Multi-view Face Detection Using Hybrid Generative Adversarial Networks and Multi-Task Cascaded Convolutional Neural Network

Kibru Alemu Terfasa

A Thesis Submitted to the Department of Computer Science and Engineering.

College of Electrical Engineering and Computing.

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in computer Science and Engineering.

Office of Graduate Studies

Adama Science and Technology University

June, 2025

Adama, Ethiopia

Multi-view Face Detection Using Hybrid Generative Adversarial Networks and Multi-Task Cascaded Convolutional Neural Network

Kibru Alemu Terfasa

Advisor: Feyisa Woyano (Ph.D)

A Thesis Submitted to the Department of Computer Science and Engineering.

College of Electrical Engineering and Computing.

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in computer Science and Engineering.

Office of Graduate Studies
Adama Science and Technology University

June, 2025

Adama, Ethiopia

DECLARATION

This Thesis entitled "**Multi-view Face Detection Using Hybrid Generative Adversarial Networks and Multi-task cascaded convolutional Neural Network"** is my work and has not been submitted to any University for a similar purpose. Proper citations duly recognize the references used in this thesis.

Kibru Alemu Terfasa		
Name of student.	Signature	Date

RECOMMENDATION OF ADVISOR

I, the Major Advisor of this thesis , hereby certify that I have closely advised the student while developing this thesis and read the draft thesis entitled" **Multi-view Face detection Using Hybrid Generative Adversarial Networks and Multi-task cascaded convolutional Neural Network"** prepared under my guidance by **Kibru Alemu Terfasa**. Therefore, I recommend the submission of the thesis to the department for further review and evaluation.

Feyisa Woyano(Ph.D.)		
Major Advisor	Signature	Date

APPROVAL SHEET

I/We hereby certify that the recommendation and suggestion given by the thesis review committee are appropriately incorporated into the final thesis entitled" **Multi-view Face Detection Using Hybrid Generative Adversarial Networks and Multi-Task Cascaded Convolutional Neural Network"** by **Kibru Alemu Terfasa**.

Major Advisor	Signature	Date
Co-Advisor	Signature	Date

APPROVAL OF BOARD OF REVIEWERS

We, the advisors of the thesis, members of the Board of Reviewers of the thesis open defense by **Kibru Alemu Terfasa** have read and evaluated the thesis entitled " **Multi-view Face Detection Using Hybrid Generative Adversarial Networks and Multi-Task Cascaded Convolutional Neural Network"** and assessed the understanding of the candidate about the proposed research. This is , therefore, to certify that the thesis is accepted, and we recommend the implementation of the thesis.

Chairperson	Signature	Date
Reviewer 1	Signature	Date
Reviewer 2	Signature	Date

Final approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and College Graduate Committee (CGC).

Department	Signature	Date
College Dean	Signature	Date
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGMENT

Above all, I would like to express my deepest gratitude to Almighty God for His boundless mercy and unwavering support during my entire research work, from inception to successful completion. This undertaking would not have been feasible without His blessings.

I am truly grateful to my advisor, **Dr. Feyisa Woyano(PH.D.)** for his outstanding advice, mentoring, and support during my research, His perceptive criticism and well-considered recommendations have greatly influenced the focus and quality of my research work.

I would like to express my deepest gratitude to all members of the Data Science, AI, Computer Vision, and Computer Networks SIG for providing a supportive academic atmosphere and for their precious support during my postgraduate course and research period. Their dedicated commitment and persistence have contributed extensively to advancing my studies and facilitating the seamless flow of my research.

Lastly, my families and friends deserve special recognition for their support and daily prayers during this endeavor.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

API	The Application Programming Interface
BCE	Binary Cross-Entropy
CAD	Couple-Agent Discriminator
CAPG-GAN	Couple Agent Pose-Guided Generative Adversarial Network
CPU	Central Processing Unit
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
FD	Frontal detected
FP	False Positive Rate
GAN	Generative Adversarial Network
GF	Generated frontal
GD	Generated Detected
GT	Ground Truth face
IDE	Integrated Development Environment
GPU	Graphics Processing Unit
IOU	Intersection Over Union
MTCNN	Multi-Task Cascaded Neural Network
O-Net	Output Network
OS	Operating System
PG	Pose guided Generator
P-Net	Proposal Network
RAM	Random Access Memory
ReLU	Rectified Linear
R-Net	Refine Network
Tanh	Hyperbolic Tangent Activation Function
TP-GAN	Two Path Generative Adversarial Network
VGG19	Visual Geometry Group 19-layer model

LIST OF NOTATION AND DESCRIPTION

Notation	Description
A1	Agent 1 Discriminator
A2	Agent 2 Discriminator
Dx	Discriminator output
Gx	Generator output
θD	Parameter of Discriminator
θG	Parameter of Generator
Θ	Model parameter
α	LeakyReLU negative slope
I	Input image (non_frontal image)
K()	Key(A linear transformation of the input)
h()	A function computing attention scores,
g()	A function applied after weighted sum of values
f()	Final attention output
E	Expectation operator
LG	An actual image fetched from the dataset.
P1	Pose embedding of input(current) image
P2	Pose embedding of input(target) image
Q()	Query(positions in the input.)
p(z)	A latent vector (noise) drawn from a simple distribution
G(z)	Pre-specification of the latent
V()	Vector(A linear transformation of the input)
-	Signifies loss

ABSTRACT

Multi-view face detection is the fundamental technology of an open human-computer interaction environment. Despite satisfactory development in frontal face detection, non-frontal extreme poses are still persistent challenge. In real-world scenarios, non-frontal views are common due to camera positioning, subject movement, or occlusion. As the head rotates, essential facial features such as the eyes, nose, and mouth become partially occluded, significantly degrading detection performance. To address the issues, we employed a two-stage solution with the combination of face frontalization and detection, Initially, Couple agent pose guided Generative Adversarial Network *employed to generate a good quality frontal face images from their profile counterparts, thereby enhancing the effectiveness of conventional detection methods. This followed by mitigate application of Multitask cascaded convolutional neural networks for detection. By integrating CAPG-GAN for frontalization and MTCNN for detection, our hybrid solution framework improved the ness and face detection across diverse poses and occlusion conditions. The performance of the suggested hybrid CAPG GAN-MTCNN model was evaluated on two benchmark datasets: Multi-PIE and CAS-PEAL-R1. The proposed model attained high structural similarity in generated frontal images with SSIM measures of 94.5% and 91.45% for Multi-PIE and CAS-PEAL-R1, respectively. Additionally, Human perceptual experiments also confirmed the visual quality and identity consistency of frontalized faces with near-perfect ratings even for extreme profiles. Based on our standard face detection metrics, our proposed CAPG- GAN-MTCNN model significantly improves multi-view face detection performance under extreme pose variations and occlusions. On the Multi-PIE dataset, our proposed approach achieved an average IOU of 0.792 to 0.949 and Recall of 0.885 to 0.976, surpassing the baseline MTCNN advanced and also on the CAS-PEAL-R1 dataset our proposed model achieved an average IOU of 0.756 to 0.935 and Recall of 0.86 to 0.95, surpassing the baseline MTCNN++ . The PR curves are also proof of the performance of the proposed mode, on the Multi-PIE dataset, the model accomplished an AUC of 0.985, whereas 0.965 on the CAS-PEAL-R1 dataset. These all results confirm that the CAPG-GAN-MTCNN framework offers a substantial advantage in detecting faces under extreme up to 90˚ pose challenging, making it a strong candidate for real-world face detection applications.*

Keywords: Couple Agent, Generative Adversarial Network, Invariant Pose, Multi-Task Cascaded Convolutional Neural Network, Multiview-Face Detection, Pose Guide

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

Face detection is a fundamental technology required for achieving natural human-computer interaction (C. Zhang and Zhang, 2014). The challenge of detecting faces from multiple viewpoints in an open environment poses severe problems arising from the large variations in facial appearances and shapes (Wu et al., 2017). While frontal face detection has been widely considered a solved problem, multi-view face detection remains challenging due to dramatic appearance variations over different poses. Face detection in images is a challenging task in research, mainly due to the variations brought about by human faces in terms of occlusion- when the face is partly covered by objects like long hair or sunglasses and variations in pose frontal and profile views and non-frontal views (Mohammadian et al., 2020)(C. Zhang and Zhang, 2014). Traditional face detection systems face challenges when dealing with these challenging conditions. A variety of methods has been applied to address this issue; however, multi-view face detection remains highly challenging due to the significantly more complicated variations of orientation. Current detection systems work well for the identification of frontal faces but decline in performance as more challenging scenarios are presented to them, such as the detection of faces at various angles (Wu et al., 2017).

To address the issues of multi-view face detection, we take a two-stage approach combining face frontalization and detection. We use CAPG-GAN first to create high-quality frontal face images from non-frontal inputs to improve the performance of face detection with pose variations. While CAPG-GAN is excellent in generating realistic frontal faces, extreme poses, low resolution, occlusion and lighting variations can lead to artifacts such as texture inconsistency and missing details. To overcome these issues, we adopt another approach frontal face detection using MTCNN. The MTCNN framework is used to detect the generated frontal images of CAPG-GAN, thus ensuring that only high quality and well-aligned facial representations are sent for further analysis.

In this work, we adopt an image-based pose handling approach, where faces are synthesized to similar poses for alignment, enabling pose-invariant face detection. Generative Adversarial Network (GAN) based techniques can generate and synthesize frontal images from non-frontal images of realistic faces that cause profound social concerns and security problems. The development of Generative Adversarial Networks (GANs)(Goodfellow et al., 2014) led to a dramatic increase in the realism in generating high-quality face images, including PGGAN(Karras, 2017), Style-GAN (Karras et al., 2019), StyleGAN2(Karras et al., 2020), and StyleGAN3(Karras et al., 2021) and TP-GAN. These GAN-generated (or synthesized) fake faces are hard to distinguish by human eyes. Such synthesized faces are easily generatable and can be directly leveraged for disinformation, which might lead to profound social, security, and ethical concerns. GANs are used to generate synthetic images that span a large area of variations in extreme pose and different orientations. The newly generated samples supplement the original training data, filling those areas where real-world examples are few or predominantly frontal.

The three stages of MTCNN are capable of face detection, bounding box refinement, and face landmark locations; those are (two corner Mouth, nose, and eyes detection(Khan et al., 2024). The combination of CAPG-GAN for frontalization and MTCNN for detection forms our hybrid approach, which greatly improves the reliability of multi-view face detection, thus enhancing performance and ness in varying poses and challenging real-world environments. Despite significant progress in recent years, the task of face detection in unconstrained environments still presents a multitude of challenges, particularly when dealing with diverse facial orientations, varying illumination, self-occlusions, and low-resolution inputs. Accurate and reliable face detection plays a crucial role in various real-world applications, including surveillance systems, biometric authentication, driver assistance systems, and social media filters. The increasing demand for automated systems that can operate effectively under these conditions emphasizes the need for more advanced and adaptive detection pipelines that can handle the complex variability present in natural images. Recent advances in deep learning, particularly the introduction of convolutional neural networks (CNNs) and generative adversarial networks (GANs), have significantly boosted the performance of face detection and synthesis tasks. Hybrid approaches that integrate image generation with detection algorithms have demonstrated superior performance over conventional methods by improving the quality of training data and enhancing the model's ability to generalize across different scenarios. Such strategies enable the creation of pose-invariant representations, which are essential for addressing the intrinsic diversity of face appearances under real-world conditions. The proposed method takes advantage of the merits of CAPG-GAN and MTCNN to address the fundamental problems associated with multi-view face detection. The CAPG-GAN model strengthens the dataset by generating frontal

face images of high quality, thus increasing detection ness and reliability against pose variations. Meanwhile, the MTCNN framework provides a and effective solution for detection enrichment and precise localization of facial landmarks, which is important for further facial analysis-related applications. This joint solution not only solves the issue of orientation variation but also lays a good foundation for future research on human-centered computer vision systems.

1.2. Motivation of the Study

Real-time face detection is a basic requirement for many computer vision systems nowadays, including social networks, augmented reality, security systems, and surveillance systems. Multi-view face detection remains a challenging issue due to the enormous amount of variations in pose that are found in real environments, although frontal face detection has been largely mastered. When faced with faces in other than frontal views, most of the conventional face detection techniques tend to fail which degrades performance particularly in scenarios that require differentiation from multiple perspectives.

The motivation for this work stems from the urgent requirement to construct strong multi-view face recognition systems. Detection systems capable of addressing the issues related to non-frontal orientations. Traditional face detection algorithms become vulnerable in the presence of non-frontal faces perspectives, resulting in less effectiveness in applications needing robust identification from Various perspectives (Zafeiriou et al., 2015).

Multi-task Convolutional Neural Networks (MTCNNs) have demonstrated enormous potential in improving detection accuracy by learning associated tasks like face detection and landmark Localization, on the other hand, may be restricted in usefulness when confronted with extreme postures. and false positive cases (Khan et al., 2024). Generative Adversarial Networks (GANs) offer an attractive solution in that they generate realistic frontal synthetic data capable of emulating facial representations at all possible orientations (Goodfellow et al., 2014). MTCNNs can be integrated with GANs to enhance ness by frontalizing faces with pose invariance, making detection easier for challenging-to-detect faces. This integration improves the model's ability across varying poses ultimately boosting face detection performance in real-world scenarios.

1.3. Statement of the Problem

Multi-view face detection is a very important task in computer vision, particularly for applications like surveillance, biometric authentication, and human-computer interaction. Nevertheless, the detection of faces taken at different pose angles is still a great challenge. In real-world scenarios, the majority of facial images are taken at non-frontal views because of camera positioning. As the head turns extremely, prominent facial features such as the eyes, nose, and mouth be occluded to some extent partially, resulting in a substantial drop in detection. This limitation affects not only the face detection system's trustworthiness but also later tasks such as recognition and verification.

Several Previous studies have explored different methods to over comes this challenges (Santemiz et al.2024)(Zafeiriou et al.2015)(C.Zhang and Zhang, 2014)(Prasad et al.2021)(Minaee et al.2021)(Bhangale et al.2022) however multi-view face detection continues to face challenges due to extreme pose angles(with extereme pose angle above 75˚), leading to reduced detection accuracy and increase false positive, especially in case of partial face detection. Despite significant advances towards ness enhancement, existing models frequently suffer from failure to detect faces consistently across views and cases of partial occlusion. (Khan et al.,2024) also noted that the existing fails to generalize well the abundance of false positives at extreme pose angle or partial face detection, in which side-view facial patches are incorrectly labeled as full faces. Demonstrate vulnerability to false positives in situations of large pose variation and partial face presence, thereby affecting the overall validity of the system.

To bridge this gap and improve the system’s accuracy and resilience in a range of real-world scenarios, we proposed Multi-view face detection using hybrid Generative adversarial networks and Multi-task cascaded convolutional networks.

1.4. Research Questions

This study attempted to find the answers to the following questions:

RQ1.To what extent can our proposed model achieve better performance than baseline?

RQ2. To what extent can hybrid GAN-MTCNN model detect extreme pose angles in Multiview image?

R Q 3. How does the proposed hybrid approach compare to existing models regarding image detection accuracy using IOU and Recall evaluation metrics?

1.5. Objective of the Study

1.5.1. General Objective

The general objective of this study is to develop Multi-view face detection using hybrid generative adversarial networks and multi-task cascaded convolutional neural networks.

1.5.2. Specific Objectives

The specific objectives Outlined below addressed to accomplish the general objective.

- ◊ To utilize publicly accessible a multi-view face image dataset include a variety in poses.
- ◊ To pre-process available data to make it suitable for our model.
- ◊ To design a GAN-based architecture by integrating attention mechanisms and MTCNN.
- ◊ To train the proposed hybrid GAN with MTCNN model on the prepared dataset.
- ◊ To evaluate model performance using metrics like IOU, SSIM, and human perception to compare with baseline and previous research models on Multi-view face detection.

1.6. Significance of the Study

This study have several benefits in the fields listed below.

- ◊ Security and Surveillance: The implementation of Multi-view face detection significantly enhances the accuracy of the face recognition system. This is especially the case from several perspectives in the challenging environment, like posed and occlusion scenarios. This advancement will improve security, protocols, and surveillance technology.
- ◊ Human-Computer Interaction: Multi-view face detection facilitates more effective communication between user and computer making it easy to communicate with around and with computers, and this in turn makes it possible to track and identify the faces even in such situations when the users are either walking or turning. Which leads to a more interesting and personalized orientation of their users.
- ◊ Autonomous cars: It will be crucial for fully autonomous cars to accurately detect and track pedestrians and drivers across multiple angles. It contributes to enhancing security and helps the automatic cars' inefficient decision-making in complex traffic conditions most of the time.

1.7. Scopes and Limitation of the Study

1.7.1. Scopes of the Study

This research aims to detect multi-view face images from invariant poses, specifically up to 90° pose degree. It aims to detect faces from multiple angles, ensuring performance in detecting faces in various orientations. The research focuses on the publically available Multi-PIE dataset and CASPEAL-R1; specifically targeting extreme face pose orientation and frontal views to address challenges in multi-view face detection. The generated outputs are limited to moderate resolutions (e.g., 128x128 pixels) due to computational constraints, as handling higher resolutions requires additional resources and model adjustments outside the study's current capacity. This will be achieved through the implementation of combining hybrid generative adversarial networks and multi-task cascaded convolutional networks, designed to improve Multiview face detection tasks. The proposed model has been conditioned by inputting multi-view face images, followed by the evaluation of its results in comparison to previous Multiview face detection approaches.

1.7.2. Limitation of the Study

Owing to limited time and resources, the following were not accounted for in this research:

◊ This study only accounts for Multiview face detection and not Multiview face recognition tasks.

◊ Our study only conducted by considering pose angles up to 90° into account for detection. Pose angles greater than 90° (e.g., far profile or occluded views) were not considered in the evaluation.

1.8. Contribution of the Study

◊ We integrated MTCNN with the CAPG-GAN component having an encoder-decoder network architecture which involves introducing residual connections within the pyramid module and multi-view face detection.

◊ We incorporated self-attention mechanisms of on a pose-guided generator encoder block to improve quality of generated and detect frontal face.

◊ We used VGG19-based perceptual loss to enhance the structural quality and realism of frontalized faces, respectively, in order to enhance the ness and performance of self-occlusion face detection systems.

◊ We experimented on Multi-PIE and CAS-PEAL R1 databases in which we efficiently detected frontal-generated faces having a pose angle up to 90° for both the databases compare with state of art.

1.9. Organization of the Research

This research is divided into seven distinct chapters, which are planned as below:

Chapter One – Introduction: The introductory chapter is an overview of the study that is conducted. It includes the background, problem statement, research questions, objectives, scope, and limitations specific to the research.

Chapter Two – Review of the Literature and Related Studies: The second chapter presents pertinent Literature and related works on Multiview face detection. Augmentation techniques and theoretical Framework of deep learning models are discussed.

Chapter Three – Research Methodology: The research methodology employed herein is explained. It includes data used in this project, data preparation, design aids, structure and evaluation of methods used in the study.

Chapter Four – Proposed Solution: A complete explanation of the proposed solution is given in this chapter. Hybrid CAPG GAN with MTCNN. It contains the proposed solution process flow, proposed model architecture, and model training pseudocode.

Chapter Five – Implementation: This chapter outlines the experimental design and its implementation. It includes the configuration of the environment required for the implementation, code for proposed solution, and the experimental classes.

Chapter Six – Results and Discussion: Presents the findings and results derived from the experiments conducted to address the research question as well as realize the research objectives.

Chapter Seven – Conclusion and Future Directions: Chapter Seven is a summary of a synthesis of research carried out and discussions of potential future directions for investigation.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORK

2.1. Overview of Multiview Face Detection

Multiview face detection refers to the detection of human faces at different orientations, e.g., side faces, different lighting conditions, different facial expressions, occlusions, and different sizes, which are frequent in real-world environments. Traditional face detection methods generally fail to handle big changes in face orientations because they are based on human- designed features and assume that faces are in the forward direction. Deep learning has made tremendous progress, especially with CNNs. They allowed us to automatically learn important features from large datasets, drastically improving the way we recognize faces from different angles. A seminal paper by Zhang et al. (2016) introduced the MTCNN, which combined face detection and alignment into one system, making it better at detecting faces from multiple views. Even so, multiview face detection remains hard, particularly in the presence of extreme poses and occlusions, which still motivates research on pose-invariant representation learning and real- time capability on edge devices.

2.2. Multiview Face Detection Technique

2.2.1. Traditional Techniques on Multiview Face Detection

Before the advent of deep learning revolution, Multiview face detection relied primarily on hand-engineered features and conventional machine learning algorithms. Haar cascades, histogram of oriented gradients (HOG), and support vector machines (SVMs) were the pillars of face detection systems. Pose-specific classifiers were used by these methods to detect faces of varying orientations by training several models on frontal, profile, or tilted face datasets. For example, the Viola-Jones detector was extended to Multiview contexts by learning a number of classifiers between views, which were applied in parallel or cascade fashion to cover a bigger range of orientation (Li et al., 2004). While successful under constrained conditions, these methods suffered from sensitivity to illumination change, occlusion, and broke down in wild conditions because of shallow representation and rigid template matching strategies.

Another active direction in recent Multiview face detection followed probabilistic and kernel-based methods. Li et al. (2001) proposed using a probabilistic method that modeled variations in facial appearance between different views using Principal Component Analysis (PCA). Kernel-based machine learning models were then employed to effectively model the nonlinear

relationships among various views and facial orientations (Li et al., 2001; Fu et al., 2001). These methods demonstrated improved generalization but were still hampered by small training sets, computational expense, and poor ness to real-world noise. Despite their limitations, classical approaches enabled pose-aware face detection and offered valuable insights that would influence deep learning-based systems later on.

2.2.2. Deep learning model on Multiview face detection

Deep learning has brought tremendous enhancement in the ness of multiview face detection. Convolutional Neural Networks (CNNs) facilitate the automatic extraction of features from different facial directions, lighting conditions, and occlusions. A good example is the Multi-task Cascaded Convolutional Neural Network (MTCNN), which can detect faces and align them at the same time through the utilization of deep CNNs (Ma & Wang, 2018). Other notable architectures include the Deep Dense Face Detector (DDFD) and Multi-view Perceptron (Zhu et al., 2014), which explicitly learn shared representations from various face views. These networks excel in comparison to traditional approaches, leading to detection that is even in unconstrained conditions. State-of-the-art models also incorporate methods such as data augmentation, multi-tasking, and attention mechanisms in an effort to tackle variations in face appearances more effectively. For example, the Funnel-Structured Cascade (FuSt) detector uses deep features in conjunction with an alignment-aware approach to achieve improved accuracy amidst large pose variations (Wu et al., 2017). Deep Multi-View Representation Learning models likewise utilize several different views to enhance detection and recognition performance cooperatively (Li et al., 2016). These models not only improve multi-view detection but also allow for easy integration with follow- up processes such as recognition and expression analysis, thereby rendering them indispensable to contemporary biometric systems.

2.3. GAN

In technical terms, GANs are made up of two artificial neural networks called Generator and the discriminator and the generator are both trained together with the principles of adversarial learning. In addition, engage in a minimax game. The generator is also trained to deceive the discriminator D (x, 2) by generating realistic images that mimic the real data set are created, while the discriminator is trained to stay uninfluenced.by the generator. The fake images will be extremely noisy in the first few runs. As a result, the discriminator will receive the fake and real images as input and give the probability that it originated from a real dataset, prior to returning "feedback" to the generator through backpropagation step. Figure 2.1 illustrates the general structure of a Generative Adversarial Network. (Ben Aissa et al., 2024).

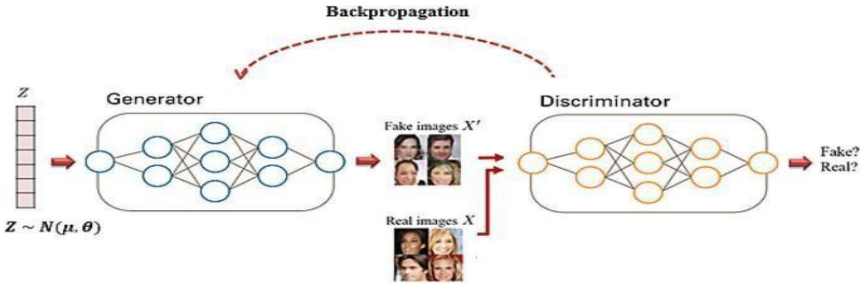


Figure 2.1: GAN structure.

Source: (Ben Aissa et al., 2024)

Edge-GAN integrates the target pose information and the input image to produce a primitive representation. Image with a rough target outline, which is then improved to a higher quality using adversarial training procedures. StarGAN (Choi et al., 2018) is the first to utilize a single generator network to create images in a variety of areas while maintaining the inherent characteristics of the produced goods. Feature information intact. DR-GAN(Tian et al., 2018) consists of a single-pathway framework to discover identity features that are invariant to a view. TP-GAN and CRGAN (Tran et al., 2017) have the same framework comprising two pathways. TP-GAN utilizes two novel encoder-decoder architectures to encode the out-of-plane rotation of the principal structure and the non-linear local texture transformation Generative Adversarial Network (GAN) based techniques can generate and synthesize realistic faces that cause profound social concerns and security concerns.

Multi-view face detection in the open scenario is a challenging issue due to the extreme variations in extreme pose angle, illumination, face expressions, and self-occlusion (Mohammadian et al.,2020 evidence indicates that the majority of faces captured in videos and images are not frontally captured, which consequently, the ability to process multi-view faces is a requirement for most face applications.

2.3.1.GAN for face Frontalization Techniques

Face frontalization is an essential part of multi-view face detection and recognition systems, aiming to generate high-quality frontal views of faces taken from non-frontal perspectives. Conventional methods have long been troubled by difficulties in maintaining the representative features of facial identity, particularly under adverse conditions,extensive occlusions, and diverse illumination. The advent of Generative Adversarial Networks (GANs) has revolutionized this process, and it is now feasible to generate photorealistic frontal faces with high structural integrity and invariant identity.

Some architectures of Generative Adversarial Networks (GANs) have been designed to solve the issue of pose normalization.occlusions like sunglasses, hair, or shadows. The Disentangled Representation GAN (DR-GAN) created a framework that learns pose-invariant representations by jointly optimizing identity discrimination and pose generation (Tran et al., 2017). The Two-Pathway GAN (TP-GAN) took the realism of synthesized images to another level by incorporating a global facial structure pathway with a local feature synthesis pathway, resulting in symmetrical and detailed frontal reconstructions (Huang et al., 2017). On the other hand, FF- GAN employed a 3D Morphable Model (3DMM) to exploit geometric priors for the sake of enhancing face reconstruction in examples with extreme pose variation.

Aside from these application-oriented architectures, general image-to-image translation generative adversarial networks have been applied to frontalization. Pix2Pix, a supervised conditional GAN, learns an explicit profile photo to frontal face mapping with paired datasets. Although it works well when images are well-aligned, the paired sample requirement makes it less scalable in less controlled settings (Isola et al., 2017). Alternatively, CycleGAN makes unpaired image translation possible by enforcing cycle consistency across the profile and the frontal domain. Although CycleGAN provides more flexibility for datasets without exact alignment, it can introduce identity inconsistencies or structural artifacts (Zhu et al., 2017). More broadly, Conditional GANs (cGANs) have been used for attribute-controlled frontalization, where pose or semantic direction helps to generate the desired frontal outputs. These models allow for fine-grained control but often require large annotated datasets to guarantee the maintenance of synthesis quality.

These GANs have established the state-of-the-art-level advancement in frontalization, CAPG-GAN (Couple-Agent Pose-Guided GAN) provides unique benefits for real-world application in multi-view face detection pipelines. Unlike the more computationally intensive GAN models, CAPG-GAN is a lightweight and efficient architecture specifically tailored to process radical pose changes up to 90 degrees with identity consistency. This model integrates spatially adaptive normalization with a dual-agent learning framework to enable synthesis via extreme changes in facial directions. It also proposes specialized loss functions for preserving identity-specific features along with attribute-level consistency at the same time, even in the presence of occlusions like sunglasses, hair, or shadows. What makes CAPG-GAN unique among practical systems is its easy compatibility with downstream detectors, particularly the Multi-task Cascaded Convolutional Neural Network (MTCNN). Its lightweight design and high-quality frontal output make it easily compatible with

MTCNN, which relies on clear and frontal inputs to realize optimal detection performance. When combined, CAPG-GAN serves as a frontalization process that converts multi-view inputs to normalized frontal representations, which are then detected and processed by MTCNN for landmark localization and bounding box.

2.4. MTCNN

Multi-task cascaded convolutional network (MTCNN) proposed to deal with face detection together and alignment, which include P-Net, R-Net,O-Net,three-stage multi-task CNNs. The generated image pyramid passes through three consecutive sub-networks for generating the candidate. windows and bounding box regression vectors used for calibrating the candidates. Explore the MTCN face detection model network structure, the three-stage network, the P-Net only is not conveyed in the complete connection layer. A crucial method of enhancing performance of the convolution network is weight sharing. The quantity of connections among neurons is lower than that of the fully connected neural network (Guo et al., 2020).

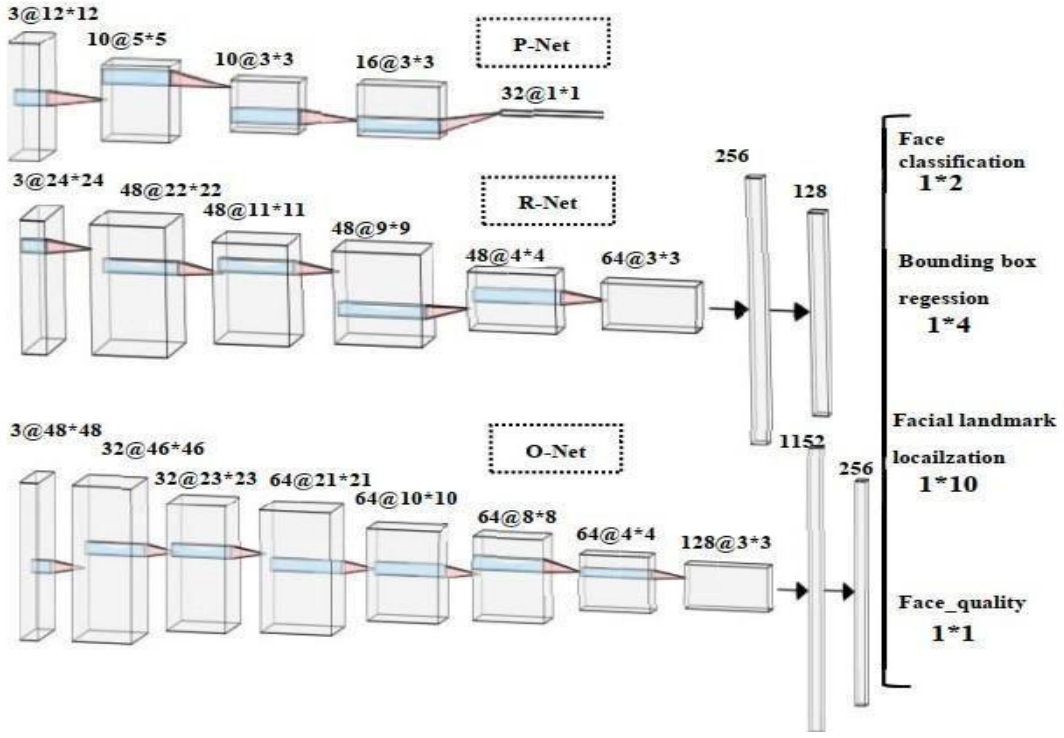


Figure 2.2: MTCNN structure.

Source: (Guo et al., 2020)

Ma and Wang, (2018) put forward the Multi-Task Cascaded Convolutional Networks (MTCNN) that unifies facial detection and landmark localization via a tri-stage framework using multi-task learning. The MTCNN is composed of three distinct stages, which include the following: Candidate windows are first generated rapidly using the initial layers of a convolutional neural network referred to as the Thesis Network (PNet). Then, the candidate windows are refined

according to the predicted bounding box regression vectors (K. Zhang et al., 2016).

In the second stage, refuses a large number of non-face windows through more complex CNN called Refinement Network (R-Net), which can further refine candidate windows(K. Zhang et al., 2016).These candidates in the next stage through an R-Net had refined. In the convolution layers, two kinds of the 3x3 filters and the 2x2 filters are mixed to convolution the input of 24x24 size. Finally, connect a fully connected layer. To achieve face classification and bounding box identification, the final layer performs a 1x1 convolutional filter with two independent channels (L. Zhang et al., 2021).

At stage three, a more powerful CNN, i.e., the Output Network (O-Net), refines the output once again(K. Zhang et al., 2016). O-Net gives the final bounding box and facial landmarks location. In comparison with R-Net, O Net includes one more convolution layer, and the complete connection layer is 256. It is also in a position to finish the location and output of the facial landmarks of the nose, left eye, right eye, left mouth corner, and right mouth corner of the face candidate box(L. Zhang et al., 2021).In MTCNN, weight sharing is important in reducing the amount of parameters as the network recycles weight in different layers. This lowers the amount of interconnectivity between neurons compared to a fully connected layer, thereby enhancing computational efficiency and lowering the risk of overfitting.

2.5. Related Work

Ma and Wang, (2018) Developed a three-stage multi-view face detection system using Multi-task Cascaded Convolutional Networks (MTCNN), which includes P-Net, R-Net, and O-Net, respectively; the model is in dealing with multi-view faces and has learned such large datasets as WIDER FACE, CelebA, FDDB, AFLW, and FERET.

Farfade et al., (2015) Deep Dense Face Detector (DDFD) leverages AlexNet and non-maximum suppression sliding window for multi-view face detection without pose labels. DDFD leverages AFLW, PASCAL Face, AFW, and FDDB datasets but suffers from class imbalance, occlusion handling, and bounding-box regression accuracy caused by annotation misalignment.

Huang & Alhlffee, (2023) TP-GAN face frontalization architecture, as outlined by Huang and Alhlffee (2023), is progressed through embedding a Decision Forest within the GAN discriminator while executing a data augmentation approach that is attentive to task-specific demands. Evaluation of the model was conducted on the Multi-PIE, FEI, and CAS-PEAL datasets.
datasets and outperformed TP-GAN on image quality and recognition performance. The model has been tested and shown to have a satisfactory performance but is still bounded by challenges like

extreme pose and contextual challenges under different environments.

Bhangale et al., (2022) Consequently, the suggested framework uses MTCNN for face detection and VGGNet for face recognition for varied poses and illuminations and has been experimented with in-house, ORL, and FERET datasets. Though the model is demonstrated to be accurate under laboratory-controlled environments, it is demonstrated to fail under extreme poses, illumination changes, and occlusions; furthermore, the computational time and storage requirements of VGGNet become the limiting factors for scalability to large datasets or real-time scenarios.

Zhong et al., (2022) To deal with the issues of gross detection errors and insufficient recognition accuracy in multi-view facial expression recognition, this paper presents a multi-view face detection and expression recognition approach using RetinaFace. It initially utilizes ResNet-50 as the base feature extraction network in RetinaFace to facilitate multi-view face detection. This successfully avoids overfitting issue brought by the heavyweight network ResNet-152 and the false detection issue brought by the lightweight network MobileNet-0.25 in the original RetinaFace algorithm.

Khan et al., (2024) provide some structural enhancements to improve the reliability and precision of facial detection by adding more neurons and introducing dropout techniques for regularization. Nevertheless, issues persist, most significantly false positive and partial occlusion issues, leading to misclassifications under scenarios with non-frontal or occluded facial states—a substantial quantity of MTCNN++.

Table 2.1 Summary of Related Work

Paper	Datasets	Method	Limitations
Multi-View Multi-Pose Face Recognition with VGGNet (Bhangale et al. 2022)	CASIA-WebFace,Multi-PIE,LFW	Combines MTCNN with VGGNet for recognition	Struggles with extreme poses, lighting variability, occlusions; complexity limits scalability for large datasets applications.
Multi-view Face Detection Using Deep Convolutional Neural Networks(2015)	Flickr Face Dataset (YFD)	Uses AlexNet with a sliding window approach and NMS	Struggles with imbalanced data, occlusions, and bounding-box regression accuracy problems due to annotation mismatches.
Improving Face Recognition by Integrating Decision Forest into GAN(2023)	Multi-PIE, FEI, CAS-PEAL	TP-GAN integrated with decision tree	Still struggles under extreme pose conditions, limited facial samples, and low-resolution data.
Multi-view Face Detection and Landmark Localization Based on MTCNN (Ma and Wang 2018)	Multi-PIE and AFLW	Utilizes MTCNN (P-Net, R-Net, O-Net) for multi-view face detection	Struggles with non-frontal faces, underperforms in frontal detection, and faces challenges in complex environments.
A multi-view face detection and expression recognition method with improved (ui Zhong, Bin Jiang, Nanxing Li,2022)	WIDER FACE, RAF-DB, AffectNet,	RetinaFace with Resnet 50	Faces challenges with extreme angles and occlusions, and requires sophisticated merging strategies for improved results.
MTCNN++: A CNN-based face detection algorithm inspired by MTCNN (Soumya Suvra Khan, 2024)	CelebA, Multi-PIE	Enhanced MTCNN	Faces issues with false positives, partial face (Extreme pose and occlusions) leading to misclassifications.

As stated in Table 2.1 have attempted to address multi-view face detection, but they face significant challenges in handling extreme poses and self-occlusions(Ma and Wang 2018).A multi-view face detection approach using MTCNN (P-Net, R-Net, O-Net), but it struggles with non-frontal faces and complex environments. Similarly, the Deep Dense Face Detector (DDFD) (2015) employs AlexNet with a sliding window approach and NMS; however, it suffers from occlusions, annotation mismatches, and bounding-box regression issues. (Huang & Alhlffee, 2023) TP-GAN face frontalization architecture is developed further by the model through the incorporation of a Decision Forest Still struggles under extreme pose conditions, limited facial samples, and low-resolution data.

More recent approaches, such as the multi-view multi-pose face recognition method by (Bhangale et al. 2022), combine MTCNN with VGGNet for improved ness; however, it struggles with extreme poses, occlusions, and lighting variability, making it less scalable for large datasets. (2022) A multi-view face detection and expression recognition method with improved however difficulties with extreme angles and occlusions, requiring additional merging strategies to enhance accuracy. MTCNN++ (Khan et al., 2024) improves upon MTCNN but still encounters issues with false positives and misclassifications in cases of extreme poses and partial faces. To address these limitations, our proposed approach integrates CAPG-GAN-MTCNN, leveraging the strengths of GAN-based frontalization and MTCNN for improved multi-view face detection. Unlike prior works that focus primarily on angles that not alleviate from frontal, our method incorporates and handles extreme pose angles up to 90˚, improved more accurate detection even in extreme pose challenges.

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1. Overview of Chapter

The following Figure 3.1 is strategies and tactics applied in order to accomplish the goal of the study. To realize the research goals outlined in the earlier section, this chapter now explores specific methodologies, procedures and techniques applied. This encompasses a full explanation of the datasets, augmentation of the data, the data preprocessing steps, overview of the baseline methods, and development tools. Finally, the evaluation metrics that were chosen to assess the model's performance.

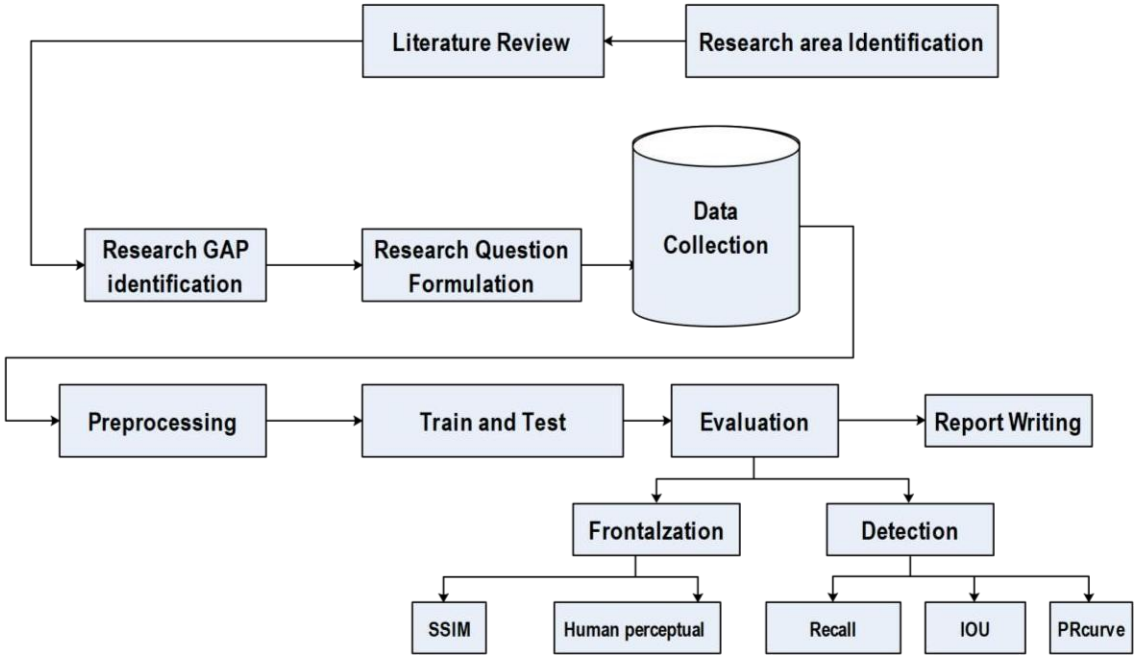


Figure 3.1: The Entire Research process

3.2.Data Collection

In this work, Multi-view face detection tackled pose invariant face datasets drawn from publicly available benchmarks in face detection, Multi-PIE RGB datasets (Gross et al., 2010) and CAS-PEAL-R1 gray scale datasets(Gao et al., 2007).

3.2.1.Multi-PIE Dataset

In total, the Multi-PIE database contains 755,370 images from 337 different subjects. Individual session attendance varied between a minimum of 203 and a maximum of 249 subjects. It provides images of subject ranging from -90° to 90° pose angles. Of the 337 subjects, 264 were recorded at least twice and 129 appeared in all four sessions (Gross et al., 2010).

Table 3.1: Summary of the Multi-PIE dataset.

Attribute	Definition
Purpose	Research in face recognition and detection under multiple controlled yet challenging conditions allows for investigations of pose and illumination invariance.
Subjects	337 individuals of different ages and sexes, offering demographical variety.
Total Images	Over 755,370, offering substantial data for machine learning models.
Pose Angles	include a full range of angles from -90° (left profile) to +90° (right profile), allowing for extensive pose variation.
Applications	For multi-view face detection, expression recognition, and illumination-invariant face recognition research.

Among the images in Multi-PIE, we used 16,275 facial images selected randomly for training and 3,487 images for testing. These images contain both moderate and extreme yaw angles, where face detection performance is generally poor. In order to ensure uniformity while training and testing, all the images were resized to 128×128 pixels. The chosen images were used in the frontalization process through CAPG-GAN and went through the face detection processes.

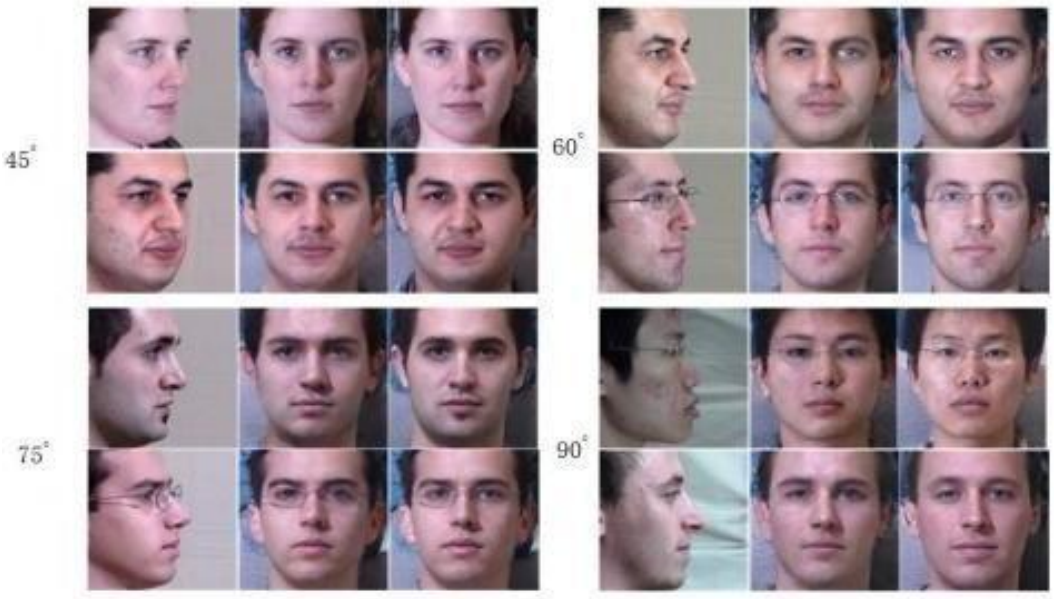


Figure 3.2: Multi-PIE dataset samples

Source: (Bao et al., 2023)

3.2.2. CAS-PEAL-R1

The CAS-PEAL face database is a large-scale Chinese facial dataset created to support global research in face detection and face recognition. Its primary objective is to provide a comprehensive resource that includes various sources of variation, particularly pose, expression, accessories, and lighting (PEAL). By incorporating these diverse factors, the database enables researchers to develop and evaluate face detection and face recognition algorithms under different conditions. Additionally, it offers exhaustive ground-truth information in a standardized format, ensuring consistency and accuracy in experimental studies(Gao et al., 2007).

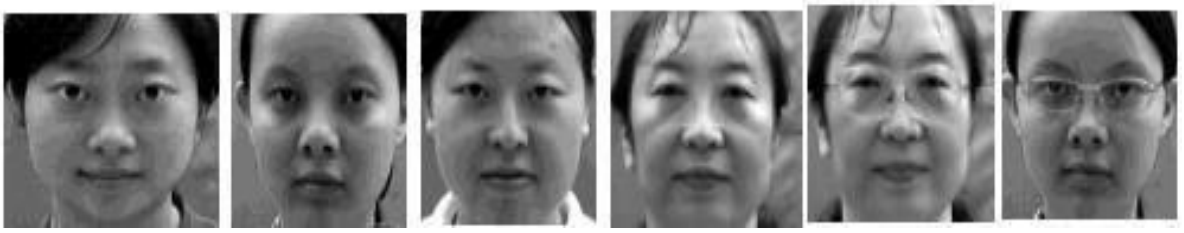


Figure 3.3:CAS-PEAL-R1 dataset samples

Source: (Gao et al., 2007)

Among the images in CAS-PEAL-R1, we used 14,469 facial images selected randomly for training and 3,100 images for testing. CAS-PEAL-R1 images were also resized to 128×128 pixels to ensure a uniform input size. Having this same data preparation step for both datasets guaranteed that they would be Compatible with the hybrid frontalization and detection system used in this study.

3.3. Pre-processing Techniques

In our research, we implemented the following pre-processing strategies to enhance data quality and optimize model performance.

◊ Brightness adjustment: We adjusted the brightness of image in range of -50 to 50.

◊ Scaling: We altered the dimension of the input images scaling their pixels, with in factor range size between 0.8 and 1.2.

◊ Color Jitter: involves arbitrary alteration of contrast and brightness.

◊ Resize: Resize Image resizing is a crucial pre-processing step. It involves adjusting the dimensions of the input images to match the model's expected input size. Resizing to a smaller size can minimize the Computational resources required and greatly speed up training. We resized the images in to $128x128$ to minimize the complexity of computation.

◊ Normalization: Normalization is a method of putting data into a consistent and comparable format. In this study, the dataset is normalized within the range of -1 to 1, which implies that the highest and lowest values for each attribute are constrained to -1 and 1, respectively.

3.4. Design and Development Tools

The study process benefited from the following software tools, including those for data preprocessing, model creation, and deployment.

3.4.1.Tool for Design

Edraw: is a diagramming tool that can be adapted to creating almost any kind of diagrams, including but not limited to organizational charts, network diagrams, processes, and flowcharts. Its user-friendly interface, combined with a wide array of shapes, symbols, and connectors, enables users to visualize even the most intricate connections and information in time.

3.4.2.Tools for Software

◊ Python [1]: Python is a language that many data scientists and machine learners use; it is popular because it's high-level and a general-purpose language.

◊ Tensor Flow [2]: is an open-source software platform to support machine learning and mathematical computation. It has resources and tools to design and deploy machine-learning models.

◊ Keras [3]: High-level Python API, based on Tensor Flow, to create deep learning models. It makes building and training a model fast and easy.

◊ Google Drive[4]: Used as a cloud storage solution, sharing and storing datasets are used in preprocessing and model training for Google Colab.

◊ Google Colab[5]: a free cloud-based Jupiter Notebook, no setup required, which offers for free computation resources like GPUs.

◊ Anaconda[6]: A Python distribution that handles Python environments and packages for data science and machine learning.

◊ Kaggle[7] offers unrestricted access to pre-installed Python libraries and free GPU and TPU acceleration, which cut down the training time for deep learning models considerably. Having access to NVIDIA Tesla P100 GPUs. This cloud infrastructure facilitated reproducible experiments, prevented local hardware dependencies, and enabled scalable model training within a consistent environment.

3.4.3.Hardware

The following hardware tools used to carry out the research.

Table 3.2: Hardware Components for Model Development

Component	**Description**
Hard Drive	Used for storing data.
GPU (Graphics Processing Unit)	Speeds up training time by parallel processing.
RAM (Random Access Memory)	Expedite the training period by handling large data volumes and speeding up computations.

[1]https://www.python.org/

[2]https://www.tensorflow.org

[3]https://keras.io/

[4]https://www.google.com/drive/

[5]https://colab.research.google.com

[6]https://www.anaconda.com/

[7]https://www.kaggle.com

3.5. Baseline Work

In order to compare our proposed model with the other existing more recent models in multi-view face detection, we have compared our model with the other existing models. As a baseline for this research work, we have selected MTCNN++, which is a CNN-based face detection algorithm derived from MTCNN. MTCNN++ was selected since it has a universal architecture and can identify faces in diverse poses; it is, however, reported to have difficulties with extreme poses and or partial face, which may result in false positives(Khan et al., 2024).

MTCNN++ architecture consists of three stages: a thesis network (P-Net), a refinement network (R-Net), and an output network (O-Net), all tuned for multi-task learning. The network is responsible for face detection by generating bounding boxes and facial landmarks. While MTCNN++ works extremely well for frontal face detection, its capacity for detecting faces with extreme poses or partial faces leads to false positives, especially if the face is at extreme angles or partially visible(Khan et al., 2024).

3.6.Feature Extraction Network

In the suggested CAPG-GAN model, Figure 3.4 VGG19 is utilized as an important feature extractor to calculate perceptual loss. The initial 16 layers of the pre-trained VGG19 model are utilized to extract high-level features from generated and real frontal images. The extracted features are compared to assess the quality of the generated images and reduce perceptual discrepancies between generated output and ground truth images. Using VGG19, the model can better capture more high-level and sophisticated visual content beyond low-level pixel-level differences, thus preserving semantic details in the frontalized images more effectively.

The utilization of MTCNN in face detection, with the VGG19 features, plays a crucial role in enhancing the accuracy of the frontal images produced. The MTCNN face detection algorithm is employed for detecting faces in the images generated, and perceptual loss from VGG19 is applied to ensure that the frontalized images are realistic and visually plausible. This integration of CAPG-GAN for frontalization with MTCNN for face detection and VGG19 as a feature extractor enables better accuracy in face generation and detection and therefore enhances the overall performance of the model.

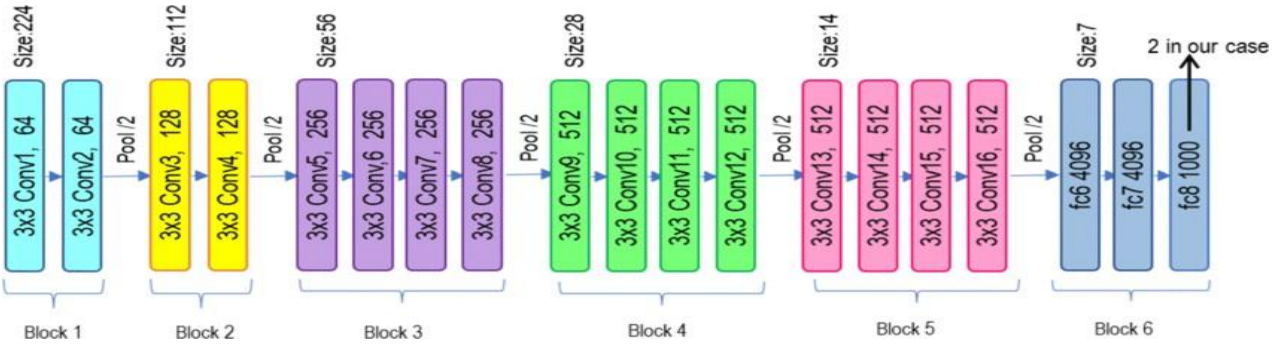


Figure 3.4 VGG19 architecture

Source (Sadhya & Qi, 2023)

3.7. Evaluation Metrics

Assessing the performance of deep learning model is the critical step in any research. Our proposed method employs the Structural Similarity Index to quantify the quality of frontal face images generated by the CAPG GAN. Human perceptual assessment is also conducted for qualitatively evaluating the performance of CAPG-GAN in generating visually plausible frontal face images. For evaluating the detection effectiveness of MTCNN in identifying generated frontal faces, we employ standard face detection metrics like Intersection over Union (IOU), Recall and Precision-Recall curve.

◊ Structural Similarity Index (SSIM): is used to measure the similarity between two images, focusing on how structural information is preserved. The SSIM algorithm provides surprisingly good image quality prediction performance for a wide variety of image distortions. An object's surface luminance is influenced by its luminance and reflectance, but for structural information, illumination data must be separated. Structural attributes are those that are independent of average contrast and luminance. SSIM involves three comparisons: contrast, luminance, and structure, and uses aligned signals X and Y. The first comparison estimates luminance as the mean intensity of the discrete signals(Naseem et al., 2023). It compares the original image and output image by extracting features: The equation for SSIM is given as (Wang et al., 2004).

SSIM Formula:

$$SSIM(i,j) = \frac{(2\mu_i\,\mu_j + c1)(2\,\sigma_{ij} + c2)}{(\mu_i^2 + \mu_j^2 + c1)(\sigma_i^2 + \sigma_j^2 + c2)} \qquad eq.\ (3.1)$$

Where:

◊ i and j are the two images being compared.

◊ μ_i and μ_j are the mean values of images i and j, respectively.

◊ σ^2 and σ^2 are the variances of i and j, respectively.

◊ σ_{ij} is the covariance between i and j.

◊ c_1 and c_2 are constants to stabilize the division when the denominators are weak (close to zero).

◊ Human perceptual Evaluation: To assess the perceptual effectiveness of the proposed CAPG-GAN frontalization model, we conducted a paired in-person evaluation using a direct face-to-face setup. Participants were individually shown both the original (real) and the frontalized (generated) face images side by side on a screen. After viewing the images, each participant was given a structured google form questionnaire to rate the frontalized outputs across five perceptual criteria: identity preservation, image quality, pose correction and realism.

Intersection over Union (IOU): This measures how much the predicted face bounding box overlaps with the actual bounding box. There is a common threshold for what counts as a correct detection is IOU. The other loss function that we make use of employs the value for Intersection over Union. The first definition of IOU can be written as:

$$\mathbf{IoU} = \frac{|T \cap P|}{|T \cup P|} \qquad eq. (3.2)$$

Where:

◊ T = Ground truth bounding box

◊ P = Predicted bounding box

◊ $T \cap P$ = Area of overlap between the ground truth and predicted boxes

◊ $T \cup P$ = Area of union of the ground truth and predicted boxes

IoU quantifies the accuracy of a predicted bounding box. The value ranges from **0** to **1**, where:

◊ **1** means perfect overlap.

◊ **0** means no overlap at all.

Recall:-Measures how many actual faces were correctly detected (**low false negatives** are preferred). The formula in your image represents,

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \qquad eq.\ (3.3)$$

Where:

◊ TP (True Positives): The number of correctly detected positive instances (e.g., correctly detected faces).

◊ FN (False Negatives): The number of actual positives that were not detected (e.g., missed faces).

Precision-Recall Curve: In the case of face detection, Precision-Recall (PR) Curve is an essential evaluation metric to estimate the performance of a face detection model. The curve graphically illustrates the balance between precision and recall, thereby providing valuable insights into the model's capability to correctly detect faces and reduce the false positive rate.

CHAPTER FOUR

4. PROPOSED SOLUTION

4.1. Chapter Overview

This chapter provides a thorough explanation of the architecture of the suggested hybrid CAPG GAN-MTCNN paradigm. Model selection, the model learning function, and pseudocode for model training are also covered in this chapter. In addition, presents the proposed hybrid approach combining CAPG-GAN (Couple agent Adversarial Pose-Guided GAN) for frontalizing extreme angle faces and MTCNN (Multi-task Cascaded Convolutional Neural Network) for detecting the frontalized faces. The method addresses the challenges of face detection at extreme angles (up to 90° pose angle) with Occlusion and under poor lighting conditions, identified as a gap in existing approaches.

4.2. Model Selection

4.2.1. GAN Models

Frontal face synthesis is particularly important to improve face detection and face recognition performance, particularly for non-frontal face images. Previous convolutional neural networks (CNNs) cannot maintain identity information when generating frontal faces since they cannot learn global structural relations effectively.CAPG-GAN addresses this shortcoming by employing adversarial learning combined with geometric constraints to generate identity-preserving high-quality frontal faces. The model adopts a global attention mechanism to ensure retention of facial information along with the creation of naturally looking results in the face in extreme pose variation. Using global and local feature extraction by CAPG-GAN improves significantly the effectiveness of face frontalization as an efficient solution for face detection.

4.2.2.CNN Models

Accurate face detection are of vital significance in face detection systems such that input images are effectively preprocessed before further processing. Multi-task Cascaded Convolutional Networks is a state-of-the-art approach that detects facial landmarks effectively and aligns faces with high precision and light weightness. Compared to traditional detection methods, MTCNN uses a cascaded deep neural network architecture, which simultaneously optimizes face detection, landmark localization and bounding box regression. This multi-task learning method improves detection performance, particularly in real-world scenarios. Through frontal generated face detection with MTCNN, overall reliability of face detection systems is enhanced.

4.3.Proposed Hybrid CAPG-GAN -MTCNN Model

The architecture of the hybrid proposed CAPG-GAN-MTCNN model for face detection using multiple views is displayed in Figure 4.1. The frontalization module in the model utilizes a CAPG-GAN while the model's face detection process is carried out using the MTCNN model. CAPG-GAN consists of a decoder, context module, pyramid module, and an encoder for which all of the components aim at generating the face images viewed realistically from any direction as frontviews.The generator is trained with a synergistic approach combining perceptual loss, multi-scale pixel-wise loss,adversarial loss and VGG19, thereby providing good-quality image synthesis with facial identity integrity and consistent pose. The discriminator, the Couple-Agent Discriminator (CAD), checks the authenticity of the synthesized images and also preserves consistency between the frontalized and the off-angle original ones. One significant adaptation of the initial CAPG-GAN model is the incorporation of an MTCNN-based face detection module.

After frontalization, MTCNN detects generated face images, measuring face visibility improvement by the GAN. Post-processing procedures such as image refinement and annotation mapping improve detection precision. With the integration of both the strength of GAN-based frontalization and hierarchical face detection of MTCNN, the model enhances the performance of multiview face detection and improved when there are extreme poses.

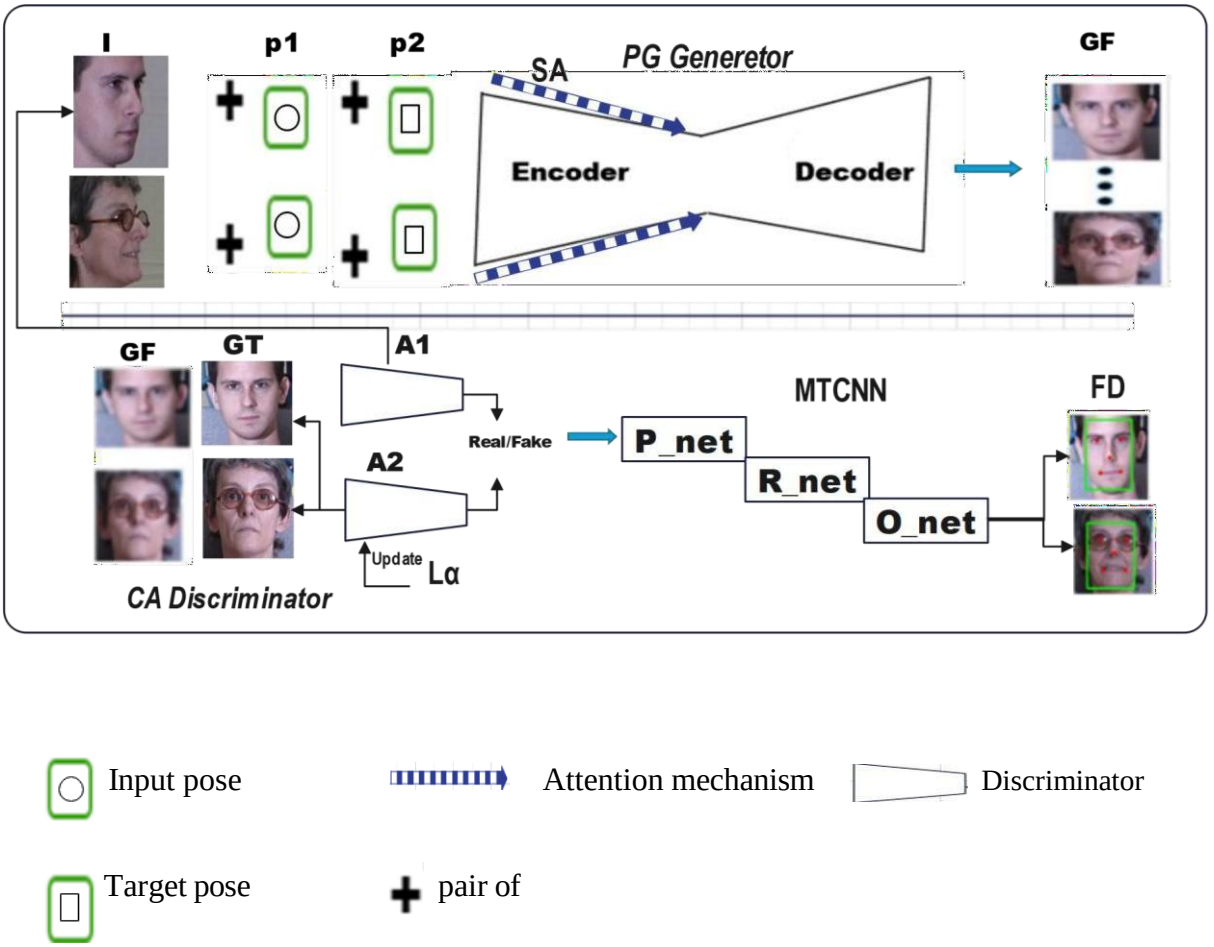


Figure 4.1: Proposed Model architecture

The architectural Figure 4.1 configuration starts with the input image (I-Input) of a face from a non-frontal view, which is fed pose information from image through: P1, the current pose, and P2(target), the desired frontal pose using heat map. Both are serve as direction inputs to the Pose Guided Generator (PG), which enables the transformation of the image into a frontalized image (GF). To enhance the quality of the generated frontal face, self-attention (SA) mechanisms have been integrated in the Pose Guided Generator. This allows the model to learn to dynamically pay attention to significant global features, retaining key facial features and context information for a more realistic result. The frontal image generated is then scrutinized by two separate discriminators, A1 (Agent 1) and A2 (Agent 2), both of which examine its genuineness from respective aspects. A1 checks GF consistent match with I and A2 checks consistent GF matches GT .The Couple Agent Discriminator (CA), further enhancing the output, conducts a combined assessment. MTCNN pre-trained is then used to detect the face in the frontal image generated (FD), thereby achieving precise face detection in this proposed approach. This structure effectively transforms off-angle faces to high-quality frontal views and consequently enhances the face detection performance.

In MTCNN P-Net, which generates a list of candidate bounding boxes in which facial features can be present.

The P-Net transforms the input image by applying a cascade of convolutional filters on it, yielding a list of feature maps. These feature maps are then passed through a series of fully connected layers to predict the probability of the presence of a face in the proposed regions of the image. R-Net is the second step in the Multi- task Cascaded Convolutional Networks (MTCNN) framework, which is tasked with refining the candidate bounding boxes from the P-Net. The network performs a twofold function; it not only refines the accuracy of the candidate bounding boxes but also determines the corresponding segments of the input image. The cropped images are resized to a fixed size and fed into several convolutional layers and fully connected layers in order to classify each bounding box as predictive of a face or not. The last step of the MTCNN algorithm is the O- Net, which further improves the precision of the bounding boxes and recognizes the facial landmarks. The O- Net takes the improved bounding boxes outputted by the R-Net and then crops the required region from the original image. The cropped areas are then reshaped to a pre-defined size and forwarded through a set of convolutional and fully connected layers in an attempt to categorize each bounding box into the class of having a face or not. The O-Net also regresses the bounding box coordinates to refine the accuracy of the position of the output face, along with detecting the coordinates of five facial landmarks (i.e., two eyes, nose, and mouth corners).

4.4.CAPG-GAN Architecture

The CAPG-GAN model consists of an encoder, a pyramid module, a context module, and a decoder in suggested approaches. The encoder consists of four convolutional layers that progressively downsample the input image, with ReLU activations being applied to introduce non-linearity. The pyramid module consists of two convolutional layers with residual connections to learn multi-scale features, enabling the model to learn representations that are more complicated. The context module is conditioned by two 1x1 convolutions to further enrich the feature representation. The decoder produces the frontalized face by four transposed convolutional layers, with a final Tanh activation function that outputs images in the range of [-1,1].

4.4.1.Pose Guided Generator

The Generator of the CAPG-GAN model is meant to transform off-angle faces into frontal faces. It consists of a three-part architecture as shown on Figure 4.2 below. The Encoder, Pyramid and Context Modules, and the Decoder. The Encoder is a convolutional layer stack that increasingly extracts information from the input face without losing important spatial details. The Pyramid and Context Modules allow to extract multi-scale features and shape feature representations, which is crucial in processing complex pose transformations. Finally, the Decoder produces the frontalized image via transposed convolutional layers, producing a Frontal Generated by CAPG-GAN output that is almost as good as the actual frontal face.

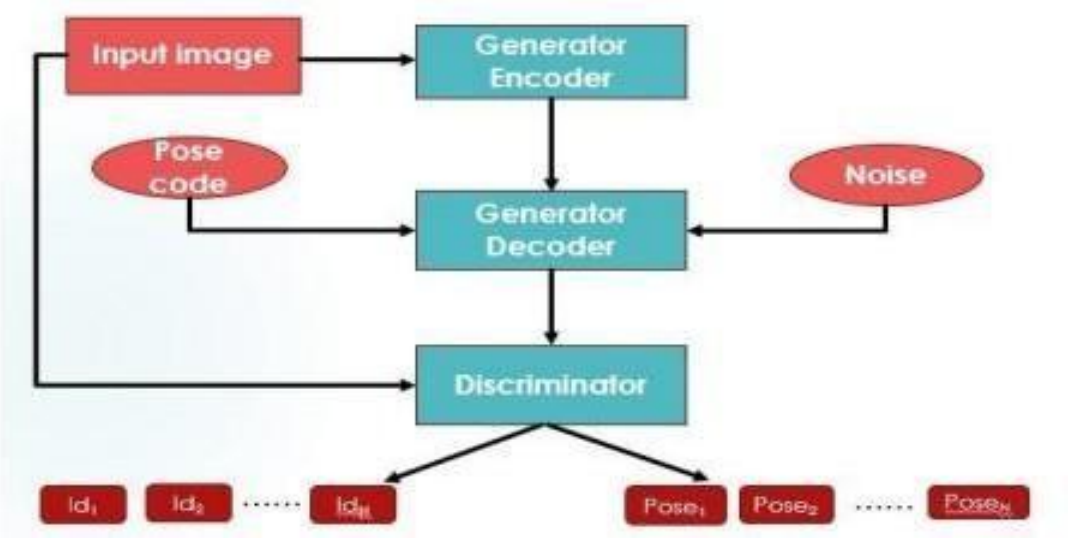


Figure 4.2: Pose Guided Generator

Source:(S & S H, 2020)

The encoder block in proposed architecture is a down-sampling block with one convolutional layer and then four down-sampling operations (conv1 – conv4) and following each down-sampling operation one self-attention block. Convolutional layer has kernel = 4 and stride = 2. Each convolutional layer is followed by the ReLU activation function and batch normalization. One self-attention block is followed after fourth layers of down-sampling operation to view the long-range relations and improve the feature representation.

The decoder block is an up-sampling block consisting of four deconvolution layers (deconv1 – deconv4). ReLU activation function and batch normalization are applied after each deconvolution layer, except for the last deconv4 layer. Tanh activation function is applied in the deconv4 layer to generate the final output.

1. Convolutional Layers (Encoder): The encoder component of the CAPG-GAN generator is constructed as a sequential stack of convolutional layers, designed to extract multi-scale semantic features from pose-varied facial inputs. It comprises four convolutional blocks, each with a 4×4 kernel, stride of 2, and padding of 1. The input image with 3 RGB channels and 1 channels gray scale is successively transformed into feature maps with increasing depth 64, 128, 256, and 512 channels—thereby reducing the spatial resolution while enriching the representation with high-level abstractions. ReLU activation functions are applied after each convolution to introduce non-linearity and promote efficient gradient flow during training. These encoder layers are pivotal in preserving essential facial attributes such as contour, texture, and pose-specific structure, which are then processed by pyramid

and context modules for enhanced representation. This hierarchical encoding ensures that the generator retains discriminative details necessary for high-fidelity frontal face reconstruction.

2. DE convolutional Layers (Decoder): The decoder in the CAPG-GAN generator mirrors the encoder in structure but performs the inverse operation—reconstructing the spatial resolution of the input face image through a series of transpose convolution (deconvolution) layers. It consists of four stages that progressively upsample the 512-channel attention-refined feature maps to generate a final 3-channel RGB image and 1-channel grayscale. Each layer uses a 4×4 kernel, stride of 2, and padding of 1, effectively doubling the resolution at each stage. ReLU activation functions follow each of the first three layers, while the final output layer uses a Tanh activation function to normalize pixel values within the range [-1, 1], matching the expected range of input images. This decoder design enables the network to synthesize realistic, pose-normalized facial outputs by progressively incorporating fine-grained details while maintaining spatial coherence learned through earlier encoding and attention stages.

4.4.2.Self-Attention Mechanism

In our proposed framework, Self-Attention is positioned strategically to the down sampling decoding steps for discovering long-range dependencies in the image. Self-Attention allows the model to pay attention to the spatial relationships between far-distant areas of the image, critical for operations like face frontalization where certain of the underlying facial features must be aligned. Self-Attention mechanism works by generating three components—query, key, and value—through convolutional layers and computing attention weights to re-weight the input feature map Figure 4.3. The process enables the model to successfully combine global information and local features with important details remaining even when the image is down sampling.

The Self-Attention mechanism is added during the down sampling phase after early feature extraction where it still adds to the model's global dependency capture capabilities between regions. By applying Self-Attention, our model can better process complex spatial configurations in images, which results in more accurate generation and positioning of facial features in frontalized faces. The use of Self-Attention results in more precise and uniform output such that important global information is retained throughout the processing pipeline.

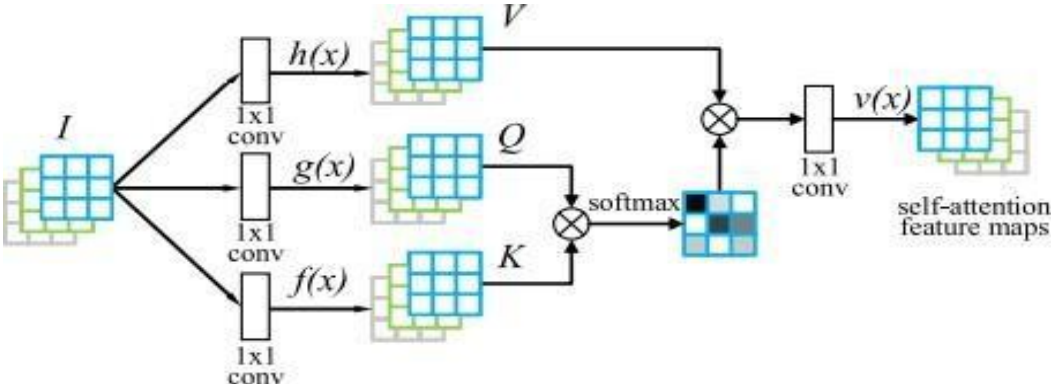


Figure 4.3: Self-Attention Mechanism

Source:(Zhu et al., 2022)

4.4.3.Couple Agent Discriminator

Couple-Agent (CA) Discriminator is designed to enhance the pose consistency and realism of frontal face images produced. The Discriminator compares a pair of images ground-truth frontal image and the generated frontal image by CAPG-GAN and decides if they are real or fake. The CA Discriminator architecture comprises four convolutional layers with LeakyReLU activation and a final sigmoid activation to output a probability score. This design allows the model to learn high- level hierarchical features from the pairs of images and to make an explicit decision boundary between synthesized and real faces.

Two complementary agents implement the discriminator. Agent 1 (A1) consumes pairs of generated faces (GF) and ground-truth faces (GT) as input, conditioned on the source image and a pose embedding (p1). A1 is worried about maintaining overall visual realism of the output face. Agent 2 (A2), with an identical architecture, conditions its selection on an alternate pose embedding (p2) to emphasize pose and structural coherence. Both agents operate independently performing real/fake classification and providing rich feedback to the generator that not only improves visual quality but also the pose accuracy of the generated faces.

After the generation and discrimination steps, their outputs are passed through a P_net,R_net, and O_net-based face detection pipeline of MTCNN. This process step ensures the generated images still possess the most important facial features needed for successful face detection (FD). With the combination of appearance realism and pose correctness, the couple-agent framework significantly enhances the generator's capacity to produce high-quality, frontalized face images suitable for downstream for face detection tasks.

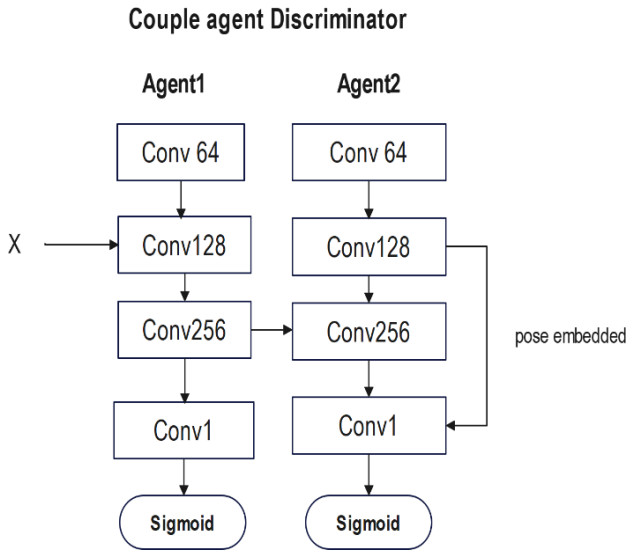


Figure 4.4: Couple Agent Discriminator

The Couple-Agent Discriminator is constructed by a stack of convolutional layers under the same architecture parameters to extract multi-scale features as shown in the above Figure 4.4. Each discriminator agent contains four convolutional layers, all of which use a 4×4 kernel size, stride 2, and padding 1. These settings progressively downsample the spatial dimensions of the input while maintaining important facial information. The stride of 2 reduces the image resolution at each layer by half, enabling the network to focus on increasingly abstract features. The padding of 1 keeps spatial dimensions in check and proportionate during downsampling. This structured layering enhances the discriminator's ability to model local textures and global structures necessary for distinguishing between real frontal faces and fake ones in face frontalization.

4.5.MTCNN on Generated Images

Once CAPG-GAN produces frontal faces, they are passed through MTCNN (Multi-task Cascaded Convolutional Neural Network) for face detection. MTCNN is a much-touted deep learning network charged with face detection and landmark location. MTCNN works through a series of stages, detecting the face first in the image, then finding key facial landmarks (eyes, nose, and corners of mouth).

The Frontal Generated by CAPG-GAN images are expected to have faces in a more frontal pose, which allows MTCNN to perform the following tasks. Face Detection**,** MTCNN detects the presence of faces in the Frontal Generated by CAPG-GAN images. Even if the input images were initially at extreme angles, the generator should have produced a front-facing image that MTCNN can accurately detect. In addition, Landmark Localization**,** MTCNN also locates five key facial landmarks (eyes,

nose, and mouth), which are crucial for understanding the alignment and accuracy of the generated frontal faces. If the key points are correctly localized, it indicates that the frontalization image process was successful detected. One of the most significant benefits of this approach is that CAPG-GAN generates images that are frontal pose, and therefore MTCNN has less difficulty detecting the faces accurately. Conversely, in the case of faces with very extreme angles like 90°, common face detection models find it difficult. But by generating a frontalized version of these faces, the Frontal Generated by CAPG-GAN images significantly enhance the ease with which MTCNN can detect and accurately position key points even from previously challenging angles.

The Cascaded Convolutional Networks Including Multiple Tasks supported suggested research.Multi-tasked Cascaded Convolutional Networks (MTCNN) are widely used on Deep Convolutional Neural Networks (DCNN) for face detection and subsequent facial feature detection(Dang & Tran, 2023). Their brief descriptions are presented in the following Section.

1) Multitasking: MTCNN is a multi-task cascaded convolutional network architecture which has the capability of performing a set of tasks simultaneously, including face detection, facial landmark detection (eyes, nose, and mouth), and bounding box definition. MTCNN enhances the efficiency of proposed task like face detection, landmark localization, and bounding box.

2) High reliability: MTCNN detects faces features by using a convolutional neural network (CNN).CNN is a stable, straightforward but effective network for image processing tasks. As a result, MTCNN is able to achieve excellent detection ness and reliability against light, noise, and different pose orientation.

3) High efficiency: MTCNN is made to be fast and efficient. MTCNN is able to process images with sequential feature localization and detection stages, complementing with methods like convolutional networks and local awareness convergence. The following three structured used in our research

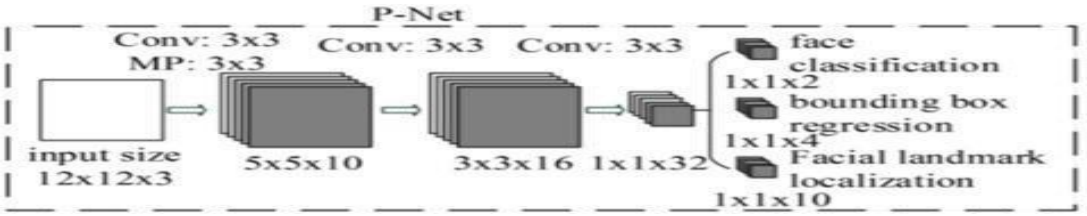


Figure 4.5: P-Net Network Structure

The diagram in Figure 4.5 results of P-Net are relatively rough, so further tuning is further done using R-Net. The R-Net structure diagram is shown in Figure 4.6. This structure is very similar to P-Net. It will enter into R-Net through the candidate window of P-Net, reject most of the false windows, and

continue to use bounding box regression.

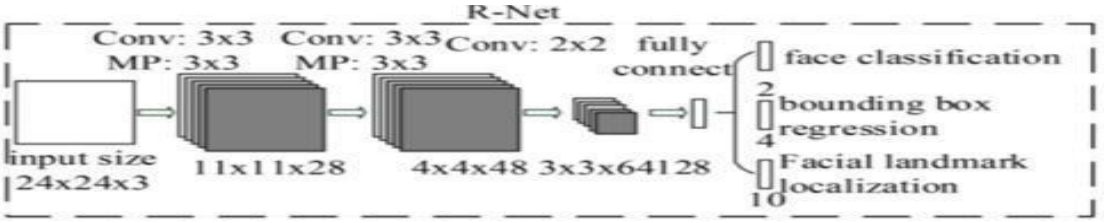


Figure 4.6: R-Net Network Structure

Finally, O-Net is used to output the end face frame and the feature point coordinates. As with the first two steps, the only difference is to generate 5-feature point coordinates. The network structure of O-Net is shown in Figure 4.7.

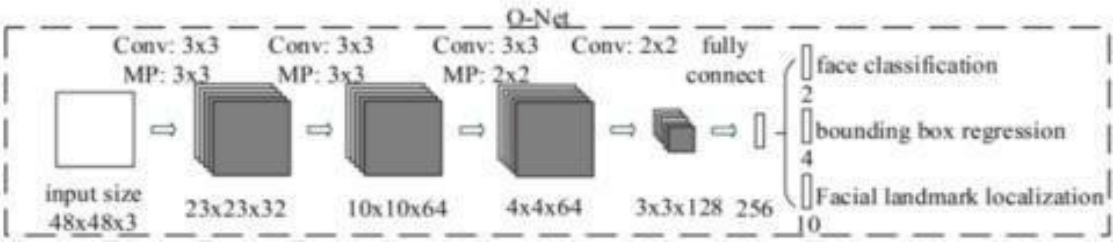


Figure 4.7: O-Net Network Structure

Source:(Dang & Tran, 2023)

4.6. Model Learning Function

4.6.1. Activation Functions

Different activation functions are used in the various components of the network for efficient learning. **ReLU** (Rectified Linear Unit) is used in the encoder, decoder, pyramid, and context modules of the generator to prevent vanishing gradients and enhance feature learning, and **Tanh** is used as the output activation function to normalize the pixel values to [−1,1] [−1,1] [−1,1] to accommodate normalized image datasets. In the discriminator, **LeakyReLU** (α=0.2) is used in convolutional layers to avoid the danger of neuron death and to provide a limited gradient flow for negative inputs, and Sigmoid is used in the final layer to map outputs to probabilities suitable for binary classification (real or fake)

4.6.2. Loss Functions

To achieve a balance among adversarial training, perceptual similarity, and pixel-wise reconstruction, various loss functions are used.

Adversarial Loss (Binary Cross-Entropy): Ensures that the generator produces images with a real while still permitting the discriminator to differentiate between real and fake images. Its equation is:

The equation in the image is:

$$L_{\text{GAN}} = -[\log D(y)] - \mathbb{E}[\log(1 - D(G(x)))]. \qquad eq.\ (4.1)$$

Content Loss (Perceptual Loss with VGG19): Helps the generator to equalize deep feature representations of synthetic and real images for better visual quality. It employs the formula

The equation in the image is:

$$\mathscr{L}_{\text{content}} = \|(y) - \phi(G(x))\|^2_{\ 2} \qquad eq.\ (4.2)$$

Where ϕ represents the VGG19 feature extractor.

Pixel-wise Loss (L2 Loss): Ensures pixel-to-pixel resemblance between the actual and synthesized images, formulated as:

$$\mathscr{L}_{L2} = \|y - (x)\|^2_{\ 2} \qquad eq.\ (4.3)$$

Multi-Scale Pixel Loss: It enforces ness by calculating L2 loss at multiple resolutions, designed as follows:

Multi-Scale Pixel Loss:

$$\mathscr{L}_{MS} = \sum_s \|y_s - (x)\|^2_{\ 2} \qquad eq.\ (4.4)$$

Description:

◊ $\mathscr{L}_{L2}$ is the standard L2 loss (mean squared error) between the ground truth y and the generated output (x).

◊ $\mathscr{L}_{MS}$ is a multi-scale version of the L2 loss, where the loss is computed at multiple resolutions s, helping the model to be across different scales.

4.6.3. MTCNN Learning Functions

Three-network MTCNN uses a variety of loss functions for detecting faces. P-Net uses binary cross-entropy as face classification loss and smooth L1 loss as the regression loss for improved localization. R-Net and O-Net both improve detection using binary classification loss and landmark loss (Euclidean loss on landmark locations). Here's a breakdown of the components:

$$\mathscr{L} = \mathscr{L}_{cls} + \lambda_{bbox}\mathscr{L}_{bbox} + \lambda_{lm}\mathscr{L}_{lm} \qquad eq.(4.5)$$

Where:

◊ $\mathscr{L}_{cls}$: Classification loss - measures how well the model classifies whether a region contains a face.

◊ $\mathscr{L}_{bbox}$: Bounding box regression loss -penalizes the error between the predicted and ground truth bounding boxes.

◊ $\mathscr{L}_{lm}$: Landmark loss -penalizes the error in the predicted facial landmarks (e.g., eyes, nose, mouth corners).

◊ λ_{bbox}, λ_{lm}: Weighting coefficients to balance the importance of the bounding box and landmark losses in the total loss.

4.7. Training Procedure pseudocode

The training procedure for the proposed model is outlined in the following pseudocode.

ALGORITHM: PSEUDOCODE FOR PROPOSED MODEL	
1	*Initialize pose guided generator, couple agent discriminator,* *MTCNN face detector*
2	*Input: Load Non-frontal views(I)* *Split dataset into training, validation, and test sets*
3	*Model parameters* *image_size = (128, 128, 3)* *batch_size = 32* *num_epochs = 150* *Number of epochs learning_rate = 0.0001* *Optimizer: Adam*
4	*for epoch in range (total_epochs): for batch in data_loader:* *Forward pass through the generator (G)* *Discriminator forward pass on real and fake images*
5	
6	*Save the Model and Results*
7	*Frontalized Non-frontal facial image* *Apply MTCNN* *Output: Detected Generated Frontal faces*

CHAPTER FIVE

5. IMPLEMENTATION OF THE PROPOSED SOLUTION

5.1. Chapter Overview

Deep learning solutions demand careful selection of suitable programming environments, frameworks, and hardware infrastructure in their deployment. The subsequent sections offer a comprehensive description of the hardware elements and software tools that have been adopted in developing the presented Hybrid CAPG-GAN with MTCNN model. The tools enabled experimentation on the CAPG-GAN architecture, MTCNN, optimizer parameters, and training techniques.

5.2. Working Environments

5.2.1. Hardware Components

Desktop

◊ Processor: Intel(R) Core(TM) i7-8700M CPU @ 3.20GHz 3.19 GHz

◊ Installed RAM: 8.00 GB

◊ Operating system: Window 10

◊ System type: 64-bit operating system, x64-based processor

◊ Graphics: Graphics: Intel HD Graphics 520/ NVIDIA

◊ GPU: GeForce RTX 2070 Super 8 GB RAM

Laptop

◊ Processor: Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz 1.90 GHz

◊ RAM Installed: 16.0 GB (usable 15.8 GB)

◊ Operating system: Window 11

◊ Window type: 64-bit operating system, x64-based processor

5.2.1. Software Components

◊ Python [1]: Python is a language that many data scientists and machine learners use; it is popular because it's high-level and a general-purpose language.

◊Tensor Flow [2]: is an open-source software platform to support machine learning and mathematical computation. It has resources and tools to design and deploy machine-learning models.

◊ Keras [3]: High-level Python API, based on Tensor Flow, to create deep learning models. It makes building and training a model fast and easy.

◊ Google Drive[4]: Used as a cloud storage solution, sharing and storing datasets are used in preprocessing and model training for Colab.

◊ Google Colab[5]: a free Jupiter Notebook cloud-based, no setup required, which offers for free computation resources like GPUs.

◊Kaggle[6]: Cloud-based data science platform that offers a free-hosted Jupyter Notebook environment.

Kaggle[7] offers unrestricted access to pre-installed Python libraries and free GPU and TPU acceleration, which cut down the training time for deep learning models considerably. Having access to NVIDIA Tesla P100 GPUs and Google Cloud TPUs offered high-performance processing for computation-intensive jobs like image classification and model optimization. This cloud infrastructure facilitated reproducible experiments, prevented local hardware dependencies, and enabled scalable model training within a consistent environment.

◊ Anaconda[8]: A Python distribution that handles Python environments and packages for data science and machine learning.

5.3. Training Procedure

To achieve the objectives of this study, the following processes were carried out:

◊ Data Collection: Collected multi-view face image datasets from publicly available repositories such as the CAS-PEAL-R1 and Multi-PIE datasets.

◊ Data Preprocessing: Resized all images to standard dimensions suitable for model input (128x128 pixels).Normalized pixel values to the range of [-1, 1] to suit the needs of both the GAN and MTCNN models and Data Augmentation.

◊ GPU Configuration:-Set up an NVIDIA GPU server for improved computational efficiency:

◊ implementing the Proposed Solution: Developed the hybrid model that combines the capabilities of CAPG-GAN for generating frontal face images and MTCNN for frontal generated face detection and landmark localization.

5.4. Dataset Preparation

5.4.1.Data Splitting and Loading

The data is divided into training (70% and), and test (15%) validation (15%) sets. For avoiding data leakage, the test set is divided first from the entire dataset, and then the remaining data is divided into training and validation sets. All input images are preprocessed by resizing them to 128 × 128 pixels to give consistent size, enhance computational speed, and reduce training time. Pixel intensities of images are normalized by rescaling them from their natural range [0, 255 to range [0, 1]. Normalization is beneficial for numerical stability, convergence acceleration in gradient-based optimization, equal handling of all color channels, and enhanced overall model performance.

Code Block 5.1: Dataset Loading and Splitting

```
# Load dataset
dataset = CustomDataset("/kaggle/input/kibooo/CAS-PEAL-R1/pose",
                        "/kaggle/input/kibooo/CAS-PEAL-R1/frontal", transform=transform)

train_size = int(0.7 * len(dataset))
val_size = int(0.15 * len(dataset))
test_size = len(dataset) - train_size - val_size
train_dataset, val_dataset, test_dataset = torch.utils.data.random_split(dataset, [train_size, val_size, test_size])
train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=32)
test_loader = DataLoader(test_dataset, batch_size=32)
```

5.5. Dataset preprocessing Implementation

As mentioned earlier, this study used a range of data preprocessing methods that were intended to be applied to the training images to make more preparedness of dataset and to improve performance of our model. Data augmentation techniques can create different versions of the images.

Code Block 5.2: Dataset augmentation

```
transform = transforms.Compose([
    transforms.Resize((128, 128)),
    transforms.ColorJitter(brightness=0.5, contrast=0.5),  # Remove saturation/hue to minimize noise
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5]),
])
```

5.6. Proposed Model Implementation

The suggested CAPG-GAN model is a pose-guided frontal face generation model consisting of a pose-guided generator and a couple-agent discriminator. The generator consists of an encoder, a pyramid module, a context module, and a decoder. The encoder employs convolutional layers with ReLU activations to extract hierarchical features. The pyramid module does multi-scale feature aggregation, and the context module improves the feature representations through residual connections. The decoder produces the frontalized face image from the aggregated features using transposed convolutions, which gives a Tanh activation to output pixel values between [-1, 1]. Pose guidance is integrated throughout the generator to generate proper frontalization from various input poses. The couple-agent discriminator consists of two parallel discriminators for improved supervision of both global layout and local details, each with convolutional layers with LeakyReLU activations and an extra Sigmoid layer for real/fake classification.

5.6.1.Pose Guided Generator Implementation

Table 5.1: Encoder Block

Encoder Block Layers	
	`def build encoder(self):`
	`return nn.Sequential(`
1	`nn.Conv2d(3, 64, kernel size=4, stride=2, padding=1),`
	`nn.ReLU(),`
2	`nn.Conv2d(64, 128, kernel size=4, stride=2, padding=1),`
	`nn.ReLU(),`
3	`nn.Conv2d(128, 256, kernel size=4, stride=2, padding=1),`
	`nn.ReLU()`
4	`nn.Conv2d(256, 512, kernel size=4, stride=2, padding=1),`
	`nn.ReLU()`
	`)`
`Average pooling(kernel size=(2,2), stride=(2,2),)`	

Table 5.1 describes the contents used in the encoder block of the generator. There are four convolutional layers utilized in the encoder block, each with a kernel size of 4×4, padding of 1 and a stride of 2. After each convolutional layer, a ReLU activation function is applied to introduce non-linearity and mitigate the vanishing gradient problem. Unlike the initial plan, Batch Normalization is not applied in the encoder block according to the provided implementation. Additionally, an average pooling layer with a kernel size of (2,2) and a stride of (2,2) is used after the convolutional layers to further downsample the feature maps and capture more abstract representations.

Table 5.2: Pyramid and context modules Layers Block

Pyramid and Context Modules Layers

```
  def build pyramid(self):
          return nn.ModuleList([
              nn.Conv2d(512, 512, kernel size=3, stride=1, padding=1
1             nn.ReLU(),
              nn.Conv2d(512, 512, kernel size=3, stride=1, padding=1),
2             nn.ReLU()
          ])

      def build context(self):
          return nn.Sequential(
1             nn.Conv2d(512, 512, kernel size=1),
              nn.ReLU(),

2             nn.Conv2d(512, 512, kernel size=1),
              nn.ReLU()
          )
```

Table 5.2 specifies the Pyramid and Context modules used in the generator. The Pyramid module consists of two convolutional layers, both having kernel size 3×3, stride one, and padding 1 to maintain spatial dimensions. A ReLU activation function is used after every convolutional layer to add non-linearity and allow the model to learn complex hierarchical features

The Context module is composed of two 1×1 convolutional layers, which are used to augment the feature representations by focusing on channel-wise transformations without altering the spatial size. As in the Pyramid module, each convolution is followed by a ReLU activation function to enhance the expressive capability of the model. Both of these modules capture both the multi-scale spatial information and contextual relationships that are vital for high-quality image generation together.

Table 5.3: Decoder Block

Decoder Block

```
    def build_decoder(self):
        return nn.Sequential(
1           nn.ConvTranspose2d(512, 256, kernel_size=4, stride=2, padding=1),
            nn.ReLU(),
2           nn.ConvTranspose2d(256, 128, kernel_size=4, stride=2, padding=1),
            nn.ReLU(),
3           nn.ConvTranspose2d(128, 64, kernel_size=4, stride=2, padding=1),
            nn.ReLU(),
4           nn.ConvTranspose2d(64, 3, kernel_size=4, stride=2, padding=1),
            nn.Tanh()  # Use Tanh for output to scale it between -1 and 1
        )
```

Table 5.3 defines the contents of the decoder block in the generator. The decoder block consists of four deconvolutional (transpose convolutional) layers with a kernel of 4×4, a stride of 2, and padding of 1. After each deconvolutional layer, a ReLU activation function is used to introduce non-linearity.The final deconvolutional layer uses a Tanh activation function to scale the output between -1 and 1. This smooth and controlled generation of image output is made possible by this configuration. The decoder block upsample the feature maps progressively, allowing images of high resolution to be reconstructed from encoded features.

5.6.2.Couple Agent Discriminator

Table 5.4: Couple Agent Discriminator

Couple Agent Discriminator

```
class CoupleAgentDiscriminator(nn.Module):
    def __init__(self):
        super(CoupleAgentDiscriminator, self).__init__()
        self.agent1 = self.build_discriminator()
        self.agent2 = self.build_discriminator()

    def build discriminator(self):
        return nn.Sequential(
1           nn.Conv2d(3, 64, kernel_size=4, stride=2, padding=1),
            nn.LeakyReLU(0.2),
2           nn.Conv2d(64, 128, kernel_size=4, stride=2, padding=1),
            nn.LeakyReLU(0.2),
3           nn.Conv2d(128, 256, kernel_size=4, stride=2, padding=1),
            nn.LeakyReLU(0.2),
4           nn.Conv2d(256, 1, kernel_size=4, stride=2, padding=1),
            nn.Sigmoid()
        )
    def forward(self, x, pose_embedding=None):
        output1 = self.agent1(x)
        if pose_embedding is not None:
            output2 = self.agent2(pose_embedding)
            return output1, output2
        return output1
```

Table 5.4: Couple Agent Discriminator Architecture: Couple Agent Discriminator consists of two parallel discriminators, named as Agent1 and Agent2. Both discriminators consist of four convolutional layers. All convolutional layers consist of a stride of 2, kernel size 4×4, and padding 1 for downsampling the input several times for capturing important spatial features.

Following each convolution, there is a LeakyReLU activation function with a negative slope of 0.2 to prevent the dying ReLU problem and enhance the learning of small gradients. After this final convolution, a Sigmoid activation function was used to acquire output values within the range 0-1, where the probability of real or fake images takes place. Within the forward pass, the input image is passed to the first agent and the second agent an optional pose embedding if any, enabling this discriminator to receive pose-specific information where relevant.

Table 5. 5:MTCNN Block

MTCNN Block

```
# === Apply MTCNN on Frontal Generated Images ===
# Input: Image I
# Output: Bounding boxes B, Landmarks L

function MTCNN_Detect(I):

    # Step 1: Build Image Pyramid
    pyramid = build_image_pyramid(I)

    # Step 2: Stage 1 - P-Net
    proposals = []
    for image_scale in pyramid:
        candidates = P_Net(image_scale)             # Get candidate boxes
        candidates = NMS(candidates)                 # Suppress overlaps
        proposals.extend(candidates)

    # Step 3: Stage 2 - R-Net
    refined_boxes = []
    for region in proposals:
        crop = crop_and_resize(I, region)
        result = R_Net(crop)                         # Refine candidates
        refined_boxes.append(result)

    refined_boxes = NMS(refined_boxes)

    # Step 4: Stage 3 - O-Net
    final_detections = []
    for region in refined_boxes:
        crop = crop_and_resize(I, region)
        output = O_Net(crop)                         # Final boxes +
landmarks
        final_detections.append(output)

    final_detections = NMS(final_detections)

    # Output final bounding boxes and facial landmarks
    B = [detection.box for detection in final_detections]
    L = [detection.landmarks for detection in final_detections]

    return B, L
```

Table 5.5: MTCNN Framework for Frontalized Image Detection: MTCNN is utilized for the detection of faces in frontalized images generated by the CAPG-GAN model. The MTCNN is first initialized with respect to the device configuration. The generate_and_detect function manages the detection process, where off-angle images are passed through a generator to generate frontalized output. These results are transformed to PIL images and pass through the three stages of MTCNN—P-Net, R-Net, and O-Net—to detect bounding boxes and facial landmarks. Detection results are stored for each image, and visualization plots bounding boxes and landmarks.

5.7. Hyperparameter Configuration

Hyperparameters are those parameters which are assigned to control the learning process of a our learning model. They are adjustable and greatly influence the performance of the model as well as the convergence characteristics of the model. In this study, a specified learning rate of 0.0001 with the Adam optimizer with optimized the model weights.

Table 5.6: Hyperparameter Configuration

Hyper parameter	**Values**
Epoch	150
Rate of Learning(G/D)	0.0001
Batch Size	32
Optimizer	Adam optimizer ($\beta1=0.5$, $\beta2=0.999$)

Table 5.6 adjusted parameters by utilizing AdaGrad and RMSProp (Kingma & Ba, 2014) in order to normalize the gradient updates against a rolling window of squared gradients, to push back against sparse data and non-stationarity of objectives. The learning rate determines the strength of each alteration to the weights of the model at updates, which affects how quickly the model learns. During our experimentation, we trained our model under different Hyperparameter configurations such as 50 epochs, 100 epochs, 150 epochs, batch size 32, and various learning rates to evaluate its performance across a range of setups. After iterative experiments, we selected the optimal Hyperparameter based on the model's stability and ability to generate high-quality frontal face images.

5.8. Experiment Class

Table 5.7: List of experiment class

Experiments	**Models**	**Datasets**
MTCNN++ Detection Evaluation	MTCNN++(Baseline)	Multi-PIE, CAS-PEAL R1
Integrate CAPG GAN and MTCNN Detection Evaluation	Hybrid CAPG GAN-MTCNN	Multi-PIE, CAS-PEAL R1
Comparison with CNN Models	MTCNN, Retina face, DDFD	Multi-PIE, CAS-PEAL R1

Table 5.7 presents a summary of the various experimental classes carried out in this work. The first experiment evaluates the baseline MTCNN++ model on the CAS-PEAL R1 and Multi-PIE datasets. The goal of this experiment is to investigate the failure modes of previous face detection approaches under large pose variations and occlusions.

In the second experiment, we present a hybrid model that integrates CAPG-GAN for face frontalization with MTCNN for detection on the synthesized frontal images, with the aim of overcoming the limitations observed in the baseline model. The CAPG-GAN approach is used to transform non-frontal facial image into high-quality frontal representations and, in turn, enable MTCNN with better accuracy and consistency in detection. Detection of these synthesized frontal images enables the model to significantly improve its ability to handle extreme pose angles up to 90°, which in most cases remain challenging for regular detection systems. This method overcomes key issues like pose variability-caused detection failure, false positives from partially occluded or partially hidden facial areas, and degradation of identity-preserving features. By leveraging generative normalization of facial representations, the CAPG GAN–MTCNN hybrid model gains improved ness and reliability under challenging multi-view facial detection conditions.

The third experiment is a comparative assessment of the suggested hybrid method with some other prevalent CNN-based face detection methods, i.e., DDFD, Retina Face and the conventional MTCNN model, on common datasets. This comparative study is performed to set a benchmark of the suggested method compared to prevailing models by analyzing parameters such as detection performance and strength in different poses and illumination levels. By subjecting all models to the same evaluation metrics, this study offers a helpful perspective on the relative weaknesses and strengths of existing detection architectures in comparison to the suggested approach.

CHAPTER SIX

6. RESULTS AND DISCUSSION

6.1. Overview of Chapter

This chapter presents the baseline work and additional various experiments face detection models. The evaluation uses the CAS-PEAL R1and Multi-PIE and datasets to compare MTCNN++ (Baseline). In addition, the suggested hybrid method that integrates MTCNN and CAPG-GAN for improving detection efficiency on frontalized images is presented. For evaluating the performance of our face detection technique, a comparative study with state-of-the-art existing methods is also conducted. The results are presented in a simple form through charts, graphs, and tables, making it easy to compare.

6.2.Experimental Results

We were conducted Multiple experiments to evaluate and used a number of iterations with different factors to train the model's network. In order to obtain comparable results for this study, we set up three using the same datasets, hardware setups, and software settings. Two datasets were used to test the effectiveness of our methodology: the Multi-PIE dataset (Bao et al., 2023) and CAS-PEAL-R1 dataset (Gao et al., 2007). Comparing our results with the state-of-the-art method reveals that the proposed CAPG GAN-MTCNN detects faces with their qualities, including self-occlusion, hair, skin, and eyeglasses, and obtains the optimal quantitative values according to quantitative assessment criteria.

6.3.Results based on Evaluation Metrics

We were conducted Multiple experiments to evaluate hybrid CAPG GAN with MTCNN-based face detection models for multi-view face detection. The existing models MTCNN++ (Baseline), were assessed their effectiveness in face detection tasks. To objectively assess the model's performance, we employed a two-step approach: frontalization and detection: The proposed method computes the quality of the generated frontal faces produced with the GAN by means of the Structural Similarity Index (SSIM), while Recall ,Intersection over Union (IOU) and Precision Recall curve are computed to assess the performance of proposed in detecting faces. Furthermore, human perceptual evaluation is performed to qualitatively evaluate the synthesis of visually plausible faces by CAPG-GAN. SSIM is a subjective measure that calculates image similarity in luminance, contrast, and structural detail, so that the synthesized frontal images preserve significant facial details. Human perception test involves displaying a few of the carefully selected front-facing images reconstructed from the multi- view face detection system in order to confirm the face detection realism and quality. IOU computes the intersection over union of predicted and ground-truth face bounding boxes. Recall calculates the Proportion of real faces correctly identified, minimizing false negatives.

6.3.1. Frontalization Results based on Evaluation Metrics

To measure the quality of facial frontalization images generated by the CAPG-GAN that is incorporated with MTCNN, we used the Structural Similarity Index (SSIM), a widely used perceptual measure for image similarity. The SSIM results of two standard datasets, i.e., CAS-PEAL- R1 and Multi-PIE, are presented in Table 6.1.

Table 6.1: SSIM Evaluation on CAS-PEAL-R1 and Multi-PIE dataset

S.NO	**Dataset**	**Experiment**	**SSIM**
1	CAS-PEAL-R1	CAPG GAN-MTCNN	91.45
2	Multi-PIE	CAPG GAN-MTCNN	94.5

The hybrid method of CAPG GAN-MTCNN proposed in this paper has also achieved an SSIM of 91.45 for the CAS-PEAL-R1 dataset and 94.5 for the Multi-PIE dataset, confirming a very strong structural and perceptual similarity between the frontal face images synthesized using the proposed system and their ground truth images. These evaluation confirm the success of the proposed model in retaining facial features with varying pose angles and illumination conditions.

6.3.2.Human perceptual Evaluation on Multi-PIE and CAS-PEAL-R1 dataset

To establish the perceptual quality of the CAPG-GAN-based frontalized face model, a systematic human study was performed on five willing subjects showed below on Multi-PIE and CAS-PEAL R1 databases. The subjects evaluated the frontalized outputs produced for six pose angles (15°, 30°, 45°, 60°, 75°, and 90°) achieved from both datasets on Table 6.2 and Table 63 below respectively. The evaluations were gathered using a systematic Google Form and examined across four critical dimensions: identity preservation, image quality, pose correction, and realism.

Table 6.2: Human perceptual Evaluation Multi-PIE Datasets

Pose	Identity	Image Quality	Pose correction	Realism
15	5	5	5	5
30	5	5	5	5
45	5	5	5	5
60	4.9	4.9	5	4.9
75	4.9	4.9	4.9	4.9
90	4.8	4.8	4.8	4.8

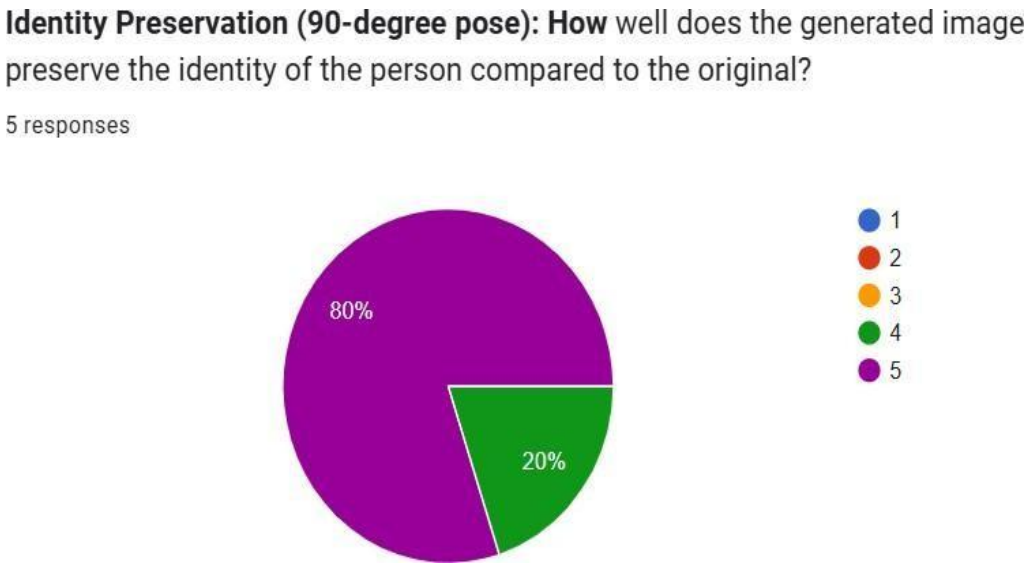


Figure.6.1 Pie Chart Multi-pie dataset

Perceptual evaluation of the synthesized frontal faces was conducted using a Google Form survey MULTI-PIE Datasets result (Figure 6.1).The model obtained perfect scores (5.0) in all categories for 15°, 30°, and 45° poses, thereby validating its capacity for preserving visual consistency under near-frontal views. At a pose angle of 60°, there was a moderate decline in scores, with averages of 4.9 for the identity, image quality, and realism metrics. At the harder angles of 75° and 90°, the model still performed exceedingly well, recording average scores of 4.9 and 4.8, respectively, on the metrics that were calculated.

Table 6.3: Human perceptual Evaluation on CASPEAL R1 Datasets

Pose	Identity	Image Quality	Pose correction	Realism
15	5	5	5	5
30	5	5	5	5
45	5	5	5	5
60	4.8	4.8	4.9	4.8
75	4.8	4.8	4.8	4.8
90	4.6	4.6	4.6	4.6

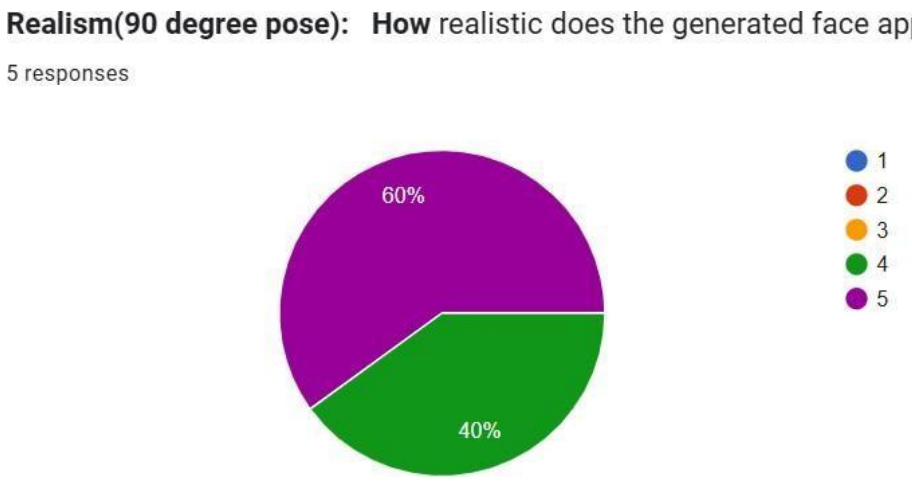


Figure.6.2 Pie Chart for CASPEAL dataset

Perceptual evaluation of the synthesized frontal faces was conducted using a Google Form survey result on CASPEAL-R1 datasets (Figure 6.2).The model scored the highest (5.0) for 15°, 30°, and 45°, reflecting excellent reality preservation and image quality. At 60° and 75°, the model rated 4.8 across all 4 criteria, and at 90°, it rated 4.6 for identity, image quality, and pose adjustment, and 4.6 for realism.

6.3.3. Detection Results based on Evaluation Metrics

Table 6.4: Evaluation of MTCNN++ and Proposed model on Multi-PIE Dataset

Model	Evaluation Metrics	15°	30°	45°	60°	75°	90°	Average
MTCNN++(Baseline)	IOU	0.92	0.88	0.82	0.78	0.70	0.65	0.792
	Recall	0.98	0.94	0.88	0.86	0.85	0.80	0.885
CAPG GAN-MTCNN(Proposed)	IOU	0.98	0.97	0.96	0.95	0.935	0.90	0.949
	Recall	0.99	0.985	0.98	0.975	0.970	0.96	0.976

In evaluating the performance of the proposed CAPG GAN-MTCNN hybrid model for Multiview face detection; we compared its performance against the baseline MTCNN++ model across multiple pose angles ranging from 15° to 90° on Multi-Pie dataset. The evaluation metrics considered include Intersection over Union (IOU) and Recall. As shown in the results, the baseline MTCNN++ model exhibits noticeable performance degradation as the pose angle increases. Specifically, the IoU decreases from 0.92 at 15° to 0.65 at 90° and Recall drops from 0.98 to 0.80. These results suggest that MTCNN++ struggles to maintain in detecting faces under extreme pose variations with noticeable performance beyond 75° pose angle.

In contrast, the proposed CAPG-GAN-MTCNN model demonstrates superior consistency and performance across all pose angles. The IOU metric remains high and stable, starting at 0.98 for 15° and only slightly decreasing to 0.90 at 90°, with an average IOU of 0.949. Similarly, the Recall metric remains strong throughout, ranging from 0.99 at 15° to 0.96 at 90°, averaging 0.976, indicating reliable face detection even under severe pose deviations.

These results clearly illustrate that integrating CAPG-GAN for face frontalization significantly improved detection performance in challenging Multiview scenarios. The hybrid framework approach effectively mitigates the limitations of existing face detection frameworks like MTCNN++, particularly in cases involving large pose variations

Table 6.5: Evaluation of MTCNN++ and proposed model on CAS-PEAL-R1 Dataset

Model	**Evaluation Metrics**	**15°**	**30°**	**45°**	**60°**	**75°**	**90°**	**Average**
MTCNN++(Baseline)	IOU	0.90	0.86	0.79	0.74	0.65	0.60	0.756
	Recall	0.97	0.905	0.895	0.85	0.84	0.70	0.86
CAPG GAN-MTCNN(Proposed)	IOU	0.97	0.96	0.95	0.92	0.915	0.90	0.935
	Recall	0.98	0.975	0.95	0.94	0.920	0.91	0.95

In Table 6.5 presented result clearly indicate that the proposed CAPG GAN-MTCNN model outperforms the baseline MTCNN++ across all evaluation metrics Recall, Intersection over Union (IOU) on the CAS-PEAL-R1 dataset. The most notable improvements are observed at higher pose angles (75° and 90°), where previous detection models typically suffer performance degradation due partial face invisibility.

In terms of IOU, the CAPG GAN-MTCNN achieves an average of 0.935, compared to 0.756 by the baseline, highlighting its superior spatial localization of facial regions. Similarly, the average Recall improves from 0.86 to 0.935, indicating that the proposed model is more in identifying all relevant face regions, even under extreme pose variations.

These outcomes confirm the efficacy of integrating CAPG-GAN-based frontalization with MTCNN for Multiview face detection tasks. The enhanced performance is especially critical in surveillance, security, and real-world applications where face detection under diverse pose variations is a major challenge.

The precision-recall curve Figure 6.3 below illustrates the performance of the proposed hybrid CAPG GAN-MTCNN model on multi-PIE and CASPEAL R1 dataset, which achieves Area under Curve (AUC) of 0.985 and 0.96 respectively.

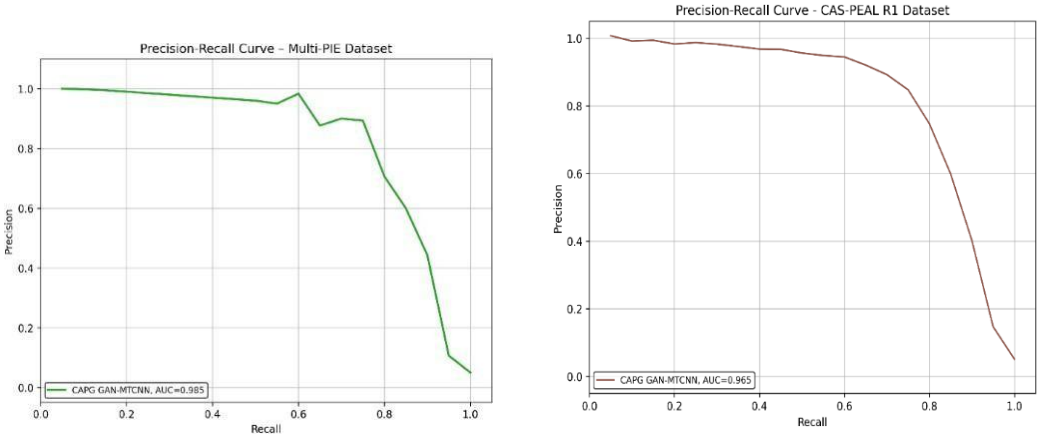


Figure.6.3: PR curve for Hybrid CAPG GAN-MTCNN

This high AUC reflects strong face detection capabilities, with high precision maintained even as recall increases. The model's can be attributed to the integration of CAPG GAN and MTCNN for frontal generated detection. The CAPG GAN component effectively frontalizes non-frontal faces rotating profiles with pose deviations of up to 90 degrees thus enabling more consistent and accurate detection.

This results in a highly resilient system that performs well in real-world conditions, where facial orientations vary significantly. The step and sustained curve on the plot indicates the model's strong performance across a range of recall levels, confirming its reliability and adaptability in unconstrained environments.

6.3.4.Detection Results of Experiments

In this Figure 6.4, the detected facial images for experiments, including the baseline model, are presented. For this experiment, we use dataset Multi-PIE and CAS-PEAL R1 outperformed on Multi-PIE dataset. The first column represents the input pose angle, while the second column shows the profile or non-frontal input image. The next two columns display the face detection MTCNN++ before frontalization, representing the non-frontal detection outcomes. The final three columns correspond to our implemented model, showing the ground truth (GT), the frontal generated image (FG), and the generated detected (GD).

Detection Results of Experiments sample for Multi-PIE Dataset

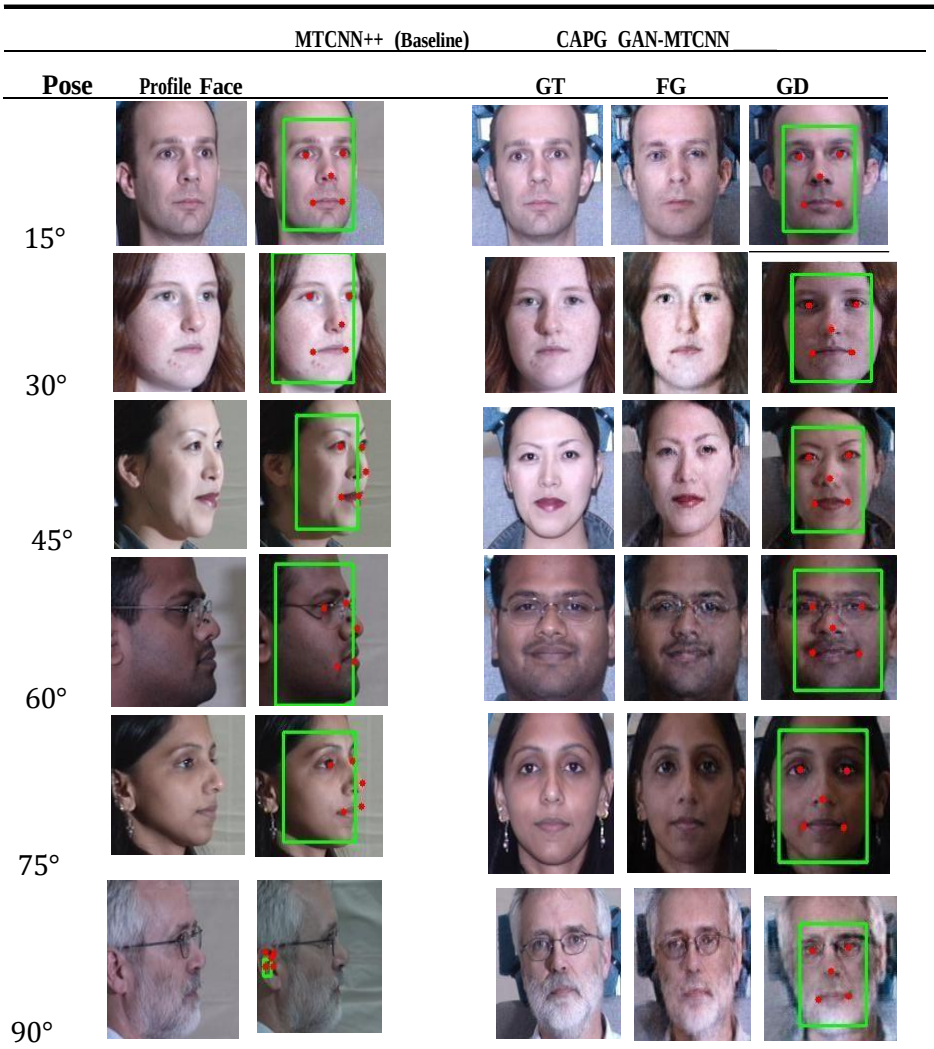


Figure 6.4:Multi-PIE facial images detection in different experiment classes.

While using MTCNN++, we experienced false positives since they are more sensitive to partial faces. Most of the time, a portion of the face including the forehead, ear or the chin was incorrectly classified as a whole face. This is more apparent at extreme angles, which results in a higher rate of false positives. At angles above 90° MTCNN++, possess a greater false positive rate, as they cannot detect faces at extreme poses correctly. Nevertheless, as indicated images generated effectively, which greatly improve detection performance. The column line in Fig 6.4 presents various pose angles: 15°, 30°, 45°, 60°, 75°, and 90°. Column three also MTCNN++ is also depicted with an increased false positive rate as it has the propensity to detect partial face areas. For instance, as shown in the third column of Fig 6.4 at 90° pose the detected faces by MTCNN++ are displayed it detected the ear as whole face.

As stated on Figure 6.5 MTCNN++, possess a greater false positive rate, as they cannot detect faces at extreme poses correctly. Nevertheless, as indicated images generated effectively, which greatly improve detection performance.

Detection Results of Experiments sample for CAS-PEAL-R1 dataset

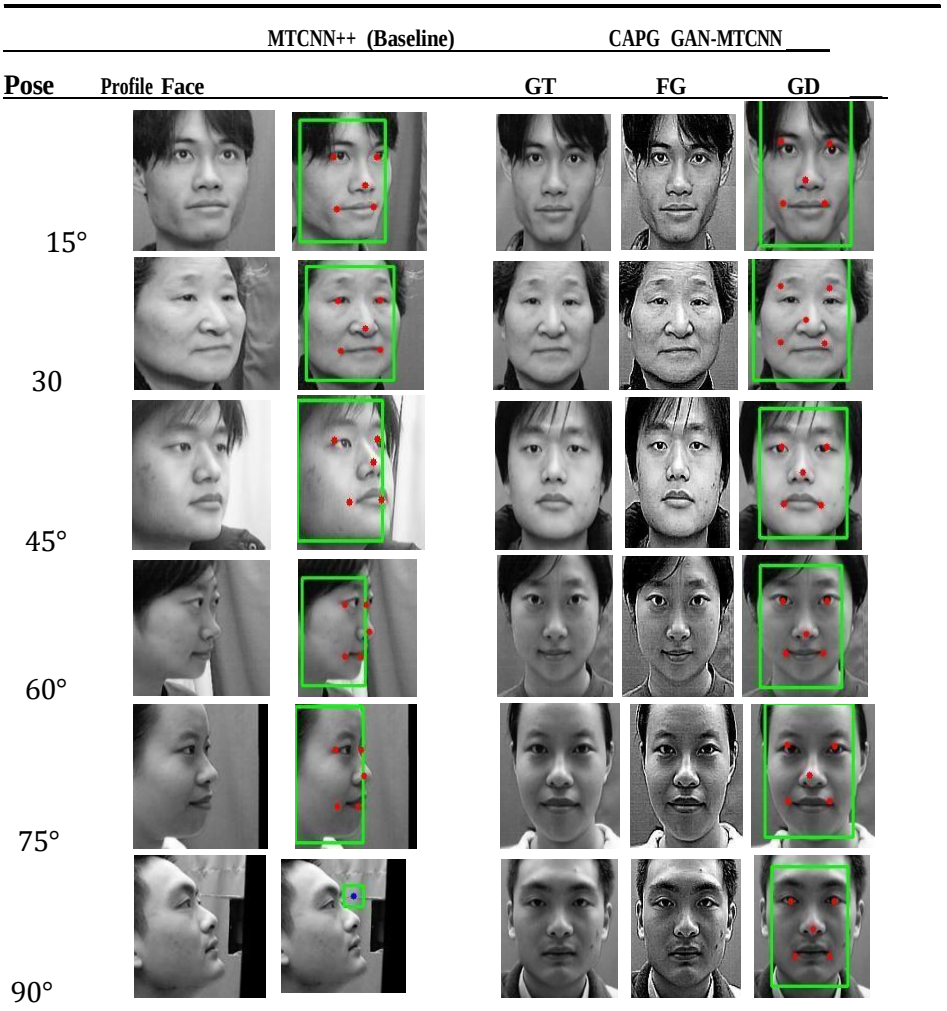


Figure 6.5: CASPEAL-R1 facial images detection in different experiment classes.

As stated on Figure 6.4 MTCNN++, possess a greater false positive rate, as they cannot detect faces at extreme poses correctly. Nevertheless, as indicated images generated effectively, which greatly improve detection performance. The column line in Fig 6.5 presents various pose angles: 15°, 30°, 45°, 60°, 75°, and 90. The third Column on MTCNN++ is also depicted with an increased false positive rate as it has the propensity to detect partial face areas, for example, in Fig 6.5, column three, **non-face region** has been falsely detected as a face.

As stated in the last column on both dataset, our proposed CAPG-GAN with MTCNN model detect the frontal generated. Our method addresses this problem by hallucinating high-quality frontal faces, which improve face detection accuracy even at extreme poses such as 90°. This shows the power of

our CAPG-GAN with MTCNN model in reducing false positives and increasing in adverse pose conditions.

6.4. CNN Models Performance

Table 6.6: CNN models Detection Models for Evaluation

Model	Evaluation Metrics	15°	30°	45°	60°	75°	90°
DDFD	IOU	0.79	0.74	0.68	0.62	0.55	40
	Recall	0.85	0.81	0.70	0.66	0.59	45
RetinaFace	IOU	0.88	0.84	0.79	0.74	0.69	0.50
	Recall	0.95	0.92	0.89	0.80	0.75	0.60
MTCNN	IOU	0.84	0.80	0.77	0.73	0.55	0.48
	Recall	0.96	0.93	0.88	0.81	0.59	0.67

Tables 6.6 presents a comprehensive performance comparison of four face detection models MTCNN, DDFD, Retina Face, and the proposed CAPG GAN-MTCNN across varying face orientations (15°, 30°, 45°, 60°, 75°, and 90°). Evaluation metrics including Intersection over Union (IoU) and recall were used to assess ness under multi-view conditions. Results indicate that all models experience a decline in performance as the face orientation deviates from the frontal view; however, the proposed CAPG GAN-MTCNN consistently outperforms the others, particularly at extreme angles.

The RetinaFace model performed well under frontal and near-frontal conditions, with its best performance at 15° (IOU: 0.88 and Recall: 0.95). Nevertheless, its accuracy dropped significantly at higher angles, with an IoU of 0.50 and Recall of 0.60 at 90°. MTCNN showed better resilience at mid-range angles but similarly declined at extreme profiles (IoU: 0.48, Recall: 0.67 at 90°). DDFD also achieving an average IoU of 0.40 and maintaining reasonable 0.67 recall at 90°. However, the proposed CAPG GAN-MTCNN model outperformed all others, leveraging GAN-based frontalization to enhance detection at extreme angles. It achieved an IoU of 0.90 and Recall of 0.96 at 90° on MULTIPIE datasets and IoU of 0.90 and Recall of 0.91 at 90° on CASPEAL-R1 dataset, while still maintaining strong results at 90° (IoU: 0.72 And Recall: 0.74). These results affirm that the proposed hybrid model offers superior detection accuracy for multi-view face detection.

6.4.1.Comparison with the Proposed Model

The performance comparison of the proposed CAPG GAN-MTCNN model with other state of-the-art face detection methods, including MTCNN, DDFD, Retina Face, and the baseline MTCNN++, shows considerable gains in all the measures of performance, as can be seen from Tables 6.7.

Table 6.7: CNN models Detection Models for Evaluation Multi- pie

Model	Evaluation Metrics	15°	30°	45°	60°	75°	90°
DDFD	IOU	0.79	0.74	0.68	0.62	0.55	40
	Recall	0.85	0.81	0.70	0.66	0.59	45
Retina face	IOU	0.88	0.84	0.79	0.74	0.69	0.50
	Recall	0.95	0.92	0.89	0.80	0.75	0.60
MTCNN	IOU	0.84	0.80	0.77	0.73	0.55	0.48
	Recall	0.96	0.93	0.88	0.81	0.59	0.67
MTCNN++	IOU	0.90	0.86	0.79	0.74	0.65	0.60
	Recall	0.97	0.905	0.895	0.85	0.84	0.70
CAPG GAN with MTCNN	IOU	**0.97**	**0.96**	**0.95**	**0.92**	**0.915**	**0.90**
	Recall	**0.98**	**0.975**	**0.95**	**0.94**	**0.920**	**0.91**

To identify the best result in experiment, we have highlighted the best results in bold as shown above in (Table 6.7). The performance of the baseline MTCNN++ model decreased noticeably with the increase in the pose angle. IoU decreased from 0.92 at 15° to 0.65 at 90° and Recall from 0.98 to 0.80 on the Multi-PIE database MTCNN and RetinaFace tend to perform better at lower angles but had difficulty with extreme poses. MTCNN scored an IoU of 0.55 and a Recall of 0.59 when the angle was 90°, and RetinaFace scored an IoU of 0.69 and a Recall of 0.75.

The proposed CAPG GAN-MTCNN model outperformed, resulted in an IoU of 0.90 and a Recall of 0.96 on the Multi-PIE databases despite very extreme angles. The CAPG GAN-MTCNN model outperforms other models mainly due to the fact that it utilizes CAPG-GAN to transform side faces to front faces. This process generates clear front faces of individuals, even if they are facing extreme angles (75° and 90°).

Meanwhile, the CNN models, MTCNN, DDFD, and Retina Face, showed a sharp degradation in IoU and Recall performance at bigger angles (75° and 90°). The CAPG GAN-MTCNN model had high

scores with Intersection over Union (IoU) and a Recall rate of across both datasets, as indicated in Tables 6.7. This consistent performance confirms the effectiveness of the CAPG GAN-MTCNN hybrid technique to offer precise detection of faces at varied angles. As it is, this method has high relevance to real-life application across fields such as surveillance and security.

6.5.Research Question and Answer Discussion

All the research questions posed in this study were successfully addressed.RQ1: To what extent can our proposed model achieve better performance than baseline? RQ2: To what extent can hybrid GAN-MTCNN model detect extreme pose angles in Multiview image? RQ3: How does the proposed hybrid approach compare to existing models regarding image detection accuracy?

Answer RQ1: The CAPG GAN-MTCNN model significantly outperforms the baseline MTCNN++ in multi-view face detection, especially at extreme pose angles greater than 75°. As MTCNN++ IoU decreases from 0.92 at 15° to 0.65 at 90°, and Recall drops from 0.98 to 0.80, the hybrid model's IoU is still 0.90 and Recall is also 0.96 even at 90°. This is a proof that our model improves detection ness and accuracy by around than the baseline for challenging pose conditions.

Answer RQ2: The study found that the CAPG_GAN integrated with MTCNN model is capable of accurately detecting faces with extreme pose variations, particularly within the range of upto 90°. By leveraging a Generative Adversarial Network to synthesize frontal face views prior to detection, the system enhanced the performance of the Multi-task Cascaded Convolutional Networks. This frontalization step significantly improved detection accuracy and reduced the false positive rate, especially for profiles and oblique angles. These results suggest that the proposed hybrid model is well suited face detection in multi-view scenarios involving significant pose deviations.

Answer RQ3: Our hybrid approach exhibits considerable accuracy and strength improvements for face detection operations, especially in extreme poses, compared to renowned models like MTCNN, MTCNN++, DDFD, and Retina Face. Utilization of CAPG-GAN helps in high-quality front view generation of faces and greatly improves the capability of the MTCNN detector in detecting and identifying non-frontal faces by frontalization from diverse viewing angles up to 90°.

Comparison experiment is carried out with varying pose angles (15° to 90°), the performance of which was quantified using Intersection over Union (IoU) and Recall. New model achieves mean IoU of 0.949 and mean Recall of 0.976, while baseline MTCNN++ achieves 0.725 IoU and 0.860 Recall. Notably, the CAPG-GAN–MTCNN model works excellently even at the largest angle (90°), where conventional models typically falter.

These results show that not only is the hybrid approach more accurate but also better at dealing with large pose changes. The lower false positive rate and stable detection performance at all orientations render the proposed model highly appropriate for real-world face detection, especially in conditions where pose changes are the rule rather than the exception.

CHAPTER SEVEN

7. CONCLUSION AND FUTURE WORK

7.1. Conclusion

Recent facial detection model and generative model advancements have significantly improved the performance of pose-invariant facial analysis systems. A hybrid system combining CAPG-GAN and MTCNN put forward in research to effect better multi-view face detection, particularly under difficult non-frontal pose conditions. Accurate face detection under large yaw angles (e.g., 60°, 75°, and 90°) is also a fundamental challenge in previous detection systems. In order to overcome this, we incorporated an additional generative frontalization using CAPG-GAN prior to face detection using MTCNN effectively enhancing detection performance.

The generator network was constructed with a decoder that upsamples progressively through transposed convolutional layers with ReLU and Tanh activation functions. The use of decoder mechanisms in the downsampling pathway enabled the model to learn global context and long-range dependencies, thereby enhancing structural consistency in the generated frontal views. To maintain realism and pose fidelity in the generated outputs, a Couple-Agent Discriminator was used. This module synthesizes images through pose embeddings and provides dense feedback to train the generator. MTCNN detector consisting of P-Net, R-Net, and O-Net was implemented on frontalized images, where facial landmarks and bounding boxes were detected with great accuracy. Quantitative evaluation on benchmarking datasets like Multi-PIE and CAS-PEAL R1 confirmed that this hybrid model achieved better performance on metrics like Structural Similarity Index (SSIM) and human perceptual evaluation on frontalization and Intersection over Union (IoU), and Recall, PR curve on face detection especially on cases of extreme pose variations.

This research combined different loss functions adversarial, perceptual, pixel-wise L2, and multi-scale pixel losses to improve the generator network optimization process. The MTCNN module also used detection, bounding box , and landmark localization. This combined approach, through the union of generative modeling and cascaded detection, offers a new and effective approach to face detection in uncontrolled settings, generates, and detect faces even when they have varying poses.

7.2. Future Work

This study presented a comprehensive hybrid framework designed to detect faces from non-frontal images by transforming them into canonical frontal views using generative modeling techniques. While the proposed CAPG-GAN and MTCNN pipeline has shown substantial effectiveness, particularly under extreme pose variations, there remain several avenues for future research and development. One promising direction involves integrating the CAPG-GAN with the MTCNN model into a unified real-time face recognition system. Such integration could enhance computational efficiency by optimizing processing pipelines and reducing inference latency, thereby enabling more responsive and scalable applications.

Another potential extension of this work is adapting the proposed framework for video-based face frontalization and detection. Addressing temporal consistency across sequential frames could significantly improve the stability and coherence of detection results in dynamic environments. This capability would be particularly advantageous for real-time applications such as surveillance systems, emotion recognition, and behavioral analysis, where continuous and accurate face representation is essential.

SPECIAL ACKNOWLEDGMENT

This research work was funded by Adama Science and Technology University under the grant number: ASTU/SM-R/1116/25

REFERENCES

Alhlffee, M. H., & Abuirbaiha, R. A. A. (2024). The effects of dropout and weight initialization on human face classification accuracy using multiple-agent generative adversarial network. *2024 7th International Conference on Information and Computer Technologies (ICICT)*, 271–276.

Ben Aissa, F., Hamdi, M., Zaied, M., & Mejdoub, M. (2024). An overview of gan-deepfakes detection: Thesis, improvement, and evaluation. *Multimedia Tools and Applications*, *83*(11), 32343–32365.

Bhangale, K., Ingle, P., Kanase, R., & Desale, D. (2022). Multi-view multi-pose ro- bust face recognition based on vggnet. *Second International Conference on Image Processing and Capsule Networks: ICIPCN 2021 2*, 414–421.

Cai, J., Meng, Z., Khan, A. S., O'Reilly, J., Li, Z., Han, S., & Tong, Y. (2021). Identity-free facial expression recognition using conditional generative adversarial network. *2021 IEEE International Conference on Image Processing (ICIP)*, 1344–1348.

Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., & Choo, J. (2018). Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8789–8797.

Dang, T.-V., & Tran, H.-L. (2023). A secured, multilevel face recognition based on head pose estimation, mtcnn and facenet. *Journal of Robotics and Control (JRC)*, *4*(4), 431–437.

Fan, J., Wang, S., Yang, P., & Yang, Y. (2020). Multi-view facial expression recognition based on multitask learning and generative adversarial network. *2020 IEEE 18th International Conference on Industrial Informatics (INDIN)*, *1*, 573–578.

Farfade, S. S., Saberian, M. J., & Li, L.-J. (2015). Multi-view face detection using deep convolutional neural networks. *Proceedings of the 5th ACM on Interna- tional Conference on Multimedia Retrieval*, 643–650.

Gao, Y., Lv, X., & Jia, H. (2015). Real-time multi-view face detection based on optical flow segmentation for guiding the robot. *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2371–2377.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems, 27*.

Gross, R., Matthews, I., Cohn, J., Kanade, T., & Baker, S. (2010). Multi-pie. *Image and vision computing*, *28*(5), 807–813.

Guo, Q., Wang, Z., Wang, C., & Cui, D. (2020). Multi-face detection algorithm suitable for video surveillance. *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, 27–33.

Karras, T. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Karras, T., Aittala, M., Laine, S., Ha¨ rk o¨ nen, E., Hellsten, J., Lehtinen, J., & Aila, T. (2021). Alias-free generative adversarial networks. *Advances in neural information processing systems*, *34*, 852–863.

Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4401–4410.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of stylegan. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8110–8119.

Khan, S. S., Sengupta, D., Ghosh, A., & Chaudhuri, A. (2024). Mtcnn++: A cnn-based face detection algorithm inspired by mtcnn. *The Visual Computer*, *40*(2), 899– 917.

Liu, S., Dong, Y., Liu, W., & Zhao, J. (2012). Multi-view face detection based on cascade classifier and skin color. *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, *1*, 56–60.

Ma, M., & Wang, J. (2018). Multi-view face detection and landmark localization based on mtcnn. *2018 Chinese Automation Congress (CAC)*, 4200–4205.

Minaee, S., Luo, P., Lin, Z., & Bowyer, K. (2021). Going deeper into face detection: A survey. *arXiv preprint arXiv:2103.14983*.

Mohammadian, R., Mahlouji, M., & Shahidinejad, A. (2020). Multi-view face detection in open environments using gabor features and neural networks. *Journal of AI and Data Mining*, *8*(4), 461–470.

Naseem, S., Rathore, S. S., Kumar, S., Gangopadhyay, S., & Jain, A. (2023). An approach to occluded face recognition based on dynamic image-to-class warping using structural similarity index. *Applied Intelligence*, *53*(23), 28501–28519. https:

//doi.org/10.1007/s10489-023-05026-0

Prasad, N., Rajpal, B., R Mangalore, K., Shastri, R., & Pradeep, N. (2021). Frontal and non-frontal face detection using deep neural networks (dnn). *International Journal of Research in Industrial Engineering*, *10*(1), 9–21.

Tian, Y., Peng, X., Zhao, L., Zhang, S., & Metaxas, D. N. (2018). Cr-gan: Learning complete representations for multi-view generation. *arXivpreprint arXiv:1806.11191*.

Tran, L., Yin, X., & Liu, X. (2017). Disentangled representation learning gan for pose-invariant face recognition. *Proceedings of the IEEE conference on com- puter vision and pattern recognition*, 1415–1424.

van Beers, F., Lindströ m, A., Okafor, E., & Wiering, M. (2019). Deep neural networks with intersection over union loss for binary image segmentation. *Proceedings of the 8th international conference on pattern recognition applications and methods*, 438–445.

Viola, P., & Jones, M. J. (2004). real-time face detection. *International journal of computer vision*, *57*, 137–154.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE transactions on image processing*, *13*(4), 600–612.

Wu, S., Kan, M., He, Z., Shan, S., & Chen, X. (2017). Funnel-structured cascade for multi-view face detection with alignment-awareness. *Neurocomputing*, *221*, 138–145.

Yang, H., Zhang, Z., & Yin, L. (2018). Identity-adaptive facial expression recognition through expression regeneration using conditional generative adversarial networks. *2018 13th IEEE International Conference on Automatic Face& Gesture Recognition (FG 2018)*, 294–301.

Zafeiriou, S., Zhang, C., & Zhang, Z. (2015). A survey on face detection in the wild: Past, present and future. *Computer Vision and Image Understanding*, *138*, 1–24.

Zhang, C., & Zhang, Z. (2014). Improving multiview face detection with multi-task deep convolutional neural networks. *IEEE Winter Conference on Applications of Computer Vision*, 1036–1041.

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*,

23(10), 1499–1503.

Zhang, L., Wang, H., & Chen, Z. (2021). A multi-task cascaded algorithm with optimized convolution neural network for face detection. *2021 Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS)*, 242–245.

Zou, H., Ak, K. E., & Kassim, A. A. (2020). Edge-gan: Edge conditioned multi-view face image generation. *2020 IEEE International Conference on Image Processing (ICIP)*, 2401–2405

Bao, Y., Zhou, P., Qi, Y., Wang, Z., & Fan, Q. (2023). Learning Pixel Perception for Identity and Illumination Consistency Face Frontalization in theWild. *IEICE Transactions on Information and Systems*, *E106.D*(5), 794–803. https://doi.org/10.1587/transinf.2022DLP0055

Dang, T. V., & Tran, H. L. (2023). A Secured, Multilevel Face Recognition based on Head Pose Estimation, MTCNN and FaceNet. *Journal of Robotics and Control (JRC)*, *4*(4), 431–437. https://doi.org/10.18196/jrc.v4i4.18780

Duchi, J. C., Bartlett, P. L., & Wainwright, M. J. (2012). Randomized smoothing for (parallel) stochastic optimization. *Proceedings of the IEEE Conference on Decision and Control*, *12*, 5442–5444. https://doi.org/10.1109/CDC.2012.6426698

Gao, W., Cao, B., Shan, S., Chen, X., Zhou, D., Zhang, X., & Zhao, D. (2007). The CAS-PEAL large-scale Chinese face database and baseline evaluations. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *38*(1), 149–161.

Gross, R., Matthews, I., Cohn, J., Kanade, T., & Baker, S. (2010). Multi-pie. *Image and Vision Computing*, *28*(5), 807–813.

S, S., & S H, S. (2020). Generating Images of Face Poses for Pose Varying Face Recognition. *International Journal of Recent Technology and Engineering (IJRTE)*, *9*(2), 351–356. https://doi.org/10.35940/ijrte.f9998.079220

Sadhya, S., & Qi, X. (2023). Complementary Attention-Based Deep Learning Detection of Fake Faces. *Proceedings - 2023 IEEE International Conference on Big Data, BigData 2023*, 3644–3649. https://doi.org/10.1109/BigData59044.2023.10386483

Santemiz, P., Spreeuwers, L. J., & Veldhuis, R. N. J. (2024). A Survey on Automatic Face Recognition Using Side-View Face Images. *IET Biometrics*, *2024*(1). https://doi.org/10.1049/2024/7886911

Zhu, J., Li, Z., Wei, J., & Ma, H. (2022). PBGN: Phased Bidirectional Generation Network in Text-to-Image Synthesis. *Neural Processing Letters*, *54*(6), 5371–5391. https://doi.org/10.1007/s11063-022-10866-x

Zhong, R., Jiang, B., Li, N., Wu, Q., & Chang, H. (2022). *A multi-view face detection and expression recognition method with improved RetinaFace*. *12331*(Icmar), 23. https://doi.org/10.1117/12.2652194

APPENDEX

Appendix A. Sample Code of CAPG-GAN_MTCNN

```
    def __init__(self):
        super(CAPG_GAN, self).__init__()
        self.encoder = self.build_encoder()
        self.pyramid_module = self.build_pyramid()
        self.context_module = self.build_context()
        self.self_attention = SelfAttention(512)  # Apply self-attention
        self.decoder = self.build_decoder()

    def build_encoder(self):
        return nn.Sequential(
            nn.Conv2d(3, 64, kernel_size=4, stride=2, padding=1),
            nn.ReLU(),
            nn.Conv2d(64, 128, kernel_size=4, stride=2, padding=1),
            nn.ReLU(),
            nn.Conv2d(128, 256, kernel_size=4, stride=2, padding=1),
            nn.ReLU(),
            nn.Conv2d(256, 512, kernel_size=4, stride=2, padding=1),
            nn.ReLU()
        )

    def build_pyramid(self):
        return nn.ModuleList([
            nn.Conv2d(512, 512, kernel_size=3, stride=1, padding=1),
            nn.ReLU(),
            nn.Conv2d(512, 512, kernel_size=3, stride=1, padding=1),
            nn.ReLU()
        ])

    def build_context(self):
        return nn.Sequential(
            nn.Conv2d(512, 512, kernel_size=1),
            nn.ReLU(),
            nn.Conv2d(512, 512, kernel_size=1),
            nn.ReLU()
        )

def build_decoder(self):
    return nn.Sequential(
        nn.ConvTranspose2d(512, 256, kernel_size=4, stride=2, padding=1),
        nn.ReLU(),
        nn.ConvTranspose2d(256, 128, kernel_size=4, stride=2, padding=1),
        nn.ReLU(),
        nn.ConvTranspose2d(128, 64, kernel_size=4, stride=2, padding=1),
        nn.ReLU(),
        nn.ConvTranspose2d(64, 3, kernel_size=4, stride=2, padding=1),
        nn.Tanh()  # Use Tanh for output to scale it between -1 and 1
    )

def forward(self, x):
    encoded = self.encoder(x)
    for layer in self.pyramid_module:
        encoded = layer(encoded) + encoded  # Residual connection
    context_features = self.context_module(encoded)

    attention_out = self.self_attention(context_features)  # Apply self-attention

    output = self.decoder(attention_out)
    return output
```

Appendix: B Sample code of Couple agent Discriminator

Couple agent Discriminator

Appendix C. **Mtcnn on Frontal Generated imag**

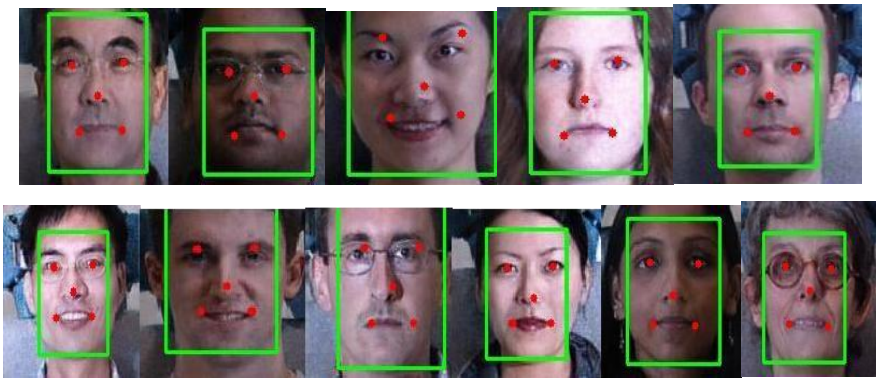
```
# --------- Discriminator ---------
Class CoupleAgentDiscriminator (nn.Module):
    def __init__(self):
        super ().__init__()
        self.agent1 = self.build_discriminator ()
        self.agent2 = self.build_discriminator ()

    def build discriminator (self):
        Return nn.Sequential (
  nn.Conv2d (3, 64, 4, 2, 1),
            nn.LeakyReLU (0.2),
            nn.Conv2d (64, 128, 4, 2, 1),
            nn.LeakyReLU (0.2),
            nn.Conv2d (128, 256, 4, 2, 1),
            nn.LeakyReLU (0.2),
            nn.Conv2d (256, 1, 4, 2, 1),
            nn.Sigmoid ()
        )

    def forward (self, x, pose_embedding=None):
        out1 = self.agent1(x)
        If pose_embedding is not none:
            # for simplicity, ignore pose in discriminator or modify as needed
            out2 = self.agent2 (pose_embedding)
            return out1, out2
        Return out1
```

Appendix: C Sample code of MTCNN

```
import os
import numpy as np
import cv2
import torch
from PIL import Image
from facenet_pytorch import MTCNN
# Set device
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
# Initialize MTCNN model (cascaded: P-Net → R-Net → O-Net)
mtcnn = MTCNN(keep_all=True, device=device)
# Define input: frontalized images from CAPG-GAN
input_dir = "/kaggle/input/capg-gan-frontalized/faces"
output_dir = "/kaggle/working/detected_faces_on_capg"

# Create output directory if not exists
os.makedirs(output_dir, exist_ok=True)

# Function to detect faces using MTCNN (P-Net, R-Net, O-Net)
def detect_faces_and_landmarks(img):
    # P-Net: Propose candidate face regions
    # R-Net: Refine candidate boxes
    # O-Net: Output final boxes + facial landmarks
    boxes, probs, landmarks = mtcnn.detect(img, landmarks=True)
    return boxes, landmarks

# Process and save results
for filename in os.listdir(input_dir):
    if filename.lower().endswith(('.jpg', '.jpeg', '.png')):
        img_path = os.path.join(input_dir, filename)
        img = Image.open(img_path).convert("RGB")
        img_np = np.array(img)
        # Detect faces and landmarks
        boxes, landmarks = detect_faces_and_landmarks(img)

        if boxes is not None:
            for i, box in enumerate(boxes):
                x1, y1, x2, y2 = map(int, box)
                cv2.rectangle(img_np, (x1, y1), (x2, y2), (0, 255, 0), 2)

                # Draw landmarks (from O-Net)
                if landmarks is not None:
                    for point in landmarks[i]:
                        cv2.circle(img_np, tuple(point.astype(int)), 2, (255, 0, 0), -1)

        # Save result with boxes/landmarks
        output_path = os.path.join(output_dir, f"detected_{filename}")
        cv2.imwrite(output_path, cv2.cvtColor(img_np, cv2.COLOR_RGB2BGR))

print("✅ Detection complete. Results saved in /kaggle/working/detected_faces_on_capg/")
```

Appendix D: Frontal Generated face image Sample Output by CAPG GAN

Appendix E: Detected Frontal Generated by MTCNN Sample Output



Appendix F: Sample Results of Human Perceptual Evaluation assessment google form

Frontalization Human perceptual Evaluation

Multi-view Face Detection Using Hybrid Generative Adversarial Networks and Multi-Task Cascaded Convolutional Neural Network

Kibru Alemu Terfasa

GT=Ground Truth
GF=Generated Frontal

Identity Preservation (15-degree pose): How well does the generated image preserve the identity of the person compared to the original?

GT GF

- 1
- 2
- 3
- 4
- 5

Appendix G: Answer For the above Question

Identity Preservation (15-degree pose): How well does the generated image preserve the identity of the person compared to the original?

GT GF

- ○ 1
- ○ 2
- ○ 3
- ○ 4
- ◉ 5

Appendix H: Evaluation Personnel List

Evaluation Personnel		
S:NO	**NAME**	**Field of study**
1	Hamba Abebe	Computer Science and Engineering(Datasecience)
2	Mekonen Bayisa	Computer Science and Engineering(Datasecience)
3	Tahir Aman	Computer Science and Engineering(AI)
4	Tilahun Tefari	Computer Science and Engineering(Networking)
5	Abdi Mosisa	Computer Science and Engineering(AI)