Section 2 details the tasks assigned during the internship along with their descriptions. Section 3 outlines the tools and technologies learned and utilized throughout the internship. Finally, Section 4 presents the conclusion summarizing the overall experience and key takeaways.

Appendix B

Project Details

During my internship at Intel Bangalore, I worked in multiple domains related to AI, contributing to both internal initiatives and open source releases. My primary focus areas included Generative AI model optimization, ONNX Runtime enhancements, validation and testing of the OpenVINO Execution Provider (OVEP), and feature test case development.

B.1 Generative AI Workloads - Quantization and Compression of GenAI Models

During my internship, I concentrated on optimizing generative AI models, specifically LLaMA and Phi-3, by applying advanced quantization and compression techniques to improve their inference performance using Intel's OpenVINO toolkit. The primary objective was to transform large language models (LLMs) into more efficient formats that can be effectively executed on Intel hardware. This involved balancing the trade-off between maintaining model accuracy and significantly reducing computational requirements and inference latency, thereby enabling faster and more resource-efficient deployment of these state-of-the-art models. Key Tasks:

- **Inference Workflow Exploration:** Investigated the process of running LLaMA and Phi-3 models using OpenVINO Intermediate Representation (IR). This involved converting models from PyTorch to OpenVINO format utilizing both the Model Converter CLI and Python APIs, enabling optimized deployment on Intel hardware.
- **Quantization for Performance:** Applied advanced quantization techniques including NNCF-based post-training quantization and AWQ (Activation-aware Weight Quantization) to compress large language models into efficient INT8 and INT4 numerical formats. AWQ, which selectively adjusts weight quantization by considering activation distributions, allowed improved accuracy retention during aggressive compression. Developed a 4-bit QDQ (Quantize-Dequantize) version of the Phi-3 model using ONNX Runtime quantization tools, significantly reducing the model's memory footprint and computa-

tional load. This compression pipeline resulted in faster inference while maintaining a minimal drop in model accuracy.

- **Operator Optimization:** Analyzed critical inference operators such as DequantizeLinear and MatMul to identify bottlenecks and opportunities for backend improvements. Implemented operator fusion optimizations, replacing multiple operators with a single FusedMatMul operator to reduce data movement and latency, thereby accelerating inference throughput.

- **Benchmarking and Performance Metrics:** Conducted extensive benchmarking of INT4 QDQ models across various hardware backends to evaluate performance and compatibility. The 4-bit quantized Phi-3 model demonstrated a substantial reduction in inference time—measured in tokens generated per second—resulting in faster real-time text generation. These benchmarks confirmed that activation-aware weight quantization combined with QDQ techniques can dramatically improve throughput on Intel hardware while preserving accuracy and stability.

- **Stable Diffusion Model Inference:** Implemented inference workflows for Stable Diffusion models using the OpenVINO Toolkit and ONNX Runtime with the OpenVINO Execution Provider on Intel's Meteor Lake GPU running Windows 11. Converted Hugging Face Stable Diffusion models to ONNX format using the Optimum CLI for seamless compatibility. Successfully executed the stable_diffusion_ort_inference script with multiple Hugging Face models and diverse prompts, generating high-quality images. Performance metrics such as inference time and GPU utilization were measured, providing insights into the acceleration benefits offered by OpenVINO Execution Provider on Intel GPUs for generative AI tasks.

- **Pre-training and Post-training Quantization of Generative AI Models**

  Quantization is a crucial technique to reduce the computational complexity and memory footprint of large generative AI models, enabling faster inference and deployment on resource-constrained hardware.

  1. Pre-training Quantization: This approach involves integrating quantization techniques during the model training process itself. The model learns to operate under reduced-precision constraints (such as INT8 or INT4) from the start or after a few initial full-precision training epochs. Techniques like Quantization-Aware Training (QAT) help the model adapt to quantized weights and activations, often leading to minimal accuracy degradation. Pre-training quantization requires access to the training data and additional compute resources but yields models that are inherently robust to low-precision inference.

2. Post-training Quantization: In contrast, post-training quantization (PTQ) is applied after the model has been fully trained in full precision (typically FP32). It compresses the trained weights and optionally activations to lower bit-width representations without retraining or with minimal fine-tuning. PTQ techniques, such as symmetric/asymmetric quantization or advanced methods like Activation-aware Weight Quantization (AWQ), allow rapid compression of large language models to INT8 or INT4 formats, significantly reducing model size and latency. While PTQ is faster and more practical when training data or compute are limited, it may sometimes cause slight accuracy loss compared to QAT.

For generative AI models like LLaMA and Phi-3, both pre-training and post-training quantization techniques are instrumental. Pre-training quantization produces models better adapted for low-precision environments, while post-training quantization enables quick deployment and optimization on Intel hardware using toolkits like OpenVINO and ONNX Runtime. This project greatly deepened my understanding of deploying efficient large language models using Intel's AI toolchain and reinforced the critical role of quantization and operator fusion in accelerating inference workloads.

B.2 Bug Fixing and Validation

B.2.1 NuGet Package Generation and Validation

Automated the generation and validation of NuGet packages for Windows environments. This task involved gaining a foundational understanding of the NuGet ecosystem, setting up workflows for package generation, and developing scripts to automate the build and validation processes. The objective was to ensure consistent and seamless creation of deployable packages. I implemented automation scripts to streamline the NuGet packaging process on Windows, gaining insights into package structure, installation, and deployment mechanisms.

B.2.2 Fixing and Updating OpenVINO Execution Provider (OVEP) Samples and ORT Build Instructions

Addressed build issues and improved sample functionality for OpenVINO Execution Provider integration in ONNX Runtime. This involved debugging and fixing broken OVEP sample code, updating deprecated methods, and refining build documentation to ensure smooth builds across different environments. These contributions helped improve the developer experience and usability of ONNX Runtime with OVEP.

B.2.3 Dashboard Generation using PowerBI for Validation Infrastructure

Created a validation dashboard using Power BI to visualize model execution metrics. Focused on analyzing ONNX Runtime validation data sheets, performing data cleaning and transformation, and building visual dashboards. Key visuals included latency trends, model execution counts across various Device Execution Providers (Device_EPs), and summary insights that enhanced transparency and decision-making in validation pipelines.

B.2.4 OVEP Validation Infrastructure Contribution

I independently managed and executed the entire OpenVINO Execution Provider (OVEP) Validation Infrastructure, contributing significantly to its development and automation. Gained hands-on experience with Jenkins pipeline setup for performance, FEIL-FIL, and unit testing, including complete configuration for devices, Execution Providers (EPs), models, and machine environments. Designed and maintained automated pipelines for OVEP feature testing, integrated with custom Python scripts for streamlined test execution. Developed and executed feature test cases for new capabilities in ONNX Runtime with OpenVINO Execution Provider. Covered multiple edge cases and end-to-end (E2E) scenarios to ensure functional correctness, stability, and compatibility across configurations.

B.2.5 Automated Testing with Pytest Framework

As part of the validation infrastructure enhancements, I extensively utilized the pytest testing framework to design, implement, and maintain automated test suites for ONNX Runtime's OpenVINO Execution Provider (OVEP). Pytest's modular and flexible architecture allowed me to create comprehensive and reusable test cases that covered a wide spectrum of testing needs—from unit-level validation of individual operators and configuration parameters to integration and end-to-end functional tests simulating full model inference pipelines.

I designed parameterized tests to systematically evaluate multiple configuration combinations, such as execution providers, precision modes (e.g., FP32, INT8, INT4), and hardware targets (CPU, GPU, NPU). These tests ensured correctness and consistency across multiple environments and helped detect performance regressions, fallback behaviors, or missing operator support during continuous integration (CI) runs.

I also implemented assertions for verifying OpenVINO-specific outputs such as .blob cache generation, EP fallback logic, and execution provider selection based on runtime conditions. Logging and coverage tools were integrated to track test stability and measure validation depth over time. This effort not only strengthened the reliability and robustness of OVEP within

the ONNX Runtime ecosystem but also contributed to better developer confidence and faster iteration cycles during feature development and bug fixes.

Key activities included:

- Designing test cases to verify new features and bug fixes in the OVEP, ensuring code correctness and robustness.
- Developing parameterized tests with pytest to efficiently run the same tests across multiple configurations, such as different models, devices, and execution providers.
- Integrating pytest-based tests into Jenkins CI/CD pipelines for automated and continuous validation, enabling rapid feedback on code changes.
- Leveraging pytest fixtures to manage setup and teardown processes, such as initializing test environments, loading test data, and cleaning up resources post-execution.
- Utilizing pytest's rich reporting and logging capabilities to improve test visibility, helping the team quickly identify and troubleshoot failures.
- Writing custom pytest plugins and hooks when necessary to extend testing functionality and tailor it to specific project requirements.

This rigorous use of the pytest framework significantly improved the reliability and maintainability of the OVEP validation infrastructure. It streamlined regression testing workflows, reduced manual testing effort, and ensured higher code quality throughout the development lifecycle.

Appendix C

Technologies

C.1 Technologies Used

Machine Learning / Deep Learning / AI Frameworks:

- **ONNX (Open Neural Network Exchange):** ONNX provides a unified, open standard for representing deep learning models across a variety of frameworks such as PyTorch, TensorFlow, and more. I leveraged ONNX extensively to validate model integrity, analyze model graphs at the operator level, and ensure seamless interoperability between different AI frameworks and hardware backends. This allowed me to perform fine-grained compatibility checks and optimize inference workflows by understanding how models translate across platforms.

- **OpenVINO (Open Visual Inference and Neural Network Optimization):** Intel's OpenVINO toolkit was central to my internship work, serving as the primary platform for optimizing and deploying AI models on Intel hardware, including CPUs, integrated GPUs (iGPU), discrete GPUs (dGPU), and Neural Processing Units (NPU). I gained hands-on experience with model conversion pipelines—from frameworks like PyTorch to OpenVINO Intermediate Representation (IR)—and applied quantization techniques (FP16, INT8, and INT4) to compress models for improved runtime efficiency. Additionally, I contributed to improving the OpenVINO Execution Provider (OVEP) integrated with ONNX Runtime, enhancing performance, backend compatibility, and developer usability.

- **ONNX Runtime (ORT):** ORT is a high-performance, cross-platform inference engine designed to execute ONNX models efficiently across various hardware accelerators. I utilized ORT extensively to run complex AI models, such as YOLOv3 for real-time object detection and Stable Diffusion for generative AI. I explored multiple execution providers, including CPU MLAS (Machine Learning Acceleration Subsystem), OpenVINO Execution Provider, and specialized NPUs, to benchmark performance, debug integration issues, and develop robust test cases that verify feature functionality and stability.

- **Hardware Backends: CPU, GPU (iGPU, dGPU), NPU** Throughout the internship, I worked closely with diverse hardware backends to optimize model inference performance. CPUs provided a general-purpose baseline, while GPUs (both integrated and discrete) enabled parallel acceleration of deep learning tasks. NPUs, specialized accelerators for AI workloads, offered substantial power efficiency and speed improvements. My work with Intel's NPUs involved validating the inference of quantized models (e.g., INT4 QDQ Phi-3) and analyzing backend compatibility, achieving significant gains in throughput and latency over traditional CPU/GPU execution.

- **Package Management and Build Tools**

  - **NuGet:** Automated the generation, packaging, and validation of NuGet packages for Windows environments. This included scripting workflows for continuous integration and ensuring the integrity and reliability of packages for distribution within the development ecosystem.
  - **Python:** Utilized extensively for automation scripting, model conversion workflows, quantization pipelines, and writing test cases. Python's rich ecosystem enabled rapid prototyping and integration with tools like ONNX Runtime and OpenVINO.
  - **C++:** Developed standalone and integrated C++ applications for feature testing, unit validation, and performance measurement of ONNX Runtime and OpenVINO Execution Provider components. C++ proficiency was essential for low-level debugging and optimizing critical code paths.
  - **PyTorch:** Worked with PyTorch-based models as the primary development framework before converting models into ONNX and OpenVINO IR formats. Applied quantization and pruning techniques within PyTorch to prepare models for efficient deployment.

- **Data Visualization and Reporting**

  - **Power BI:** Designed and developed dynamic, interactive dashboards in Power BI to visualize a comprehensive range of model validation metrics, including accuracy, precision, recall, and F1-Score across multiple benchmarks. Integrated data from various sources such as performance logs, CSV reports, and SQL databases. These dashboards provided actionable insights into model behavior, enabled traceability across experiments, and supported data-driven decisions within the ML validation and QA teams.

- **Jenkins:** Implemented and maintained robust CI/CD pipelines using Jenkins for end-to-end automation of the machine learning lifecycle. This included model compilation, unit testing, integration testing, deployment to staging and production environments, and automated performance benchmarking. Customized Jenkinsfile scripts to incorporate conditional build triggers and parallel job execution, significantly reducing pipeline execution time and human intervention.
- **Docker:** Employed Docker to containerize machine learning applications, ensuring portability and reproducibility across development, testing, and deployment stages. Created custom Dockerfiles to encapsulate dependencies such as ONNXRuntime, OpenVINO, and required Python packages. Leveraged multi-stage builds and volume mounting to optimize container size and runtime efficiency. Facilitated seamless collaboration across teams by standardizing the development environment and minimizing environment-specific bugs.

Appendix D

Summary

During my one-year internship at Intel Bangalore, I had the opportunity to work across multiple impactful domains, encompassing development, inference, and optimization of generative AI models. My core focus was on building end-to-end GENAI pipelines, developing robust APIs, and integrating large language models (LLMs) with advanced quantization techniques to enhance performance and efficiency.

I worked extensively with cutting-edge technologies such as OpenVINO and ONNX Runtime to automate data workflows, construct scalable machine learning pipelines, and deliver high-performance inference solutions for a wide range of deep learning and traditional ML models. My role involved optimizing model execution across diverse Intel hardware platforms (CPU, iGPU, VPU/NPU), ensuring compatibility and performance tuning using OpenVINO Execution Provider (OVEP) integrated within ONNX Runtime. I also contributed to the development of quantized inference pipelines (e.g., INT8/INT4) and explored model export formats for efficient deployment.

Throughout this process, I honed my skills in Python and C++, writing robust, modular code for both prototyping and production workflows. I gained hands-on experience in continuous integration and deployment (CI/CD) environments using tools such as Jenkins and Docker, where I automated test suites, built reproducible containerized environments, and monitored performance benchmarks across builds. I regularly engaged in debugging model execution graphs, optimizing runtime memory and latency, and analyzing backend behavior through profiling tools to identify bottlenecks.

Additionally, I contributed to creating interactive validation dashboards that enabled insightful monitoring and analysis of model performance and accuracy. This comprehensive hands-on experience not only strengthened my technical expertise but also enhanced my problem-solving capabilities, providing a solid foundation for a successful career in the technology industry.