IPAPFA-GAN: Identity-Preserved and Attention-based Progressive Face Aging using Generative Adversarial Network

Teshale Gilisha Biru

A Thesis Submitted to the Department of Computer Science and Engineering

School 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

September 2022

Adama, Ethiopia

IPAPFA-GAN: Identity-Preserved and Attention-based Progressive Face Aging using Generative Adversarial Network

Teshale Gilisha Biru

Advisor: Dr. Worku Jifara

A Thesis Submitted to the Department of Computer Science and Engineering

School 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

September 2022

Adama, Ethiopia

Declaration

I hereby declare that this Master Thesis entitled "**Identity-Preserved and Attention-based Progressive Face Aging using Generative Adversarial Network**" is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Teshale Gilisha Biru	_____	_____
Name	Signature	Date

Recommendation

I, the advisor(s) of this thesis, hereby certify that I have read the revised version of the thesis entitled "**Identity-Preserved and Attention-based Progressive Face Aging using Generative Adversarial Network**" prepared under my/our guidance by Teshale Gilisha Biru submitted in partial fulfillment of the requirements for the degree of Master's of Science in Computer Science and Engineering. Therefore, I recommend the submission of a revised version of the thesis to the department following the applicable procedures.

Dr. Worku Jifara (Ph.D.)	_____	_____
Major Advisor	Signature	Date
_____	_____	_____
Co-Advisor	Signature	Date

Approval Page

I/we, the advisors of the thesis entitled "**Identity-Preserved and Attention-based Progressive Face Aging using Generative Adversarial Network**" and developed by Teshale Gilisha Biru, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

_____	_____	_____
Major Supervisor	Signature	Date
_____	_____	_____
Co-Supervisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by Teshale Gilisha Biru have read and evaluated the thesis entitled "**Identity-Preserved and Attention-Based Progressive Face Aging Using Generative Adversarial Network"** and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science and Engineering.

_____	_____	_____
Chair Person	Signature	Date
_____	_____	_____
Internal Examiner	Signature	Date
_____	_____	_____
External Examiner	Signature	Date

Finally, 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 School Graduate Committee (SGC).

_____	_____	_____
Department Head	Signature	Date
_____	_____	_____
School Dean	Signature	Date
_____	_____	_____
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGMENT

First and foremost, praises and thanks to the Waaqa, the Almighty, for His showers of blessings throughout my research work to complete the research successfully.

I would like to express my deep and sincere gratitude to my research advisor Dr. Worku Jifara for giving me the opportunity to do research and providing invaluable guidance throughout this research. His dynamism, vision, sincerity and motivation have deeply inspired me. He has taught me the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under his guidance. I am extremely grateful for what he has offered me. I would also like to thank him for his friendship, empathy, and great sense of humor.

Besides my advisor, I would like to thank Prof. Yun Koo Chung, Head of Computer Vision and Robotics SIG, for his persistent supervision, encouragement, and insightful comments until the completion of this thesis.

Last but not least, I am extremely grateful to my parents: my dad Gilisha Biru "**Abbaakoo nagaan na boqodhu**", my mom Sichale Diriba "**Harmeekoo sin jaalladha, na jiraadhu**", my sisters Aberash Gilisha, Tune Gilisha, Jitu Gilisha, my brothers Mulatu Gilisha, Demelash Gilisha, Roba Gilisha and close friends for their love, prayers, caring and sacrifices for educating and preparing me for my future.

Finally, my special thanks to Hachalu Hundessa "**Qaaliikoo nagaan boqodhu, si jaallanna, si yaadanna**" for his resistance against tyranny; by breaking down fear and structural barriers through rousing musical storytelling, and for uniting the people masses and amplifying their collective yearning for change.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF CODE SNIPPETS

LIST OF ACRONYMS AND ABBREVIATIONS

API	Application Programming Interface
CAAE	Conditional Adversarial Autoencoder
CACD	Cross-Age Celebrity Dataset
cGAN	Conditional Generative Adversarial Network
CNN	Convolutional Neural Network
CV	Computer Vision
GAN	Generative Adversarial Network
GB	Gigabyte
GPU	Graphics Processing Unit
I2I	Image to Image
IDE	Integrated Development Environment
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
IPAPFA	Identity-Preserved Attention Progressive Face Aging
IPCGAN	Identity Preserved Conditional Generative Adversarial Network
IS	Inception Score
KLD	Kullback Leibler Divergence
LReLU	Leaky Rectified Linear Unit
LSGAN	Least Square Generative Adversarial Network
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MTCNN	Multi-Task Cascaded Neural Network

NLP	Natural Processing Language
PAG-GAN	Pyramid Architecture of Generative Adversarial Network
PCC	Pearson Correlation Coefficient
PFA-GAN	Progressive Face Aging- Generative Adversarial Network
RAM	Random Access Memory
ReLU	Rectified Linear Unit
ResNet	Residual Network
RNN	Recurrent Neural Network
RGB	Red Green Blue
RQ	Research Question
SAGAN	Self-Attention Generative Adversarial Network
SGD	Stochastic Gradient Descent
SSIM	Structural Similarity Index Measure
TPU	Tensor Processing Unit
VCS	Verification Confidence Score
VGG	Visual Geometry Group

LIST OF NOTATIONS AND SYMBOLS

θG	Generator parameter
θD	Discriminator parameter
δ	Standard deviation
N	Number of age groups
ρ	Pearson correlation coefficient function
σ	Softmax function
λ	Binary gate
l	Cross-entropy loss
C_t	Target age condition
z	Latent space
D	Discriminator
G	Generator
L1	Manhattan distance or L1 norm
L2	Euclidean distance or L2 norm
s	Source age
t	Target age
E_x	Expected value overall real data instances
A	Age estimation network

ABSTRACT

Many studies on face aging have been conducted, ranging from approaches that use pure image-processing algorithms to those that use generative adversarial networks. It is preferable when computationally aging a face that the age output is close to the expected age and that the individual's characteristics are preserved. Conventionally, two types of modeling techniques have been used in this task: prototype-based and model-based methods. When transforming a face from a younger domain to an aged domain, both approaches fail to retain individual characteristics. With advancements in computer vision, generative models, particularly generative adversarial networks, have been used to perform this task (GANs). It is now possible to generate realistically aged faces of specific individuals using them. However, these methods cannot meet the three essential requirements of face aging simultaneously and usually generate aged faces with strong ghost artifacts when the age gap becomes large. So, identity-preserved and attention-based progressive face aging using generative adversarial network (IPAPFA-GAN) is proposed to mitigate these issues. The proposed architecture uses self-attention GAN to maintain identity-related information and improve quality of the generated images. And also, pixel-wise loss replaced with attention loss to alleviate accumulative blurriness. Further, least-square GAN employed for the discriminator to improve the quality of the synthesized images and stabilize training process. Furthermore, quantitative experiments validate the effectiveness of our approach. The inception score of the proposed solution has shown improvements in generating the desired images with better quality with the score of 34.23. The Pearson correlation coefficient of the proposed solution also has shown improvement in generating images with better aging accuracy and smoothness with the score of 0.993. Finally, the verification confidence score of the proposed solution has shown improvements in identity-preservation between input faces and generated face images with the score of 97.03, 95.12, and 90.42 respectively from three age groups. Comprehensive experiments demonstrate superior performance of the proposed solution over the existing state-of-the-art methods on CACD benchmarked dataset with attention loss, least square GAN, and self-attention mechanism.
***Keywords**: Face aging, GAN, self-attention, identity-preserved, generative models, IPAPFA-GAN.*

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

Recently in the computer vision and deep learning areas, there has been great interest in generative models for synthesizing images, for example, given human face image as an input and generating an aged human face. Facial aging analyzes a given face image to predict the future appearance of the individuals while preserving their unique identity. Face aging, also known as face age progression, is a useful tool for predicting how a person will appear under a certain age or age group with natural aging effects. It can be used for a variety of purposes, including locating a missing person, cross-age face verification, and digital entertainment (Zhu et al., 2018). It is gaining a lot of attention in today's environment, which demands more security and a touchless unique identification mechanism.

Even though the topic is extremely difficult, the area has attracted a lot of study interest. The majority of the difficulties stem from the strict requirements for the training and testing datasets, as well as the wide range of facial expression, race, gender, position, resolution, illumination, and occlusion in the face image. The dataset's strict requirement refers to the fact that most previous studies demand the availability of paired samples, i.e., face photos of the same individual at different ages, and some even require paired samples across a wide age range, which is extremely difficult to get. For example, the largest aging dataset, MORPH (Z. Zhang et al., 2017), only collected photos for each subject for 164 days. Furthermore, existing efforts necessitate labeling the query image with the correct age, which can be problematic at times. Existing studies separate training data into distinct age groups and learn a transformation across the groups; as a result, the query image must be labeled to accurately locate the image.

In recent years, generative adversarial networks (GANs) and their variant, conditional GANs (cGANs) (Huang et al., 2021), have been widely used to train a facial aging model with unpaired facial aging data, achieving better aging performance than traditional methods such as physical model-based methods and prototype-based methods (Z. Zhang et al., 2017) , alleviating the need for paired faces. The resulting approaches may be divided into two groups: cGANs-based

methods and GANs-based methods. The cGANs-based approaches (Q. Li and Y. Liu, 2020) usually train one network to find out multiple aging effects between two different age groups, with a target age group as the condition.

However, the faces generated by these technologies can't achieve three major face aging needs at once: image quality, aging accuracy, and identity preservation. The conditional adversarial autoencoder (CAAE) (Z. Zhang et al., 2017), for example, was developed to learn a face manifold, traversing which smooth age progression and regression can be achieved at the same time. Its generated faces are prone to blurriness and are unable to maintain the identity of the user. On the other hand, GAN-based approaches (H. Yang, D. Huang, Y. Wang, 2018b) aim to increase performance from a variety of angles. To keep face attributes consistent, Yang et al. constructed a pyramid-architecture discriminator to estimate high-level age-related features (H. Yang, D. Huang, Y. Wang, 2018b), and Liu et al. supplied the facial attribute vectors into both the generator and discriminator.

However, present approaches use a distinct network to understand aging effects between different age groups. That is to say, each network is trained separately. Due to the inherent complications of facial aging, such as diverse expressions and races, they are unable to guarantee a smooth aging result. The faces of the elderly are often younger than those of preceding generations. Furthermore, when the age gap between the source and target ages widens, these approaches frequently produce ghosted or blurred faces.

So, this proposed "identity-preserved and attention-based progressive face aging" has tried to contribute further improvement on the dataset preprocessing and network architecture to produce better results without requiring the paired datasets for training. Further, this study has contributed technical level improvements with the addition of different mechanisms and constraints to improve the qualities and performance of the work.

1.2. Motivation of the Study

In recent years, deep learning-based generative models have gained more and more interest due to astonishing advancements in the field of Artificial Intelligence. Deep generative models have shown an incredible ability to produce highly realistic pieces of content of various kinds, such as images, texts, and sounds depending on a huge amount of data, well-designed networks architectures, and smart training techniques.

The other thing is that nowadays, the human face is the essential biometric area (T, 2020) from a computer vision standpoint to automatically characterize the information related to different aspects of the face, including age. Because it is unique for each individual and provides a guarantee of accuracy and security. However, face aging remain challenging tasks in computer vision due to various intrinsic and extrinsic effects on the face (Osman AM, 2018). Even though an abundance of research is presented to resolve these practical issues, this issue remains one of the most challenging problems when real-life scenarios are considered. This study tries to solve this problem by extending the solutions presented in previous works explicitly for face aging.

1.3. Statement of the Problem

Generating high-quality aged face images from the input human face image is an active and challenging research problem. Most of the challenges arise from the rigid requirement to the training and testing datasets, also due to the large variation presented within the face image in terms of many intrinsic and extrinsic effects. The most recent work in existence requires the availability of supervised (paired age dataset), i.e., face images of the same person at different ages, and some even require paired samples over a wide range of ages, for example, the largest face aging dataset MORPH (Z. Zhang et al., 2017) requires an average period of 164 days for each person which is impractical and cumbersome. In addition, most existing methods fail to produce enhanced/realistic aging effects and high-quality images when the age gap becomes large (i.e., generate blurred or ghosted artifact faces) (Huang et al., 2021).

Because of different aging speed of different persons, this face aging approach aims at synthesizing a face whose target age lies in some given age group instead of synthesizing a face with a certain age. By grouping faces with target age together, the objective of face aging is equivalent to transferring aging patterns of faces within the target age group to the face whose aged face is to be synthesized. Meanwhile, the synthesized face should have the same identity with the input face. And also, most of the recent works cannot meet the three essential requirements of face aging: image quality, aging accuracy and identity preservation at the same time (Huang et al., 2021).

So, to overcome these challenging problems, identity-preserved and attention-based progressive face aging is proposed in this study.

1.4. Research Questions

In this study, an attempt was made to answer the following research questions.

RQ1: What is the effect of adding least square GAN for improving image quality in generating aged face images?

RQ2: What could be the effect of adding identity-preserving constraints to the human face age image generation to maintain personal identity?

RQ3: What is the effect of self-attention mechanism in the human face age image generation?

1.5. Objectives

1.5.1. General Objective

The general objective of this study is to design and implement identity-preserved and attention-based progressive face aging model using Generative Adversarial Network.

1.5.2. Specific Objectives

The following specific objectives were addressed to achieve the general objective.

- To identify the solutions that can preserve the identity-related information.
- To develop the architecture with the addition of improving loss functions.
- To add least-square GAN for discriminator for improving image quality.
- To design and implement the proposed architecture using GAN model.
- To integrate SAGAN attention mechanism to the proposed architecture.
- To evaluate the performance of the model with a test dataset using different metrics.

1.6. Significance of the Study

Face aging becoming a widely used method in the modern era as it serves numerous applications. For instance, in face recognition systems like banking, and in facial analysis like e-commerce platforms, face aging plays an important role (Sharma et al., 2021).

- Finding missing people/children
- Law enforcement/criminal investigation
- Digital entertainment
- Film movie effects

- Health sector

1.7. Scope and Limitations

1.7.1. Scope of the Study

The main focus of this study is to generate a realistic-looking facial image of a person given that an input facial image with an age to predict how a person will looks like at different ages. So, this study gives the following specific controls during generating an image:

- Robustness to many intrinsic and extrinsic factors
- Identity permanence
- Aging accuracy and smoothness
- Image quality

1.7.2. Limitation of the Study

This study doesn't cover the following due to time and resource constraints.

- It doesn't consider the style transfer.
- It doesn't consider image super-resolution.

1.8. Organization of the Study

This study is organized into seven chapters and they are roughly described as follows:

Chapter One: This chapter introduces the overview of the study and justifications for how and what is going to be done.

Chapter Two: This chapter discusses the background literature and related works on image-to-image translation, generating face age images, the theoretical framework of deep learning, and GANs.

Chapter Three: This chapter presents a clear description of how the research methodology is done, including the dataset collection mechanisms to the model evaluation metrics.

Chapter Four: This chapter describes the proposed model in detail. The proposed architecture, loss functions, and training pseudocode have been discussed in this section.

Chapter Five: This chapter explains the implementation of the proposed solution. It also provides the environment setup for the study and the code snippet of the proposed solution.

Chapter Six: This chapter explains the results obtained from the proposed solution and compares the results with other work.

Chapter Seven: This chapter summarizes and concludes the work along with the result and it explores the possible future works of the study.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORKS

2.1. Introduction

In detail, this chapter will explain the generation of aged faces from the input human face images progressively and different methods used to get the desired results in machine learning, results, limitations, and related works also presented in this chapter. The work of generating aged faces of a human is mostly done through learning and improving from their experience which is known as machine learning.

Machine learning is an algorithm that builds a prediction model by learning from data and then when new data is given, it will predict based on what it has learned or through experience. Nowadays, it is being widely used in different researches that provide services that we use in our daily activities like in Netflix, YouTube, Google, Facebook, and Twitter. This learning, now comes in three forms that are supervised, unsupervised, and reinforcement learning. Supervised learning uses labeled data to train the model for predicting future data. The labeled data are used for telling the machine what to look for while training. The supervised learning also tries to compare the output and the corresponding real (correct) one and tries to find the error which will help in updating the model for later on. Unsupervised learning does not have a labeled data for learning during training and it is mostly used in clustering. Here, when providing those data to the machine, it tries to identify the patterns of their performance and clusters them. Reinforcement learning is reward-based learning and works on a principle of feedback, which is when a machine is given input and generates the output according to the algorithm correctly, it will be taken as correct else if it is wrong it will give feedback for the model to predict again. In general, reinforcement learning is learned through trial and error.

Deep learning is a part of machine learning inspired by how a human brain filters information and works. Most of its methods use neural networks, often called deep neural networks. It has some basic architectures like convolutional neural networks and recurrent neural networks. LeNet-5 is the first CNN introduced in 1989 by (LeCun et al., 1989) for handwritten digit

recognition, and it was a breakthrough for many researchers for the efficient CNN models we are using now.

Now in deep learning, generative models are mainly used. Generative Adversarial Network was first introduced by Ian Goodfellow (Goodfellow et al., 2020). The GAN model architecture is built of two sub-models, the generator model which will learn to generate a fake sample, and, in its adversary, the discriminator which will try to learn to distinguish the synthesized image as real or fake. Here, both networks compete against each other to win, which will be explained in detail in the following section 2.3.

2.2. Face Aging

Face aging is a process of rendering a given face image to estimate the future appearance of an individual under a certain age or age group with realistic aging effects while still maintaining their personal identity (Liu et al., n.d.). Many studies have been conducted on face aging, ranging from the approaches that use pure image-processing algorithms to that use generative adversarial networks. In general, face aging techniques can be categorized into two types: Traditional methods and GAN-based methods.

2.2.1. Traditional Methods

Face aging has shown impressive progress in the previous two decades and different number of methods have been proposed to solve face aging problem. These early approaches were mainly focus on the skins anatomical structure and simulated the profile growth and facial muscle changes with respect to the elapsed time. These conventional approaches could be classified into two types: physical model-based methods and prototype-based methods. Physical model-based methods mechanically simulate the changes of facial appearance overtime, such as muscles and facial skins, via parameterized models whereas prototype-based methods calculate the average faces of individual in the same age group as the prototypes. Satisfied aged faces could be generated using these two types of techniques but they suffer from extremely high computational expenses and limited generalization ability due to their complex modeling techniques.

2.2.2. GAN-based Methods

Recently, with the great advancement of Generative Adversarial Networks (GANs) in image generation and translation tasks, many studies prefer GAN- based methods to address face aging problem. GAN- based methods and their variant, cGAN- based methods train face aging models using facial aging dataset with unpaired faces. They achieve better performance than traditional methods, alleviate the need for paired faces. GAN- based methods use multiple networks to learn aging effects between two different age group while cGAN- based methods use only one network to understand aging effects between different age group with a target age group as a condition. The difference between these two different types of methods conclude that cGANs-based methods are more flexible than GANs-based methods but GANs-based methods can produce better results in face aging.

2.3. Generative Adversarial Network

As explained above, it is a system or architecture where two neural networks are competing with one another to generate a variation in the data being generated. So, as its name indicates (generative) means creating fake data, (adversarial) comes from the concept where the discriminator and the generator are competing for winning which means the generator would try to fool while the discriminator would try not to get fooled. They both work simultaneously to learn and train the complex model. The discriminator D(x) in GAN is a classifier that is used to discriminate between true and generated data by the generator and it can be any network architecture that is compatible with the classifier (model). The inputs of the discriminator are the real input and the generated image and its output is a probability from [0 – 1]. The generator G(z) learns to create a fake image that could fool the discriminator to classify the output as real. The input to the generator is usually a random noise (z) and it tries to produce a meaningful output from it.

Mathematically, the Generator strives to produce fake examples x_{fake} such that (x_{fake}) is as close to 1 as possible.

Table 2.1: Generator vs Discriminator goal

Generator	$x_{fake,}$ such that (x_{fake}) is as close to 1 as possible.
Discriminator	x_{fake}, such that (x_{fake}) tries to be as close as possible to 0.

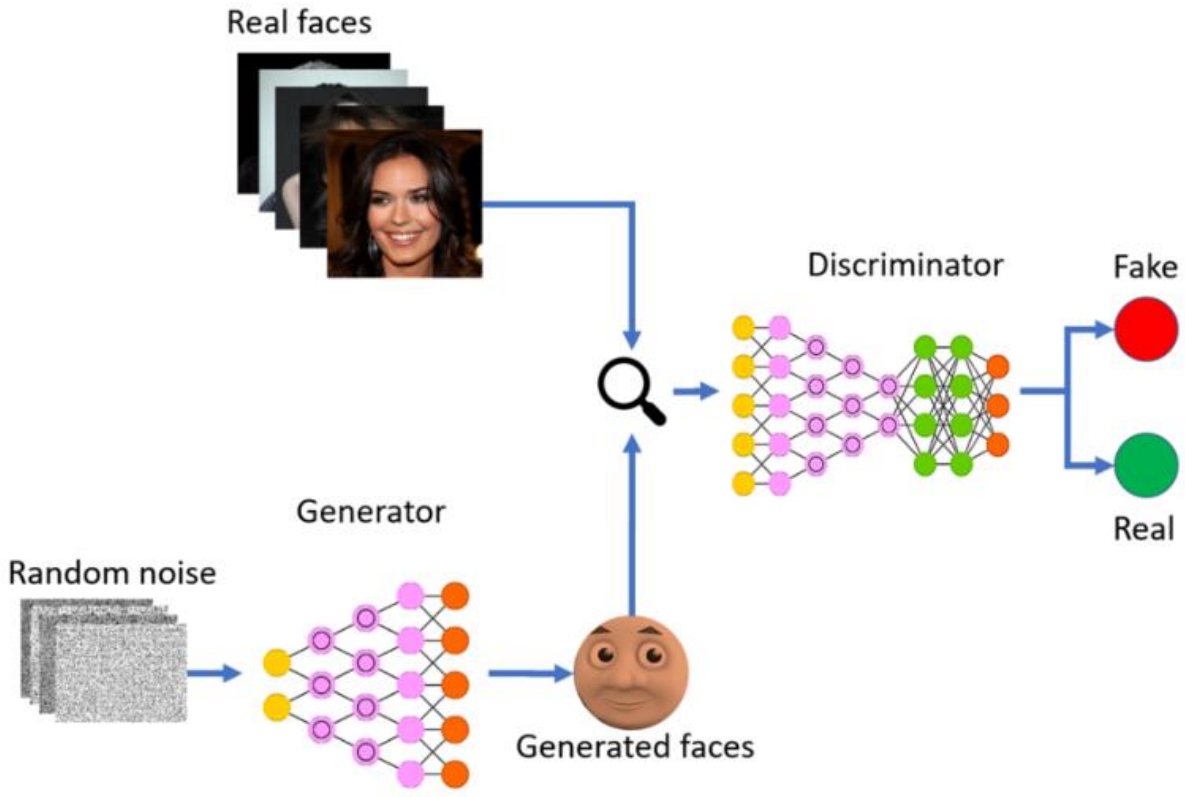


Figure 2.1: GAN Architecture

Source: Adapted from (Kulpati, 2019)

2.3.1. Internal Structure of GAN

Let's see the inside detail of the GAN. Generative Adversarial Networks are the system for training generative models in which two networks compete against each other. It has two autonomous models: Generator and Discriminator as shown in figure 2.1 above, which are denoted by differentiable functions for each network's input and parameters. The discriminator is defined by a function (x) where x (observed variable) is the input which is a real dataset. (x) gives the likelihood that x came from $pdata$ (real distribution) rather than $p(z)$ (fake distribution). It is a binary classifier with two classes, when x is real the probability is 1 and when x is synthetic the probability is 0. The discriminator can be seen as a typical CNN that transforms a 2- or 3 (grayscale or RGB) dimensional matrixes of pixels into probabilities.

The generator (z) takes input from a random noise distribution $p(z)$ where z (latent variable) is the input and generates an image as its output x_{fake}. The generated image is nourished into the discriminator network (x), which tries to classify the image as real or generated by G. The result

of the classification is backpropagated to the generator to help it learn how to produce images with a closer representation of the input data.

The loss function used in training the networks is described mathematically as (Goodfellow et al., 2020):

$$L_{adv} = min_G \max_D E_{x\epsilon X}(\log(x)) + E_{x\epsilon Z}(\log(1 - D)G(z)) \quad \text{eq. (2.1)}$$

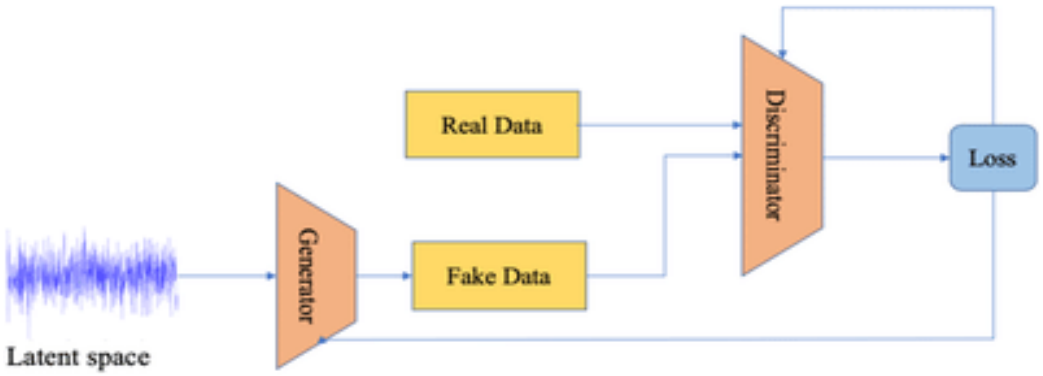


Figure 2.2: Basic structure of GAN model

Source: Cited from (G. Antipov et al., 2017)

2.3.2. Generative Adversarial Network Training

The GAN training takes place in two phases. First, freezing the generator and training the discriminator which means the generator training will be held off and there will be only a forward pass, no backward propagation. Second, freezing the discriminator and training the generator to produce a more realistic outcome that would fool the discriminator. The following is the overall steps of training Generative Adversarial Networks where the process is iterative.

1) Train the Discriminator:
a. Take a random mini-batch of real examples: x.
b. Take a mini-batch of random noise vectors z and generate a mini-batch of fake examples: $(z) = x_{fake}$.
c. Compute the classification losses for (x) and $D(x_{fake})$, and backpropagate the total error to update $\theta^{(D)}$ to minimize the classification loss.
2) Train the Generator:
a. Take a mini-batch of random noise vectors z and generate a mini-batch of fake examples: $(z) = x_{fake}$.

b. Compute the classification loss for (x_{fake}), and backpropagate the loss to update $\theta^{(G)}$ to maximize the classification loss.

2.3.3. Conditional GAN

The base GAN has already been producing a good result in synthesizing random images from a given domain. But it can be further improved by making use of class label information in the GAN model. Both the generator and discriminator models of a GAN could be taught to be conditioned on the class label. It extends the GAN model allowing the generation of images with certain attributes (conditions). This method is first introduced by (Mirza & Osindero, 2014) to direct the generation the generation of images.

The Conditional GAN (cGAN) shown in the below figure 2.3 has introduced a breakthrough for the later developments in image-to-image (I2I) translation, hand-written digits generation, and generating aged human face images.

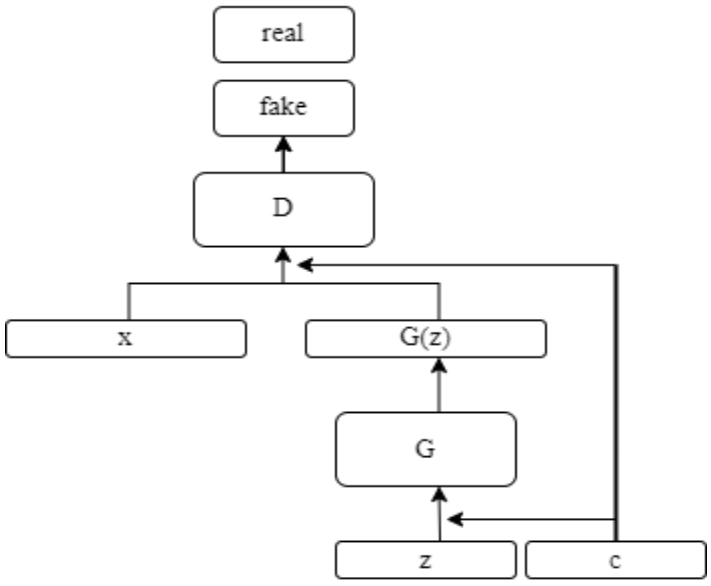


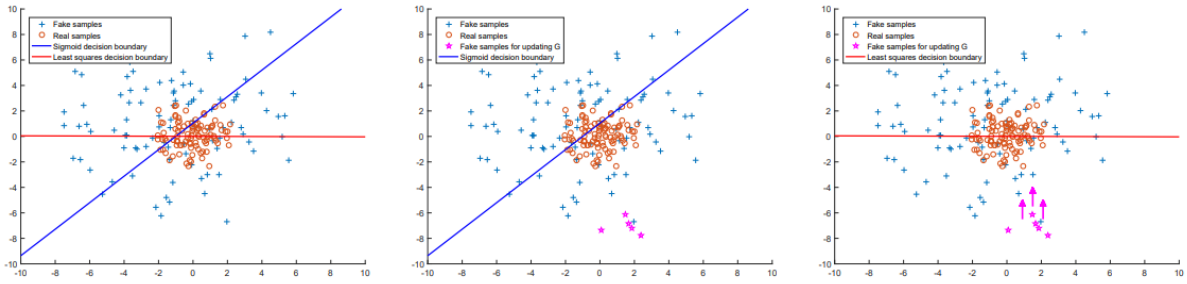
Figure 2.3: Architecture of cGAN

Source: Cited from (Mirza & Osindero, 2014)

2.3.4. Least Squares GAN

In 2016, Mau et al. (S. Zhang et al., 2020) proposed LSGAN to address the vanishing gradient problem by using least square loss rather than cross-entropy. The authors stated that the cross-

entropy loss function leads to vanishing gradient during updating generator using generated samples that are correct on decision boundary but distant from the actual data. Figure 2.4 (b) shows that when fake samples (magenta) are used to modify the generator by convincing discriminators that they are from the actual data distribution, there will be no error because they are on the correct side of the boundary. These samples, however, are far from the decision boundary and require a mechanism to bring them closer to the actual data (Mao et al., 2017). As a result, the authors propose LSGAN, a generator and discriminator loss function that uses the least-squares (LS) objective function. The LS loss function draws fake samples to the decision boundary by penalizing samples that are on the right side of the decision boundary but far from it, as shown in figure 2.4 (b). It penalizes the generated images (in magenta) and encourages the generator to generate samples close to the decision boundary.



a) b) c)

Figure 2.4: Illustration of different behaviors of two loss functions. a): Decision boundaries of two loss functions. b): Decision boundary of the sigmoid cross entropy loss function. c): Decision boundary of the least squares loss function.

Source: Adapted from (Mao et al., 2017)

2.4. Image to Image Translation

Image-to-image (I2I) translation is a class of computer vision and graphics problems where the goal is to learn the mapping between an input image and an output image using a training set of aligned image pairs. It is the task of transforming an image from one form of representation to another form of representation. It can be used for a variety of tasks, including face aging, gender swapping, object transformation, season translation, style transfer and photo enhancement (for example paintings to photographs). So, this problem was tackled in different ways using

supervised learning (with paired sets of data) or unsupervised learning where the data for training is unpaired. The majority of the currently available I2I translation is based on supervised learning that is they need paired data (Knyaz et al., 2019) and (Oza et al., 2019) for that reason, in many I2I translations scenarios, collecting paired training data is difficult and expensive. For example, (Knyaz et al., 2019) proposed a pix2pix framework, which can be used as a general solution for I2I conversion scenarios. For the training, pix2pix tries to combine the conditional GANs with the mostly used loss function called pixel-wise L1 loss between the generated image and the ground truth image. The following figure 2.5 shows the result of face aging from (Knyaz et al., 2019) paper for human face image dataset.

Input Output

Figure 2.5: Result of an image-to-image translation for face aging tasks

Source: cited from (Knyaz et al., 2019)

2.5. Attention Mechanism

Attention mechanisms have emerged from studies of human vision. A neural network is considered to be an effort to mimic human brain actions in a simplified manner. Attention mechanism is also an attempt to implement the same action of selectively concentrating on a few relevant things, while ignoring others in deep neural networks. The attention mechanism emerged as an improvement over the encoder decoder-based neural machine translation system

in natural language processing (NLP). Later, this mechanism, or its variants, was used in other applications, including computer vision (CV), speech processing, etc.

Before (Bahdanau et al., 2015) proposed the first Attention model, neural machine translation was based on encoder-decoder RNNs/LSTMs. Both encoder and decoder are stacks of LSTM/RNN units. It works in the two following steps:

- The encoder LSTM is used to process the entire input sentence and encode it into a context vector, which is the last hidden state of the LSTM/RNN.
- The decoder LSTM or RNN units produce the words in a sentence one after another

In computer vision (CV), attention mechanism also inspired great success in image-to-image translation tasks. It plays an important role in the facing since concentrating on a few relevant things of the face region. Most existing GANs-based methods usually train the model with pixel-wise loss (Z. Zhang et al., 2017) in order to keep identity consistency and background information. However, to minimize the Euclidean distance between the synthesized images and the input images can easily cause the synthesized images becoming ghosted or blurred (Knyaz et al., 2019). Specifically, this problem could be more severe if the gap between the input age and the target age becomes larger. Meanwhile the attention mechanism modifies the regions relevant to face aging, it can preserve the background information and the personal identity well without using the pixel-wise loss during training and the ghost artifacts and blurriness can be significantly reduced.

2.6. Related Work

2.6.1. Age Progression and Regression

In recent years, the study on the progression of the face age is very popular, with approaches that are classified mainly in two types, physical model-based and prototype-based. According to (Z. Zhang et al., 2017) physical model-based methods mechanically simulate the changes of facial appearance overtime, such as muscles, wrinkle, facial skins and structure etc. via a set of parameters. In order to be able to model the subtle aging mechanism better, however, a large face dataset with a large age range (for example, from 0 to 80 years) is required for each individual, which is very difficult to collect. In addition, physical modeling-based approaches are computationally expensive and do not generalize well due to the mechanical aging rules.

On the other hand, prototype-based approaches (Huang et al., 2021) often compute the average faces of people in the same age group as the prototypes. So, the difference between the prototypes of two age groups is seen as an aging pattern. However, the aging face generated from an average prototype can lose its personality (e.g., wrinkles). In order to preserve personality, (H. Yang, D. Huang, Y. Wang, 2018a) proposed a dictionary learning based method-age pattern of each age group is learned into the corresponding sub-dictionary. A given face will be decomposed into two parts: age pattern and personal pattern. The age pattern has been given to the target age pattern through the sub-dictionaries, and then the aged face is generated by synthesizing the personal pattern and the target age pattern. However, this approach presents serious ghosting artifacts.

The method based on deep learning represents the sate-of-the-art, in which the recurrent neural network (RNN) is applied to the coefficients of the eigenface for the transition of the age pattern. For instance, Wang et al. introduced a recurrent face aging framework that first maps the faces into eigenface subspace and then utilizes a recurrent neural network to model the transformation patterns across totally different ages smoothly (W. Wang et al., 2016). As one of the many supervised-based face aging methods, it needs large paired faces of the same person over a long period for training, which is impractical and cumbersome. In contrast, PFA-GAN uses several sub-networks to learn aging effects with unpaired faces.

All prototype-based approaches perform group learning, which requires the true age of the test faces to locate the transition state that may not be convenient. In addition, these methods only provide age progression from younger face to older ones. To achieve flexible bidirectional age changes, the model may need to be retrained in reverse. Face age regression, which predicts rejuvenating outcomes, is comparatively more sophisticated. Most previous work on age regression (Z. Zhang et al., 2017) are physical model-based in which the textures are simply removed based on the learned transformation on the facial surfaces. Hence, they cannot achieve photo-realistic results for baby face predictions. Next, let look at recent related work which is done through GAN.

2.6.2. Generative Adversarial Network

To solve the issues related to age progression and regression, (Huang et al., 2021) many recent works use the availability of unpaired faces to train face aging models, either using generative adversarial networks (GANs) or conditional GANs (cGANs). Most of unsupervised face aging approaches are cGANs-based (Huang et al., 2021), with a target age group as the condition. For instance, according to Zhang et al. a conditional adversarial autoencoder (CAAE) (Z. Zhang et al., 2017) achieves both facial aging and rejuvenation simultaneously by traversing on a low-dimensional face manifold. Nevertheless, projecting faces into a latent vector subspace often compromises the image quality because of the reconstruction and fails to preserve the identity-related information (Huang et al., 2021).

To overcome the above issues, Wang et al. proposed an identity preserved cGAN (IPCGAN) that uses a perceptual loss depend on a pre-trained model to preserve identity-related information (H. Yang, D. Huang, Y. Wang, 2018a). Considering that facial rejuvenation is the reverse process of facial aging, Song et al. expanded dual GANs (J. Song et al., 2018) into dual conditional GANs (Dual cGAN) to perform both face age progression and regression, and then Li et al. combined this framework with a spatial attention mechanism to better preserve identity information and reduce ghost artifacts or blurriness (Q. Li et al., 2020). (Z. Zhang et al., n.d.) proposed a multi-task learning framework, called MTLFace, to attain age-invariant face recognition (AIFR) and face age synthesis (FAS) simultaneously but their method don't more focus on the quality of synthesized images.

There are a few works directly using GANs to model the facial aging between any two age groups. For example, Yang et al. designed a discriminator with the pyramid architecture (PAG-GAN) (Yang et al., 2021), which can estimate high-level age-related details via a pre-trained neural network. In addition to preserving identity information, Liu et al. further found that the inconsistency of facial attributes still exists in previous works, and they fed the facial attribute vectors into both the generator and discriminator to suppress the unnatural changes of facial attributes (Y. Liu et al., 2019). Further, the novel approach towards face age progression and regression with template face, considered the average face for ethnicity and age (Sharma et al., 2021). The template face helped to generate the target face image for age progression and regression but fails to keep identity permanence well due to the use of average faces. (Shao et

al., 2021) proposed the paper wavelet-based multi-level GAN for face aging by combining multi-level generator and wavelet packet transform (WPT) module together to extract facial features. However, the method fail to produce the natural aging hair.

In general, cGANs-based approaches usually learn different age mappings with one or two models with a target age group as a condition, while GANs-based methods train several models for different age mappings separately. The difference between these two different types of methods arises that cGANs-based methods are more flexible than GANs-based methods but GANs-based methods can produce better results in face aging. To address the above deficiencies, identity-preserved and attention-based progressive face aging using GAN is proposed, which can enjoy best of both worlds; it is a GAN-based method but with a novel age encoding scheme, is more flexible than cGAN-based methods, and with the addition of attention loss, least squares GAN, and self-attention mechanism, achieves the best performance in image quality, identity preservation, and aging accuracy.

2.7. Summary of Related Works

The following table shows the summary of related works that have been illustrated in section 2.6.

Table 2.2: Summary of related works

Method	Objective and approach	Dataset	Age groups	Evaluation metrics	Gap
Wavelet-based multi-level GAN for face aging, (Shao et al., 2021).	Multi-level generator and wavelet packet transform (WPT) module combined together to extract facial features.	CACD, MORPH, UTKFace, FG-NET	30-, 31-40, 41-50, 51+ (CACD). 16–20, 21–25, 26–30, 31–35, 36–40, 41–45, 46–50 (MORPH).	Face verification with Face++ API. SSIM, FID, Identity verification confidence, Age estimation error. Qualitative and Quantitative comparison with prior work.	Fail to produce natural aging hair in age group 51+ of CACD i.e., there is negligence of generating gray hair for the aged group.
When Age-Invariant Face Recognition Meets Face Age Synthesis, (Z. Zhang et al., n.d.)	MTLFace proposed to learn age-invariant identity-related representation while achieving satisfied face synthesis.	LCAF, CACD-VS, CALFW, AgeDB, FG-NET	10-, 11-20, 21-30, 31-40, 41-50, 51-60, 61+.	Age Accuracy. Cosine Similarity. Qualitative and Quantitative comparison with prior work.	Only focus more on two essential requirements of face aging: Age accuracy and Identity preservation (i.e., no more focus on image quality).
CAAE, (Z. Zhang et al., 2017)	Face aging was accomplished by traversing on a low-dimensional manifold (M).	CACD, MORPH, FG-NET	0-5, 6-10, 11-15, 16-20, 21-30, 31-40, 41-50, 51-60, 61-70, 71-80	Quantitative and Qualitative analysis. Comparison to ground truth (user study). Comparison to prior work (user study).	Fails to preserve identity-related information. Prone to blurriness.

Dual cGANs, (J. Song et al., 2018)	Face aging and rejuvenation to be trained from multiple sets of unlabeled face images with different ages.	UTKFace, MORPH, CACD, FG-NET, IMDB-Wiki	0-3, 4-11, 12-17, 18-29, 30-40, 41-55, 56-65, 66-80, 81-116	Qualitative and Quantitative comparison with prior work. Comparison with ground truth.	Fails to keep identity permanence and generate ghosted or blurred aged faces.
IPcGAN, (H. Yang et al., 2018a)	For identity preservation in feature space, a pre-trained AlexNet was used.	CACD	11-20, 21-30, 31-40, 41-50, 50+	Qualitative comparison with prior work. Quantitative analysis- Inception score, age classification, face verification (user study).	Cannot handle facial aging objectives well.
Learning continuous pyramid of GANs, (Yang et al., 2021)	For aging effects, a pyramid adversarial discriminator at various scales was used, as well as an adversarial scheme for training one generator and multiple parallel discriminators.	MORPH, CACD, FG-NET	31–40, 41–50, 50+	Quantitative analysis- Face++ verification method was used. Qualitative analysis- Comparison to prior work (user study).	Inconsistency of facial attributes or unnatural changes of facial attributes.
Face age progression using exemplar face templates, (Elmahmudi	Face template generation based on the average face for a given ethnicity and age.	Manual collection of ethnicity-related datasets,	10, 20, 30, 40, 50, 60, 70, 80	Qualitative analysis - Comparison to prior work. Quantitative analysis - User study, CS (Cosine Similarity), SSIM (Structural Similarity Index) metric used, image map	Since the templates are based on the formulations of an average face for the corresponding ethnicity, gender, and for the predefined range of ages,

& Ugail, 2021)	A suitable face template was used to generate the target age.	FEI, MORPH, FG-NET		method (IMG-online).	personalized features cannot be preserved well.
Personalized face aging using aging dictionary, (Shu et al., 2015)	In a linear combination, the aging layer and a personalized layer generated automatic aging images of the face.	CACD, MORPH	0–5, 6–10, 11–15, 16–20, 21–30, 31–40, 41–50, 51–60, 61–80	Qualitative comparison - Ground truth. Quantitative comparison - User study with prior work.	This approach presents serious ghosting artifacts.

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1. Chapter Overview

The methodology employed in the implementation of identity-preserved and attention-based progressive face aging using GAN as well as the mathematical formulation of the evaluation metrics are described in this chapter. This chapter goes through the methodology, dataset, and experimental measures used to address the study questions in further detail.

The following figure 3.1 shows the overall research process workflow. The block diagram consists of different stages from identifying research area, identifying research gap and proposing solution, reviewing of literature, setting research questions and objectives, designing proposed solution, dataset preprocessing, and utilizing the datasets for training and testing the model and finally evaluating the model.

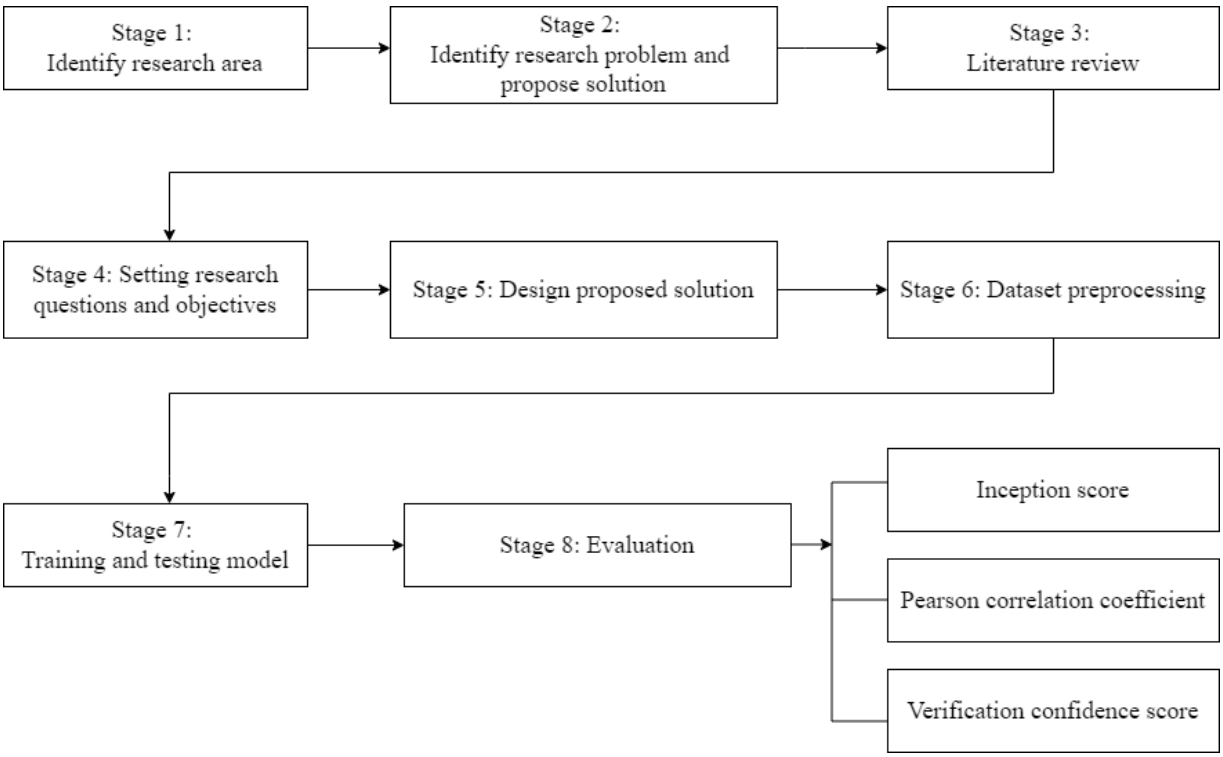


Figure 3.1: The overall research process workflow

3.2. Dataset Collection

The main goal of this research is to generate aged human face image based on a given real face image at different ages. Specifically, to conduct this study, there is a benchmarked age dataset that is publicly available: Cross-age celebrity dataset (CACD) (Huang et al., 2021).

- **CACD dataset:** contains 163,446 face images of 2,000 celebrities which were collected via Google Image Search using celebrity name and year as keywords with few controlled conditions. Therefore, it is possible to estimate the ages of the celebrities on the images by simply subtract the birth year from the year of which the photo was taken. CACD has large variations in pose, illumination, and expression than other datasets.

Figure 3.2: CACD dataset samples

CACD dataset link: https://bcsiriuschen.github.io/CARC/

Splitting strategy: The dataset is used to train and test the proposed model. The dataset should be splitted into training and testing for the proposed model. From the total 12,000 randomly selected dataset samples, 9,600 (80%) used for training and 2,400 (20%) used for testing as shown in the following figure 3.3.

Why 80:20? Based on the literature, the best percentage ratio is 80:20 (Gholamy et al., 2018). If your data is not too small it doesn't really matter whether you choose an 80:20 split or a 90:10. Indeed, we choose 80:20 because there is less training data i.e., it is less computationally intensive than 90:10.

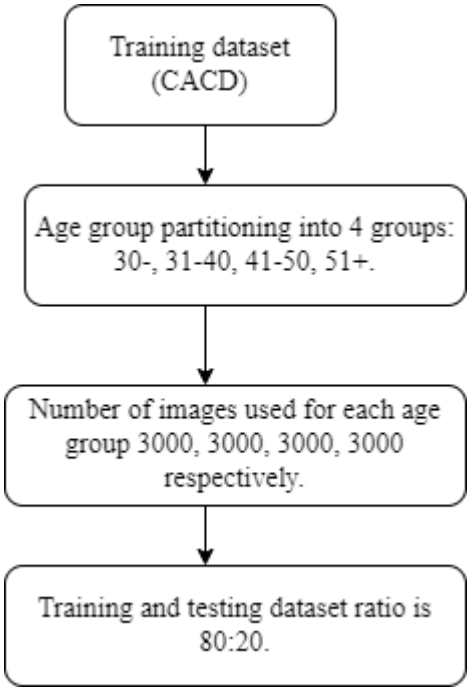


Figure 3.3: Datasets partitioning for the proposed work

To avoid data imbalance in CACD, the equal number of training images selected to create a balance between each age group evaluation.

Why CACD dataset? This dataset is selected because of three reasons. The first reason is due to our GPU's speed, this work employed this dataset, which is very easy to analyze when compared to other datasets. The second reason is that there are two important features of face aging dataset.1) large number of subjects (celebrities), and 2) large number of face images per subject captured at many different ages. Therefore, the CACD dataset has a large number of subjects or

celebrities per face images. The third reason is that this dataset has been used in many other studies, and this study used the dataset to compare it to other studies.

3.3. Data Preprocessing

Figure 3.2 above shows sample images taken from the dataset. These pictures highlight some of the variances in distance, pose, expressions, lighting, angles, and resolutions found in the data. The differences between images make it difficult for the model to find and learn patterns. Therefore, the data needs preprocessing before training the model. The preprocessing pipeline is as follows:

3.3.1. Detect and align face

The first step in the preprocessing pipeline was to find the pixel coordinates for each face in each image. This is important because to remove all background noise and only train the age estimation model to evaluate the faces in the images. To achieve this, each image was processed using the MTCNN (Multi-Task Cascaded Neural Networks) face detector (K. Zhang et al., 2016). It properly detects faces even with different sizes, lighting and strong rotations and also uses color information, since CNNs get RGB images as input. It is very accurate and robust.

3.3.2. Crop and reshape resolution (Resize image)

The original image has different height and width, for example, some image has a size of 300 x 400 so this should be converted into the same size of the image. The proposed model accepts the image in the size of 256 × 256 due for this reason the original image should be converted into a model acceptable size. Converting the image into 256 × 256 sizes is used to minimize the number of the parameters in the model. To achieve this, bounding boxes were used to resize the image so each image has its bounding box.

3.3.3. Image Normalization

It is a technique of image processing that is used to transform quite a number of pixel intensities values. The resized image was normalized into the [-1, 1] range. The formula for normalized the data into [-1, 1] is formulated as:

$$\text{output} = \frac{\text{input image} - \text{mean}}{\delta} \qquad eq. (3.1)$$

Where the δ represents the standard deviation. The value of mean and δ is set to 0.5. To calculate the normalization, the image should convert into the [0, 1] range. How? When the image passes through the transform PyTorch module, it gives an expected range value of the tensor image [0, 1] define by tensor dtype.

3.3.4. Grouping images into bins

The fourth step of the pipeline was to group images into bins, where each bin represented a distinct age-range. Different combinations were tried, with the goal of replicating typical aging periods where individuals are as similar as possible. Based on the baseline work, the result divides all ages into N non-overlapping age groups, where N = 4. More specifically, the age ranges in age group 1, 2, 3, and 4 correspond to 30−, 31 − 40, 41 − 50, and 51+, respectively.

3.3.5. Cleaning

The last step was to validate the labels of our images. Few images in CACD have wrong labels or mismatches between the face image and its annotation which makes the dataset very challenging. So, removing wrongly labelled data is an important step in our pipeline, as the performance of the resulting models are highly dependent on the quality of the data. To handle this, a tentative age estimation network model was trained on the datasets and used to filter clearly mislabeled images from the dataset.

3.4. Development Tools

For this study, different types of development tools were employed to design and implement the proposed study. These tools include prototype development tools and platforms, UML Modeling tools, and other relevant tools to the research. The following sections give a brief detail about these development tools.

3.4.1. Design Tools

Design tools are mediums used for the creation, presentation, and interpretation of design concepts. EdrawMax was used for designing purpose in the proposed system. It is a 2D business technical diagramming software that assists in the creation of flowcharts, organizational charts, mind maps, network diagrams, floor plans, workflow diagrams, business charts, and engineering diagrams.

3.4.2. Hardware Tools

The following hardware tools were used in the research implementation.

Table 3.1: Hardware Tools

No.	Tools	Used for
1.	RAM	Accelerating the training process cooperatively with GPU
2.	Hard Disk	Storing large datasets
3.	GPU	Increasing the computation and accelerating the training process

3.4.3. Software Tools

For implementing the study by coding, different writing software and coding tools were used and those are listed below with their description.

Table 3.2: Software Tools

No.	Tools	Specification
1.	Anaconda	• It is a free and open-source distribution of python and it mainly focuses on providing everything for machine learning and data science applications. • The distribution comes with the built-in Python interpreter and various packages.
2.	Jupyter Notebook	It is an open-source web application that includes coding and real-time visualization.
3.	Pytorch	• It is a Python machine learning framework based on Torch. • It's perhaps the most broadly utilized platform for deep learning research. • Why PyTorch? Because Python developers should feel more comfortable while coding with PyTorch than with other deep learning frameworks as well as easy to learn and intuitive syntax.

4.	Python	Python programming used to implement and demonstrate this thesis requires many of the drivers used to configure and package to install some device with the computer.

3.5. Baseline Works

The efficiency of this method was tested using a variety of benchmark models. The proposed model was compared against face aging models based on Generative Adversarial Networks. PFA-GAN (Huang et al., 2021) is taken as baselines for the experiments.

PFA-GAN generated high-quality images using generator (sub-generators) and a single discriminator. It achieves a better performance in aging faces of higher resolution (2×) with enhanced aging details and realistic aging effects than existing cGANs based methods. It produced impressive results in a number of image-to- image translation tasks by using PatchDiscriminator as discriminator network. In addition, to better describe the face age distribution for an improved aging accuracy, an age estimation network (A) is incorporated into the progressive face aging framework.

Why PFA-GAN as baseline model? Because this model is used to show the viability of the proposed model from the approach which employ multiple generators (sub-generators) and a single discriminator. The model optimizes sub-networks in an end-to-end training to alleviate the accumulative errors, blurriness, and sense the generated faces from previous sub-network. The purpose of this model is used to show how the proposed model has improved the quality, aging accuracy and the identity preservation issue in the face aging by adding certain constraints. The model achieved the best performance in CACD datasets. Extensively, existing experimental results demonstrated superior performance of PFA-GAN over the existing cGAN-based methods, including the state-of-the-art one, on the CACD benchmarked dataset.

3.6. Feature Extraction Network

An image encoder is used to extract the feature of the image. Image encoders used deep neural networks such as convolutional neural networks (CNN), is widely used to analyze images. Since its debut, there have been numerous varieties of CNN models, VGG-16 being one among them. The VGG-Face CNN descriptors are used as the global feature extractor based on the VGG-

Very-Deep-16 CNN architecture. VGG-Face (Chen & Haoyu, 2019) is most commonly used in face recognition models. On the ImageNet dataset, this model achieves 92.7% top-5 test accuracy. To extract the information of the produced image, this study employed the VGG-Face descriptor model. Why VGG-face model was selected? Because it is perfect for producing facial feature extraction (Cao et al., 2018).

3.7. Evaluation Metrics

Evaluation metrics are used for measuring the qualities of a model. These metrics can be objective and subjective evaluation metrics. For GANs, the objective functions of generators and discriminators often measure their performance relative to their respective opponents. For example, it may be measuring how much the generator is fooling the discriminator. Some of the objective metrics that are used for GANs are Inception Score (IS), Pearson Correlation Coefficient (PCC), Verification Confidence Score (VCS), and Fréchet Inception Distance (FID). But from these metrics, the following metrics was used to evaluate this work.

3.7.1. Inception Score (IS)

Inception score (Huang et al., 2021) is commonly used to evaluate the quality and diversity of generated images. It is a popular metric for judging the image outputs of Generative Adversarial Networks (GANs). It uses the pre-trained VGG-Face descriptor (Huang et al., 2021) to evaluate image quality of faces and extract the identity related semantic representation of a face image. A higher inception score indicates better quality of the generated face image.

$$D_{KL}(P||Q) = -\sum_{x \in X}^{n} P(x)\log\left(\frac{\mathrm{P(x)}}{\mathrm{Q(x)}}\right) \qquad \text{eq. (3.2)}$$

$$IS(G) = exp(E_{x\sim pg} D_{KL}\big(P(y|x)\big|P(y)\big) \qquad \text{eq. (3.3)}$$

Where $D_{KL}\big(P(y|x)\big|P(y)\big)$ represents the KL divergence between the conditional distribution $P(y|x)$ and the marginal distribution $P(y)$, and G the generated image. Both distributions are computed from the features extracted from the images. KL divergence (D_{KL}) is a measure of how similar/different two probability distributions are.

3.7.2. Pearson Correlation Coefficient (PCC)

It is a well-known metric to measure the aging accuracy and smoothness of different methods (Huang et al., 2021). PCC has an output value between -1 and +1, with +1 representing total positive linear correlation, 0 representing no correlation, and -1 representing total negative linear correlation. The mean PCC on all testing samples can be calculated as follows:

$$PCC = \frac{1}{m}\sum_{i=1}^{m}\rho(y_i, \bar{y}) \qquad \text{eq. (3.4)}$$

Where $\rho =$ the Pearson correlation coefficient function.

$m =$ the total number of samples.

$\bar{y} =$ the generic age sequence including the mean ages of each age group from real data, and

$y_i =$ the i-th age sequence containing N ages of the input face and the other N-1 aged faces. The ages are estimated by Face++ APIs. The higher PCC indicates the higher aging accuracy and the smoother aging result.

3.7.3. Verification Confidence Score (VCS)

Face verification experiments are performed using Face++ APIs to investigate identity preservation during face age progression. We checked whether the aged and input faces had the same identities for each input young face. Face++ is used for face comparing which means checking the likelihood that two faces belong to the same person. So, you will get a confidence score and threshold to evaluate the similarity.

Face++ link: https://www.faceplusplus.com/face-comparing/

CHAPTER FOUR

4. PROPOSED IPAPFA-GAN APPROACH

4.1. Chapter Overview

This chapter describes the proposed solution architecture with mathematical expressions and its necessary diagrammatic representation to the face aging problems. The proposed solution of generating aged human face based on a given face image at different ages is presented in this chapter. The generated image should have a better quality of visualization and preserve identity-related information of its original image. Consequently, it mainly introduces the model architecture used to generate the aged image from the human face and also the different optimizing loss functions and training scheme is described.

4.2. Proposed Approach Architecture

The proposed architecture mainly depends on the base PFA-GAN (Huang et al., 2021) and some additional loss functions like attention loss between the input face and generated one that will help in preserving the identity related information of the face and keep the identity-irrelevant information such as the background unchanged to improve the quality of the image. Here, mixed identity consistency loss adopted, which includes attention loss, structural similarity (SSIM) loss, and feature-level loss for identity preservation purpose. A feature-level loss is used to keep identity-consistency in a high-level feature space. But this loss alone could lead to unexpected changes to the identity-irrelevant information. So, the image reconstruction loss, i.e., mean absolute error (MAE) is introduced between the input and output images to produce the sparse aging effects and make the background unchanged. It may produce some outliers in the resultant aging effects. So, SSIM loss is leveraged to balance the identity-related information for feature-level loss and identity-irrelevant information for MAE.

In addition, the base PFA-GAN has introduced the age estimation loss to ensure that the generated images satisfy the target age condition apart from being photo-realistic because some synthesizes images may not satisfies the target age condition while face aging distribution. So, to overcome this problem an age estimation network included in the progressive face aging

framework to regularize the face age distribution by minimalizing age estimation loss including age group classification loss and age regression loss.

Using a standard GANs discriminator which is optimal usually results in making the generator training hard and unstable. Mostly these discriminators attempt to squeeze the output into two ends that are either 0 or 1. The GAN training is a competing game between a generator and a discriminator, in which discriminator aims to discriminate fake images from real ones, and generator tries to fool discriminator by generating more realistic fake images. Once reaching a balance, generator can produce faithful images indistinguishable from the real ones. But conventional GANs suffer from maintaining a healthy competition between generator and discriminator, leading to an unsatisfied performance. So, to improve the quality of the generated images and stabilizing the training process, least-squares GANs employed for the discriminator in this work. More specifically, least-squares loss function adopted by least-squares GANs to force generator to produce samples toward the decision boundary rather than the negative log-likelihood loss function used in the conventional GANs.

Further, the architecture of PFA-GAN is improved by introducing an attention mechanism, which will help the network in capturing the fine details from the different parts of the image and this is explained in section 4.2.7. Also, the feature extraction network used for extracting the identity-related semantic representation of a face image is described in detail the following section.

4.2.1. Feature Extractor

For the extraction of personal identity-related information of the generated face image, VGG-face descriptor was used. It is used as the global feature extractor that encodes a grayscale portrait into a vector representation which later can be used to train the system to have overall coloring effects on the grayscale portrait using fusion technique. The Feature extraction has CNN-based architecture, VGG-face 16-layer architecture. The process starts by inputting the face image. The feature extractor preprocessor is the method utilized for transfer learning purpose. So, an optimization model of transfer learning with a pre-trained model architecture, the VGG-face model, used to get the information about the input face image. Mainly, the features extracted from an image can be numeric vectors. So, the model will use this vector to describe specific features in an image.

In VGG-face model, all the hidden layers use ReLu as its activation function. ReLu is more computationally efficient because it results in faster learning and it also decreases the likelihood of vanishing/exploding gradient problems to improve the model performance.

Figure 4.1: Feature extraction process

Source: Cited from (Astawa et al., 2021)

In figure 4.1 above, the feature descriptor takes an image and encodes interesting information into a series of numbers. It acts as a sort of numerical fingerprint that can be used to differentiate one feature from another.

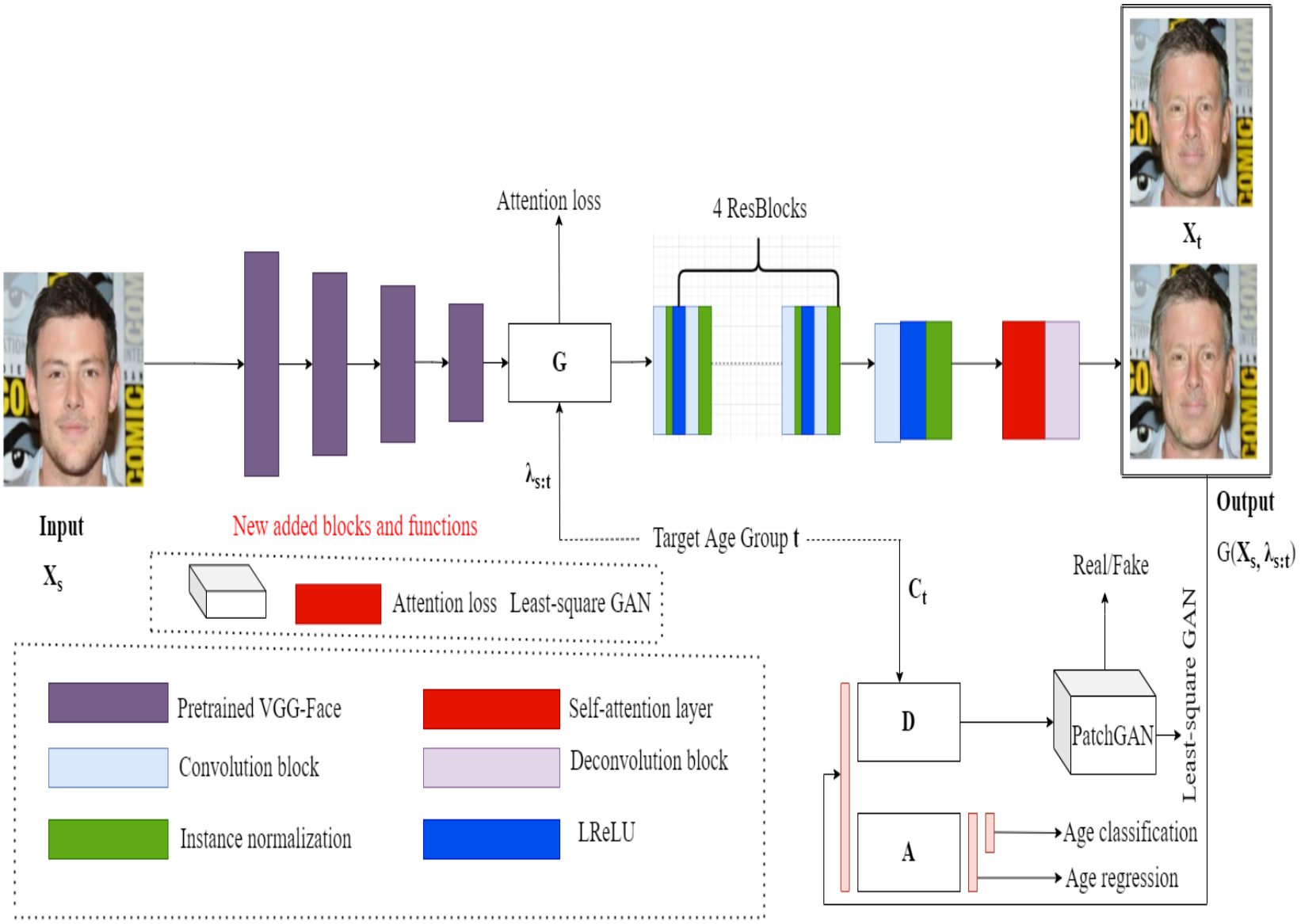


Figure 4.2: Proposed IPAPFA-GAN Architecture

Figure 4.2 above shows the GAN framework for IPAPFA-GAN. The generator G is used to age a young face $\boldsymbol{X}_s$ into an aged face $G(\boldsymbol{X}_{s,}\ \lambda_{s:t})$ that the discriminator D cannot tell apart from the real face $\boldsymbol{X}_t$. The $\boldsymbol{\lambda}_{s:t}$ is the binary gate vector for the generator to achieve progressive face aging from age group s to t, while $\boldsymbol{C}_t$ is the age condition for the discriminator to align target age with the generated images. For improved age accuracy and smoothness, the pre-trained age estimation network A is used to compute the age estimation loss, which includes an age group classification accuracy and an age regression error. PatchGAN penalizes the discriminator at the scale of patches if each N-N patch in an image is real or fake.

4.2.2. Progressive Face Aging Framework

The cGANs-based approaches use one-hot encoding to encode the target age group t as a vector $c_t \in R^{1\times N}$, whose elements are all 0s except for a single 1 to indicate the target age group. Even if this encoding scheme is convenient to some extent, it uses a single network to learn various aging effects between any two age groups, failing to generate high-quality faces when the age gap becomes large. To solve these problems, a novel progressive aging framework presented based on the fact that faces gradually age over time. The aging process formulated into a progressive neural network made up of several sub-networks, each of which only learns specific aging effects between two adjacent age groups.

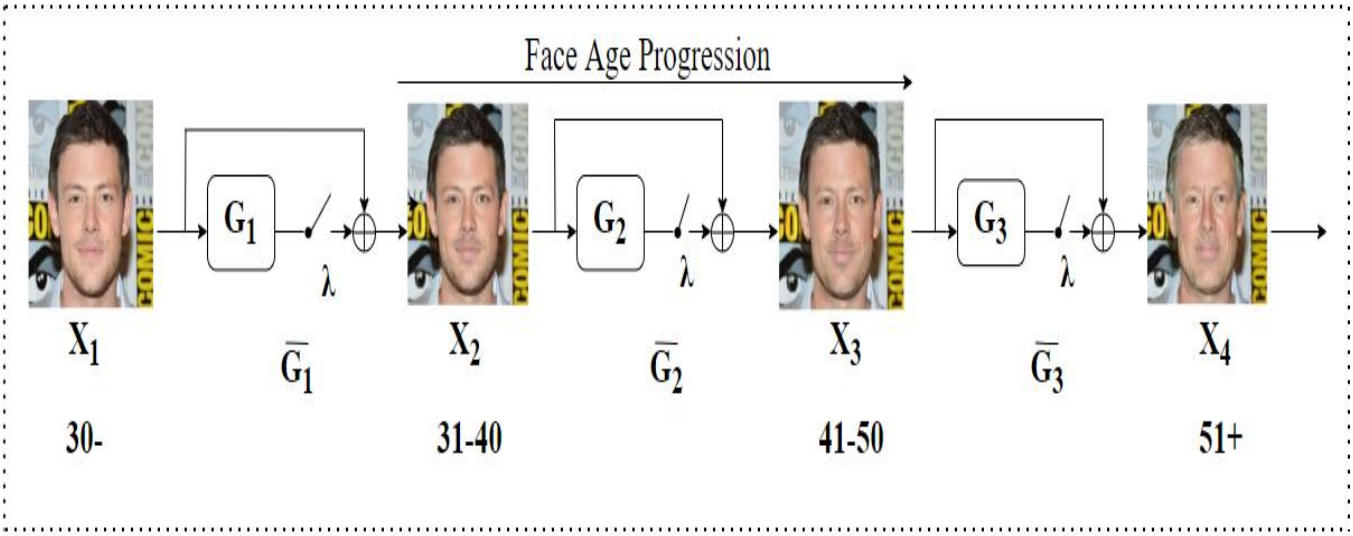


Figure 4.3: Progressive face aging framework

Source: Adopted from (Huang et al., 2021)

Figure 4.3 above illustrates the progressive face aging framework for a face facing processes with 4 age groups. Each sub-network $\bar{G}_i$ aims at aging faces from age group i to i + 1; i.e., $\boldsymbol{X}_{i+1} = \bar{G}_i(\boldsymbol{X}_i)$, which consists of a residual skip connection, a binary gate λ_i, and a light sub-network G_i outputting aging effects. The residual skip connection from input to output can prevent the sub-network from memorizing the exact copy of the input face through performing identity mapping from input to output in each sub-network, enabling sub-network to learn the aging effects. The binary gate λ_i can control the aging flow and determine if the aging mapping needs the sub-network G_i to be involved.

Therefore, the progressive aging framework from the source age group s to target one t can be expressed mathematically as follows:

$$\boldsymbol{X}_t = \bar{G}_{t-1} \circ \bar{G}_{t-2} \circ \dots \circ \bar{G}_s(\boldsymbol{X}_s), \qquad \text{eq. (4.1)}$$

where ○ denotes the function composition.

Via introducing skip connection, it is possible to recast the target age group into a sequence of binary gates that control the aging flow. The changes from age group i to i + 1 can be reformulated as:

$$\boldsymbol{X}_{i+1} = \bar{G}_i(\boldsymbol{X}_i) = \boldsymbol{X}_i + \lambda_i G_i(\boldsymbol{X}_i) \qquad \text{eq. (4.2)}$$

Here, $\lambda_i \in \{0, 1\}$ is the binary gate controlling if the sub-network G_i is involved in the path to target age group. This means, $\lambda_i = 1$ if the sub-network G_i is between source age group s and target age group t, i.e., $s \leq i < t$; otherwise $\lambda_i = 0$.

As depicted in figure 4.3, within the framework, age progression, for example from age group 1 to 4 can be expressed as:

$$\boldsymbol{X}_4 = \underbrace{\boldsymbol{X}_3 + \lambda_3 G_3(\boldsymbol{X}_3)}_{\text{Aging effects: } 3 \rightarrow 4} \qquad \text{eq. (4.3)}$$

$$= \underbrace{\boldsymbol{X}_2 + \lambda_2 G_2(\boldsymbol{X}_2) + \lambda_3 G_3(\boldsymbol{X}_3)}_{\text{Aging effects: } 2 \rightarrow 4}$$

$$= \underbrace{\boldsymbol{X}_1 + \lambda_1 G_1(\boldsymbol{X}_1) + \lambda_2 G_2(\boldsymbol{X}_2) + \lambda_3 G_3(\boldsymbol{X}_3)}$$

Aging effects: 1→4

When predicting the aged face from age group 2 to 3, the above equation reduces to $\boldsymbol{X}_3 = \boldsymbol{X}_2 + G_2(\boldsymbol{X}_2)$ because the gate vector for this aging mapping is (0, 1, 0), bypassing sub-networks G_1 and G_3. With this novel age encoding scheme, it can be shown that the framework is quite flexible in modeling the age progression between any two different age groups.

Finally, given a young face $\boldsymbol{X}_s$ of source age group s, the aging process of $\boldsymbol{X}_s$ from s into an old age group t could be expressed as follows:

$$\boldsymbol{X}_t = G(\boldsymbol{X}_{s,}\ \lambda_{s:t}), \qquad \text{eq. (4.4)}$$

where $G = \bar{G}_{N-1} \circ \bar{G}_{N-2} \circ \ldots \circ \bar{G}_1$ represents the whole progressive face aging network, and $\lambda_{s:t}$ controls the aging process, where subscript $s{:}\,t$ indicates the elements in $\lambda_{s:t}$ are all 0s except for the indices from s to t − 1 that are 1s.

4.2.3. Generator Architecture

The generator is used for generating a realistic-looking image and provides such performance via training, by back-propagating and updating weights in it to get the realistic synthesized image. The proposed architecture consists of $N - 1$ sub-generators $\{\bar{G}_i\}_{i=1}^{N}$ when N given age groups as shown in the figure 4.3 above. Each sub-generator takes colorful facial images of size $256 \times 256 \times 3$ as an input. The network structure is an encoder-decoder network with residual block as a bottleneck in between to convey the information from encoder to decoder. As shown in figure 4.4 below, the sub-generator/sub-network consists of two downsamplings with 4 residual blocks in between followed by the self-attention layer and two upsamplings followed the instance normalization layer for each sub-generator. Therefore, to describe the contents of each block:

Downsampling blocks contain:

- Convolution
- Instance normalization and
- LReLU activation function

Residual blocks contain:

- Convolution
- Instance normalization and
- LReLU activation function

Upsampling blocks contain:

- Convolution
- Deconvolution (transposed convolution)
- Instance normalization and
- LReLU activation function

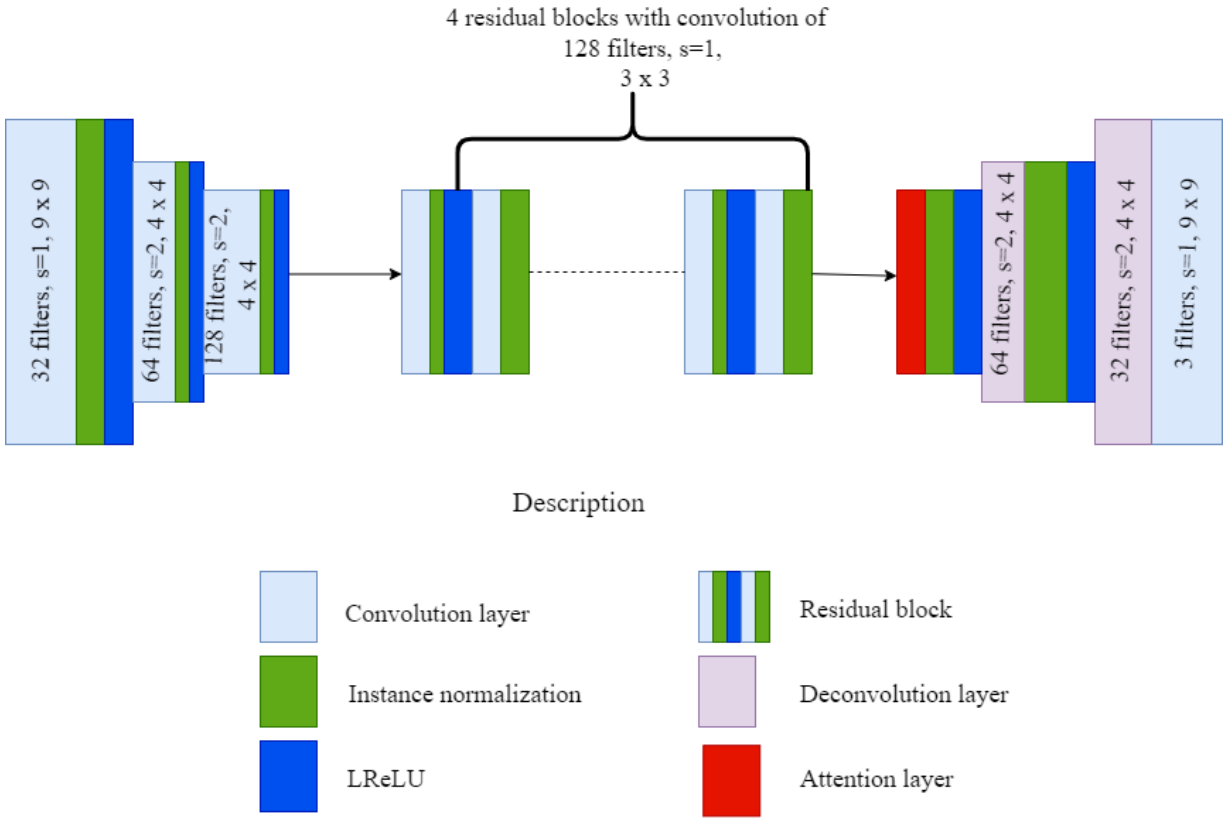


Figure 4.4: Generator architecture

4.2.4. Residual Blocks

Over the last years, many researchers are attempting to improve the performance (like reducing the error rate) and solve complex problems of neural networks by adding more layers (using deep neural networks). But it has been facing difficulties when adding more layers as it results

in problems in training and also the result of the accuracy saturates or even degrades. Therefore, to overcome this problem, ResNet introduced the concept called residual blocks with a technique called skip connections without hampering the performance. ResNet (He et al., 2020) was introduced first in 2015 and has shown great results in the ILSVRC 2015.

As depicted in figure 4.3, the skip connection from input to output enables the sub-network to learn aging effects without memorizing the exact copy of the input face. So, the backbone in the residual block was the introduction of the skip connection which changes the output of the layers and connects activations of a layer to further layers by skipping some layers in between as shown below mathematically in eq. 4.5 and eq. 4.6.

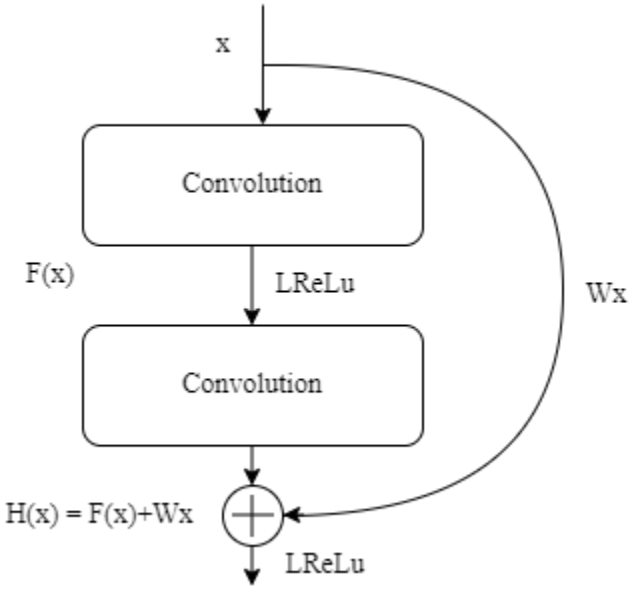


Figure 4.5: Residual block

Source: cited from (He et al., 2020)

$$H(x) = f(Wx + b) \quad \text{eq. (4.5)}$$

$$H(x) = f(Wx + b) + x \quad \text{eq. (4.6)}$$

Where $H(x) =$ output

$f(x)$ = activation function

$x =$ input vectors of the layers

$b =$ bias

$Wx =$ weight of the layer

In the above equations, we can see eq. 4.5 is after the addition of the skip connection and eq.4.6 is before the addition of the skip connection. In general, adding skip connection solve many deep learning associated problems. If any layer hurt the performance of architecture, then it will be skipped by regularization. So, this results in training a very deep neural network without the problems caused by vanishing/exploding gradient.

4.2.5. Discriminator Architecture

As the discriminator network, PatchDiscriminator is adopted in the proposed architecture. It has produced impressive results in a number of image-to-image translation tasks like face aging (Isola et al., 2017). When dealing with image generation issues, using only L2 and L1 loss does not promote high-quality image generation because it produces blurred results due to failure to capture the high frequencies. So, it is adequate to concentrate on high frequencies to generate high-quality images. PatchGAN discriminator is designed by Philip Isola et.al (Isola et al., 2017) and it penalizes areas at the scale of patches. It is a mechanism that solves this issue by capturing high frequency to generate high-quality images. PatchGAN discriminator makes the network produce high-quality results since it penalizes only structure at the local scale of patches. The discriminator architecture and the output stay the same with PatchGAN (Isola et al., 2017).

The discriminator adopted in the proposed architecture contains the following structures:

- Convolution
- Spectral normalization
- LReLU activation function

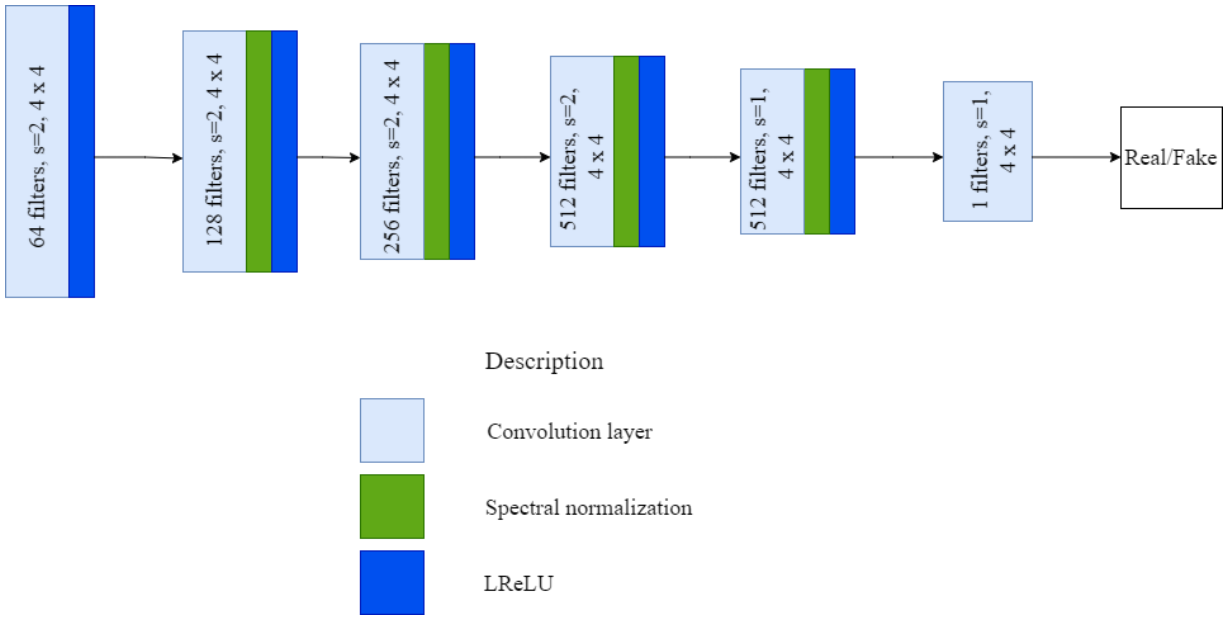


Figure 4.6: Discriminator architecture

4.2.6. Age Estimation Network

An age estimation network (A) is used for estimating the age of a person from an image. It is incorporated into the progressive face aging framework to better characterize the face age distribution for an improved aging accuracy. It has much fewer parameters than the pre-trained model adopted in (H. Yang, D. Huang, Y. Wang, 2018a). Existing works typically use either an age classifier term or an age regression term to check whether the generated face fits to the target age group, which may be inadequate to characterize the face age distribution. Therefore, to learn the age distribution in PFA-GAN, a deep expectation (DEX) (Rothe et al., 2018) term used that computes a softmax expected value for age estimation. Formally, for an input face $\boldsymbol{X}$, the estimated age $\hat{\boldsymbol{y}}$ can be calculated as follows:

$$\hat{\boldsymbol{y}} = \sum_{i=0}^{100} i \times \sigma[A(\boldsymbol{X})]_i \qquad eq.\,(4.7)$$

Where $\sigma =$ a softmax function. The number of output neurons in the age estimation network $A(\boldsymbol{X})$ was empirically set to be 101, where one neuron is expected to respond to one certain age. This should cover a wide age range from 0 to 100, and apparently includes the age ranges of the dataset used in this work. This age estimation network (A) is pre-trained on the training data

instead of training with the progressive face aging framework to frozen and regularizes the generator for an improved aging accuracy.

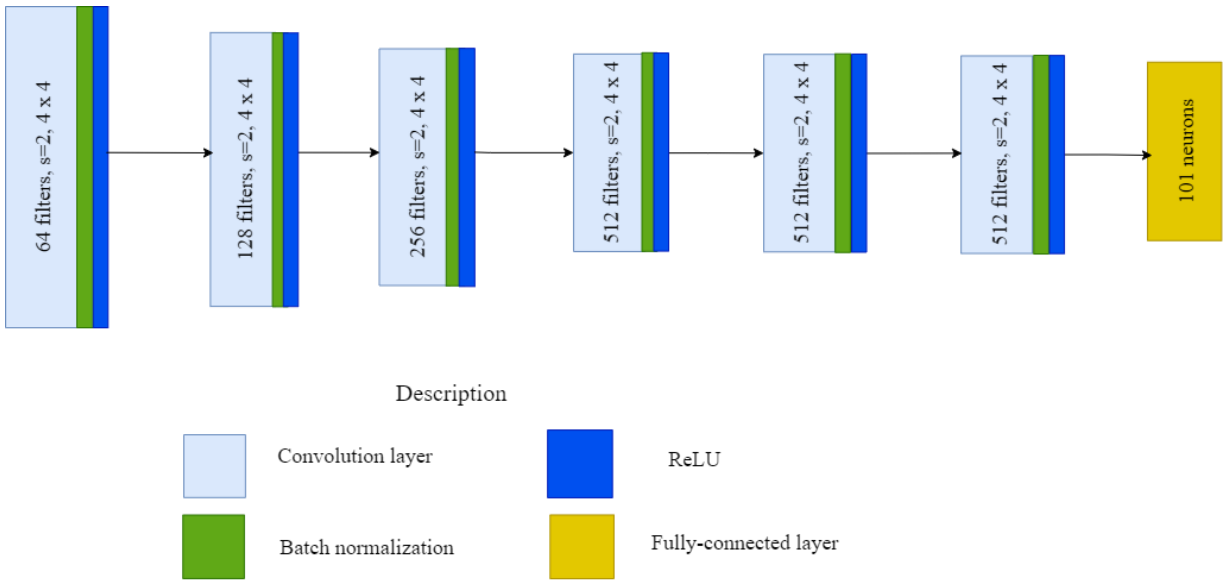


Figure 4.7: Age estimation network architecture

4.2.7 Self-Attention Mechanism

Convolutional GANs usually have much more difficulty in modeling the image distributions of multi-class datasets. They could synthesize images with few structural constraints like sky easily but fail when images are having a specific geometry or structural patterns, for example, dogs because convolution is a local operation that depends only on the size of the kernel. This means, when calculating any part of the convolution output, it has no relation with the other part except for the small local region in the size of the kernel. Therefore, to solve this problem, Self-Attention GAN (H. Zhang et al., 2019) has provided an attention mechanism that helps to keep the efficiency and long-range dependencies between the output of the computed convolution.

The working principle of Self-Attention GAN (SAGAN) is described in figure 4.8 below.

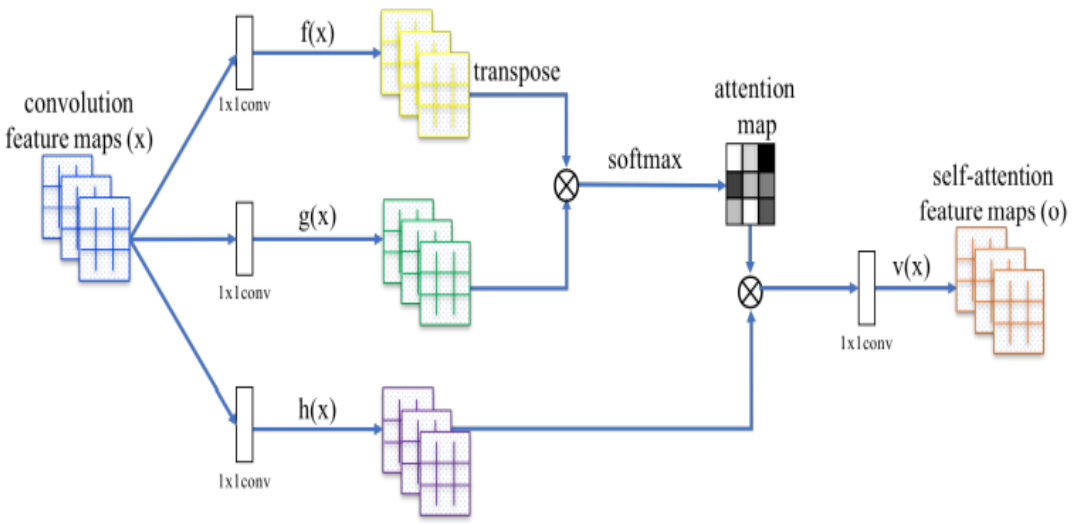


Figure 4.8: SAGAN attention mechanism

Source: Cited from (H. Zhang et al., 2019)

As depicted in figure 4.8 above, the feature map from a previous convolution will be further passed through three 1x1 convolutions (f, g, h) separately, which are used to reduce the number of channels in the image. The self-attention process starts after passing the feature map through the three convolution layers, which is the output of f is transposed and matrix multiplied with g, and softmax is taken on all rows which will produce the attention map output as shown above. Finally, the attention map output is multiplied with h to get the final self-attention feature maps. So, this layer maps helps a network in capturing the fine details from distant parts and it has produced great results of IS and FID (H. Zhang et al., 2019).

The proposed architecture's generator is made up of several sub-generators, and the output of one sub-generator is fed into the next. When one sub-generator produces ghosted faces during the aging process, the following sub-generators can detect such anomalies and force that generator to produce satisfied faces via back-propagation. In a way, the latter sub-generators provide an attention mechanism for the earlier ones to age faces effectively. Hence, the self-attention layer was used in the generator architecture of the proposed work before the upsampling block to focus on these areas of the face image relevant to face aging through only learning the residual images—aging effects as shown in figure 4.4.

Because the attention mechanism only modifies the regions relevant to face aging, the proposed method can effectively preserve background information and personal identity without utilizing pixel-wise loss, significantly reducing ghost artifacts and blurring (Zhu et al., 2020).

4.3. Model Learning Functions

It is well known that planning a viable learning function is exceptionally important for producing great results. The loss function is used to assess the gap between the generated image and the real image which is the optimization protest of the GAN. The smaller the loss function value, the smaller the gap between the generated image and the real image, and the superior the reconstruction impact. A sensible learning function can give exact gradient information for network training, consequently improve reconstruction execution/performance.

To achieve the main goal of this study different learning functions like loss function is introduced on the base PFA-GAN (Huang et al., 2021) for a better quality image generation. Hence, the proposed work is based on the general idea of it. The overall loss functions used for the proposed solution comprise three components to meet three essential requirements of face aging: Adversarial loss ($\mathcal{L}_{adv}$) aims to get better image quality, age estimation loss ($\mathcal{L}_{age}$) used to obtain a better age accuracy, and identity-preservation loss ($\mathcal{L}_{ide}$) seeks to maintain identity loss. So, the total loss used in this work is as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{adv} + \mathcal{L}_{age} + \mathcal{L}_{ide} \qquad \text{eq. (4.8)}$$

4.3.1. Adversarial loss

The GANs training is a competing game between a generator (G) and a discriminator (D). In this competition, discriminator aims to discriminate fake images from real ones and generator tries to fool discriminator by generating more realistic/faithful fake images. Once reaching a balance, generator can produce realistic images that cannot be discriminated from the real ones. But, conventional GANs suffer from preserving a well competition between generator and discriminator which leads to an unsatisfied performance. In addition, regular GANs use the sigmoid cross entropy loss function to hypothesize the discriminator as a classifier. However, during the learning process, this loss function may cause the vanishing gradients problem. To address this problem, (Mao et al., 2017) proposed least square GAN which uses the least square loss function for discriminator. So, in this proposed work least square GANs employed for the

discriminator to improve the quality of the generated images and stabilize the training process. Explicitly, least squares GANs use the least squares loss function to force the generator to generate samples toward the decision boundary rather than the negative log-likelihood loss function used in the conventional GANs.

For the given a young face $\boldsymbol{X}_s$ from age group s, the output of generator from s to an old age group t is $G(\boldsymbol{X}_{s,}\ \boldsymbol{\lambda}_{s:t})$. In the context of least squares GANs, the adversarial loss ($\mathcal{L}_{adv}$) is given as follows:

$$\mathcal{L}_{adv} = \frac{1}{2} E_{\boldsymbol{X}_s}\left[D\left(\left[G\left(\boldsymbol{X}_{s,}\ \boldsymbol{\lambda}_{s:t}\right); \boldsymbol{C}_t\right]\right) - 1\right]^2 \qquad \text{eq. (4.9)}$$

4.3.2. Age estimation loss

The generated face images are expected to fulfill the target age condition. An age estimation network (A) is integrated into the progressive face aging framework to regularize the face age distribution by reducing age estimation loss. The age estimation network is pretrained on the training data and frozen in the framework to regularize the generator towards more accurate age estimation. The age estimation loss ($\mathcal{L}_{age}$) between estimated age ($\widehat{\boldsymbol{y}}$) and target age ($\boldsymbol{y}$) for the generator (G) is computed as follows:

$$\mathcal{L}_{age} = E_{\boldsymbol{X}_s}\|\boldsymbol{y} - \widehat{\boldsymbol{y}}\|_2 + \ l(A(\boldsymbol{X})\boldsymbol{W}, \boldsymbol{c}_t] \qquad \text{eq. (4.10)}$$

Where $\boldsymbol{W} \in \mathbb{R}^{\boldsymbol{101\times N}}$ represents the final fully-connected layer for age group classification task producing the age group logits and l is the cross-entropy loss for age group classification. And also, the eq. 4.10 is the loss function to pretrain age estimation network (A) and $\boldsymbol{W}$.

4.3.3. Identity-preservation loss

In this work, mixed identity-preservation loss is used between the input face and the produced one to keep the identity-related related information of the face and preserve the identity-irrelevant information like the background unchanged. The mixed identity-preservation loss includes:

Attention loss: Most existing works including our baseline paper, use a pixel-wise loss to maintain identity consistency and keep background information, but it generates blurry face images (Zhu et al., 2020). So, to solve this problem, attention loss is used instead of pixel-wise

loss to keep the background information and preserve the personal identity, significantly reducing the ghost artifacts and blurring. It is calculated as follows:

$$\mathcal{L}_{att} = \lambda E_{\boldsymbol{X}_s}[G(\boldsymbol{X}_{s,}\ \boldsymbol{\lambda}_{s:t}) - \boldsymbol{X}_{\mathrm{s}}]^2 + E_{\boldsymbol{X}_s}\left\|G(\boldsymbol{X}_{s,}\ \boldsymbol{\lambda}_{s:t}) - \boldsymbol{X}_{\mathrm{s}}\right\|_2 \qquad \text{eq. (4.11)}$$

Feature loss: used to keep identity consistency in a high-level feature space. It is computed as follows:

$$\mathcal{L}_{fea} = E_{\boldsymbol{X}_s}\left\|\phi\left(G(\boldsymbol{X}_{s,}\ \boldsymbol{\lambda}_{s:t})\right) - \phi(\boldsymbol{X}_{\mathrm{s}})\right\|_F^2 \qquad \text{eq. (4.12)}$$

Where $\|.\|_{\mathrm{F}}$ denotes the Frobenius norm and ϕ represents the activation output of the 10th convolutional layer of the VGG-Face descriptor. In this loss, to alleviate unexpected changes to the identity-irrelevant information, image reconstruction loss that means mean absolute error (MAE) is used so that the sparse aging effects are produced and the background is remains unchanged.

Similarity (SSIM) loss: used to balance the identity-related information for feature-level loss and identity-irrelevant information for MAE. It is defined as follows:

$$\mathcal{L}_{ssim} = E_{\boldsymbol{X}_s}\left[1 - SSIM\left(G(\boldsymbol{X}_{s,}\ \boldsymbol{\lambda}_{s:t}), \boldsymbol{X}_{\mathrm{s}}\right)\right] \qquad \text{eq. (4.13)}$$

Lastly, the identity-preservation loss for generator (G) is given as follows:

$$\mathcal{L}_{ide} = \alpha_{att}\mathcal{L}_{att} + \alpha_{ssim}\mathcal{L}_{ssim} + \alpha_{fea}\mathcal{L}_{fea} \qquad \text{eq. (4.14)}$$

Where $\alpha_{att}, \alpha_{ssim}$ and α_{fea} are the hyperparameters to regulate the balance between the three losses.

4.3.4. Final loss

In order to produce a photo-realistic face that belongs to the target age group and has similar identity as the input image, the final loos function to optimize generator (G) is computed as follows:

$$\mathcal{L}_G = \lambda_{adv}\mathcal{L}_{adv} + \lambda_{age}\mathcal{L}_{age} + \lambda_{ide}\mathcal{L}_{ide} \qquad \text{eq. (4.15)}$$

The discriminator (D) is optimized by minimizing the following loss:

$$\mathcal{L}_D = \frac{1}{2}E_{\boldsymbol{X}}[D([\boldsymbol{X}; \boldsymbol{C}]) - 1]^2 + \frac{1}{2}E_{\boldsymbol{X}_s}\left[D\left(\left[G(\boldsymbol{X}_{s,}\ \boldsymbol{\lambda}_{s:t}); \boldsymbol{C}_t\right]\right)\right]^2 \qquad \text{eq. (4.16)}$$

Where the first one is calculated over all real faces from all age groups and the second one over all produced/generated fake faces from all target age groups.

4.4. Training Pseudocode

This section describes the training algorithms used in this study. The whole training of this work is optimized through training the pseudocode given in the algorithm below. Therefore, at each iteration, the input face image is selected from the corresponding dataset randomly. The procedure in algorithm 1 was used to train the identity-preserved and attention-based face aging using the GAN model in the same manner with which most GANs are trained. The entire configuration and training process is aimed at reducing the difference between the ground truth and the synthesized face image. The bold italicized indicates the added learning functions.

Algorithm 1: The overall training algorithm of the proposed model.

Input: real image ($\boldsymbol{X_s}$).

Output: synthesized face images ($G\left(\boldsymbol{X_s}, \boldsymbol{\lambda}_{s:t}\right)$).

1. Preprocessing the dataset.
2. Split the dataset into training and testing file (80 (train):20 (test) ratio).
3. Train discriminator (D):
   3.1. Concatenate C_t with the feature maps for aligning conditions with synthesized images.
   3.2. Calculate discriminator least-square loss from the synthesized images.
   3.3. Minimize classification loss by updating discriminator parameters.
4. Train generator (G):
   4.1. Use skip connection from input to output to learn aging effects.
   4.2. Add self-attention to the proposed model for generator.
   4.3 Adversarial loss is computed in the context of least-square GAN.
   4.4. Attention loss is calculated from the input face and generated ones.
   4.5. Feature loss is calculated from the input face and generated ones.
   4.6. SSIM loss is computed from the input face and generated ones.
   4.7. Identity-preservation loss is calculated from the combination of attention, feature, and SSIM loss.
   4.8. Minimize reconstruction loss by updating generator parameters based on feature and SSIM loss.

5. Train age estimation network (A):

5.1: Integrate A into progressive face aging (PFA) framework.

5.2. Age estimation loss is calculated between estimated age and target age for generator.

6. Finally, get an identity-preserved face aged image.

CHAPTER FIVE

5. IMPLEMENTATION OF THE PROPOSED SOLUTION

5.1. Chapter Overview

In this chapter, the implementation tasks of the proposed solution are comprehensively described, including the environment setup, data preprocessing, and training. In addition, some code sample snippets for the implementation are explained in this section.

5.2. Working Environments

This section describes the environment settings and their specifications for the hardware used in implementing the model.

- Desktop:
  - Operating System: Windows 10
  - Processor: Intel® Core™ i7-8700 CPU @ 3.20GHz 3.19 GHz
  - Installed memory (RAM): 8.00 GB
  - System Type: 64-bit operating system, x64-based processor
- Desktop:
  - Operating System: Ubuntu 20.04.2 LTS
  - Processor: Intel® Core™ i5-4590 CPU @ 3.30GHz x 4
  - Graphics: NVIDIA Corporation TU104 [GeForce RTX 2070 SUPER] / GeForce RTX 2070 SUPER/PCIe/SSE2
  - Installed memory (RAM): 8.00 GB
  - System Type: 64-bit

5.3. Environment Setup

For the development of this study, various kinds of software and Integrated Development Environments (IDEs) are used.

Anaconda: It is a free and open-source distribution of python and it mainly focuses on providing everything for data science and machine learning application.

Jupyter Notebook: It is an open-source web application that contains visualization and coding in real-time.

Colab: It is a cloud-based notebook that allows writing a python code on any browser with a good computing time and resources using Tensor Processing Unit or Graphical Processing Unit for training.

Sublime Text: It is a shareware cross-platform source code editor with a Python application programming interface (API) and natively supports many programming languages and markup languages (Yadav et al., n.d.). It's jam-packed with powerful features like multi-line editing, build systems for dozens of programming languages, regex find and replace, a Python API for developing plugins, and more.

5.4. Training Procedure

To accomplish the overall objectives of this study, the following processes carried out.

- Dataset collection
- Data preprocessing
- Configuring Graphical Processing Unit (GPU) and the dataset
- Implementing the work (from training to testing)

Therefore, in order to provide more information about the work that has been done, a detailed description of each stage is provided.

5.5. Learning Hyperparameters

Hyperparameters are the variables which determine the network structure (for example, number of hidden units) and how the network is trained (for example, learning rate). They are set before training i.e., before optimizing the weights and bias. Training hyperparameters need to be carefully selected for the GAN to learn the model architecture very well. In this work, Adam optimization method was used to optimize the model. It is an optimization algorithm that is used to update network weight iterations based on training data and provided to replace the classic stochastic gradient descent (SGD) process. The model was initialized by He initialization. Why Adam is selected? Because it is simple to implement, requires less memory and computationally efficient. It is really efficient when working with large problem involving a lot of data or parameters (Kingma & Ba, 2015).

The other hyperparameters used is the learning rate (lr) which controls how quickly the model is learned/adapted to the problem. So, this learning rate need to be tuned to give better results in training and mostly, this tuning process could be done through trial and error. In this work, age estimation network (A) was pretrained on the training dataset for 50 epochs with an initial learning rate of 1.0×10^{-4}, a batch size of 4 and a learning rate decay factor of 0.7 after every 15 epochs. A 4 mini-batch size is selected to fit the memory requirements of the GPU, and it is faster for learning (Y. Lecun, 2018). In this work, the learning rate value is not the same for the overall training instead it decays after every 50 epochs during the training linearly from large to small for generator and discriminator. The learning rate that is used as the baseline paper implemented and the weights are added for the corresponding losses while training for adjusting their values.

Table 5.1: Learning hyperparameters

No.	**Hyperparameters**	**Value**
1.	Learning rate (lr)	0.0001
2.	Batch size	4
3.	Epoch	200
4.	Exponential decay rates for the first moment (β_1)	0.5
5.	Exponential decay rates for the second moment (β_2)	0.99
6.	Weight of identity-preserving loss	0.02
7.	Weight of adversarial loss	100
8.	Weight of age estimation loss	0.4
9.	Alpha ssim (α_{ssim})	0.15
10.	Alpha feature (α_{fea})	0.025
11.	Alpha attention (α_{att})	0.02
12.	Optimizer	Adam
13.	Initialization	He

5.6. Experiment Class

Conducting experiments and testing them are the important things to evaluate and get the desired quality image. Therefore, for this work, four classes of experiments were conducted. The first class experiment was conducted by using the baseline work (PFAGAN) which uses pretrained VGG-16 as a feature extractor. The second-class experiment (PFAGAN-AL) is the initial work of the proposed model. Pixel-wise loss was replaced by attention loss to preserve identity-related information and increase the visual quality of generated face images, significantly reducing the ghost artifacts and blurring. In the third experiment (PFAGAN-AL-LS), least-square GAN was employed to improve the quality of synthesized images and stabilize the training process. In the fourth experiment (IPAPFA-GAN), attention mechanism (Self-Attention GAN) was added to enhance the PFAGAN by keeping the background information unchanged and preserving the identity information. In the baseline work, pixel-wise loss was used to keep background information unchanged but this loss generates blurred face images. So, attention mechanism was implemented to overcome this problem. In general, all the conducted experiments are summarized in the table as follows.

Table 5.2: Experimental class lists

Notations	**Experiment**	**Dataset**	**Model**
PFAGAN	Baseline work	CACD, MORPH	PFAGAN
PFAGAN-AL	Initial work of the proposed model with attention loss	CACD	Enhanced PFAGAN
PFAGAN-AL-LS	PFAGAN-AL with least-square GAN	CACD	Enhanced PFAGAN
IPAPFA-GAN	PFAGAN-AL-LS with attention mechanism (***Full implementation***)	CACD	Enhanced PFAGAN with modification

5.7. IPAPFA-GAN Implementation

Totally, IPAPFA-GAN consists of three network architectures. They are generator, discriminator, and age estimation network. The generator fool discriminator by generating fake images that look like real images and the discriminator discriminates whether the synthesized

image is real or fake. An age estimation network is integrated into the progressive face aging framework to characterize the face age distribution better, thereby improving aging accuracy and smoothness.

As described in section 4.2.3, the input to each sub-generator is a colorful face image of size $256 \times 256 \times 3$. Its architecture consists of 11 layers, which are three convolutional layers (downsampling layers) followed by four residual networks, one self-attention layer, two deconvolutional layers and one convolutional layer (upsampling layers). Except for the last one, all the convolutional layers are followed by instance normalization and Leaky Rectified Linear Unit (LReLU) activation function with a slope of 0.2 for negative input. The discriminator architecture consists of a sequence of six convolutional layers. Each convolutional layer except the first and last ones is followed by spectral normalization and LReLU activation function whose slope is 0.2 for negative input. An age estimation network architecture has six convolutional layers and one fully-connected layer in which the last layer consists of 101 neurons corresponding to all possible age ranges from 0 to 100.

Table 5.3: Detail network architecture of IPAPFA-GAN

Sub-generator/Sub-network				
Layer	Filter/Stride	Normalization	Activation	Output Shape
Conv	9 × 9/1	Instance	LReLU	32 × 256 × 256
Conv	4 × 4/2	Instance	LReLU	64 × 128 × 128
Conv	4 × 4/2	Instance	LReLU	128 × 64 × 64
ResBlock × 4	3 × 3/1	Instance	LReLU	128 × 64 × 64
Self-attention	4 × 4/2	Instance	LReLU	128 × 64 × 64
Deconv	4 × 4/2	Instance	LReLU	64 × 128 × 128
Deconv	4 × 4/2	Instance	LReLU	32 × 256 × 256
Conv	9 × 9/1	-	-	3 × 256 × 256
Discriminator				
Layer	Filter/Stride	Normalization	Activation	Output Shape
Conv	4 × 4/2	-	LReLU	64 × 128 × 128
Conv	4 × 4/2	Spectral	LReLU	128 × 64 × 64

Conv	4 × 4/2	Spectral	LReLU	256 × 32 × 32
Conv	4 × 4/2	Spectral	LReLU	512 × 16 × 16
Conv	4 × 4/1	Spectral	LReLU	512 × 15 × 15
Conv	4 × 4/1	-	-	1 × 14 × 14
Age Estimation Network				
Layer	Filter/Stride	Normalization	Activation	Output Shape
Conv	4 × 4/2	Batch	ReLU	64 × 128 × 128
Conv	4 × 4/2	Batch	ReLU	128 × 64 × 64
Conv	4 × 4/2	Batch	ReLU	256 × 32 × 32
Conv	4 × 4/2	Batch	ReLU	512 × 16 × 16
Conv	4 × 4/2	Batch	ReLU	512 × 8 × 8
Conv	4 × 4/2	Batch	ReLU	512 × 4 × 4
Linear	N	-	-	N
Linear	101	-	-	101

5.7.1. Residual Blocks Implementation

In this work, residual skip connection from input to output is used to prevent the sub-networks from memorizing the exact copy of input face images.

```
class ResidualBlock(nn.Module):
    def __init__(self, channels, norm_layer):
        super(ResidualBlock, self).__init__()
        self.main = nn.Sequential(
            get_norm_layer(norm_layer, nn.Conv2d(channels, channels, 3, 1, 1)),
            nn.LeakyReLU(0.2, inplace=True),
            get_norm_layer(norm_layer, nn.Conv2d(channels, channels, 3, 1, 1)),
            self_attention = Self_Attention(filter = 4, stride = 2),
        )

    def forward(self, x):
        residual = x
        x = self.main(x)
        return F.leaky_relu(residual + x, 0.2, inplace=True)
```

Code Snippet 5.1: Code snippet for residual blocks

5.7.2. PatchDiscriminator Implementation

In this study, PatchDiscriminator was adopted as the discriminator for better image translation tasks.

```
class PatchDiscriminator(nn.Module):

    def __init__(self, age_group, conv_dim=64, repeat_num=3, norm_layer='bn'):
        super(PatchDiscriminator, self).__init__()

        use_bias = True
        self.age_group = age_group

        self.conv1 = nn.Conv2d(3, conv_dim, kernel_size=4, stride=2, padding=1)
        sequence = []
        nf_mult = 1

        for n in range(1, repeat_num):  # gradually increase the number of filters
            nf_mult_prev = nf_mult
            nf_mult = min(2 ** n, 8)
            sequence += [
                get_norm_layer(norm_layer, nn.Conv2d(conv_dim * nf_mult_prev +
(self.age_group if n == 1 else 0),
                                                     conv_dim * nf_mult, kernel_size=4, stride=2, padding=1,
                                                     bias=use_bias)),
                nn.LeakyReLU(0.2, True)
            ]

        nf_mult_prev = nf_mult
        nf_mult = min(2 ** repeat_num, 8)

        sequence += [
            get_norm_layer(norm_layer,
                           nn.Conv2d(conv_dim * nf_mult_prev, conv_dim * nf_mult, kernel_size=4,
                                     stride=1, padding=1,
                                     bias=use_bias)),
            nn.LeakyReLU(0.2, True),
            nn.Conv2d(conv_dim * nf_mult, 1, kernel_size=4, stride=1,
                      padding=1)  # output 1 channel prediction map
        ]
        self.main = nn.Sequential(*sequence)

    def forward(self, inputs, condition):
        x = F.leaky_relu(self.conv1(inputs), 0.2, inplace=True)
        condition = group2feature(condition, feature_size=x.size(2),
```

```
age_group=self.age_group).to(x)
        return self.main(torch.cat([x, condition], dim=1))
```

Code Snippet 5.2: Sample code for PatchDiscriminator

5.7.3. Identity-Preserving Loss Implementation

In the proposed solution, a mixed identity-preserving loss is used to maintain identity-related information and keep identity-irrelevant information such as the background unchanged.

```
##################################Feature_loss#########################
fea_loss = F.fea_loss(g_source, source_img)
##################################SSIM_loss#########################
ssim_loss = compute_ssim_loss(g_source, source_img, window_size=10)

##################################Attention_loss#####################
att_loss= compute_att_loss(g_source, source_img, window_size=10)

##################################ID_loss############################
id_loss= α_att * att_loss + α_ssim * ssim_loss + α_fea * fea_loss
```

Code Snippet 5.3: Sample code for identity-preserving loss

5.7.4. Adversarial Loss Implementation

In this study, adversarial loss is implemented with least- square GAN for discriminator to improve the quality of synthesized images and stabilize the training process.

```
##################################GAN_loss############################
gan_logit = self.discriminator(g_source, target_label)
g_loss = 0.5 * ls_gan(gan_logit, 1.)
# g_loss = ls_gan(gan_logit, 1.)
```

Code Snippet 5.4: Sample code for adversarial loss

5.7.5. Age Estimation Loss Implementation

The generated face images are expected to fulfill the target age condition. An age estimation network (A) is integrated into the progressive face aging framework to regularize the face age distribution by reducing age estimation loss for improving aging accuracy and smoothness.

```
##################################Age_loss############################
age_loss = self.age_criterion(g_source, mean_age)
```

Code Snippet 5.5: Sample code for age estimation loss

CHAPTER SIX

6. RESULT, EVALUATION, AND DISCUSSION

6.1. Chapter Overview

After discussing detailed ideas about the implementation and the experiments performed in the previous section, this chapter presents the results obtained from the experiments and gives a comparative description of the results with tables, figures, and graphs. The performance of the proposed model is evaluated using the evaluation metrics described in chapter 3. Furthermore, the proposed model is compared with existing face aging architectures.

6.2. Face Aging

Face aging takes real image as an input and synthesizes face aged images while preserving identity-related information. This study conducts a distinctive training experiment to clarify the quantitative results of comparing the baselines on CACD dataset that has been used to train and test the proposed model. After the model training, its execution was tested on its ability to generate aged face images from a given input real image. The similarity between the original and generated images was measured using the CACD test set.

As described in table 5.2, four experiments were conducted in this study. In the first experiment, baseline work (PFAGAN) is implemented. It uses VGG-16 as the feature extraction network. However, as discussed in the result section, this experiment produces some blurry images. To address this issue, PFAGAN-AL was implemented by replacing pixel-wise loss with attention loss for the generator to produce high-quality images. To improve the quality of generated image and stabilize training process, PFAGAN-AL-LS is proposed by employing least-squares GANs to the PFAGAN-AL for discriminator. It outperforms the two experiments in terms of quantitative and qualitative measurement. Finally, IPAPFA-GAN is implemented by adding attention mechanism (SAGAN) to the PFAGAN-AL-LS. It outperforms the three experiments in terms of preserving identity information and keeping background information unchanged, generating high-quality images and obtaining a better age accuracy and smoothness. All four experiments were performed using the same hardware and software, hyperparameters, and dataset configuration for comparison, as described in section 5.6.

6.3. Experimental Results

The results for the same CACD dataset for each performed experiment are shown below. The input is taken from 30- age group to generate aged face images for the other three age groups (31-40, 41-50, 51+). The ground truth age has helped in getting better aging accuracy and smoothness. The same dataset was used for all the experiments for comparison purposes.

6.3.1. PFAGAN

Figure 6.1 illustrates the result of the baseline work on the cross-age celebrity dataset.

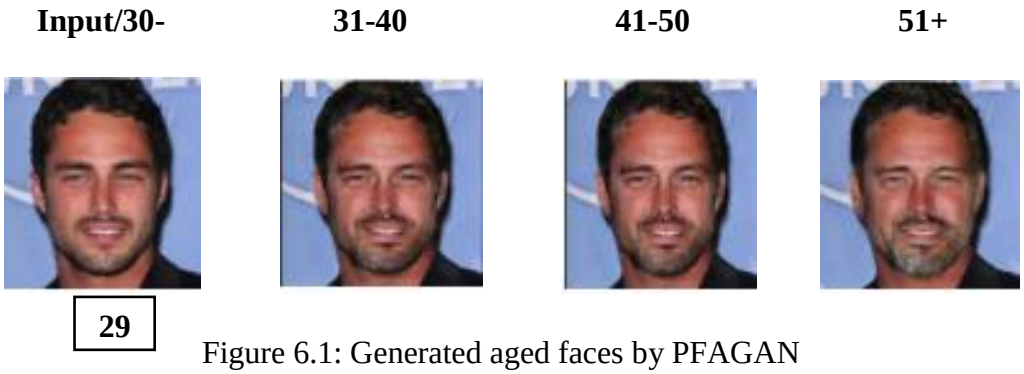


Figure 6.1: Generated aged faces by PFAGAN

6.3.2. PFAGAN-AL

Figure 6.2 shows the result of replacing pixel-wise loss with attention loss for the baseline work. It shows some improvement over the baseline work in terms of image quality and identity consistency. Pixel-wise loss preserves identity information, but produces blurry images that degrade image quality.

Figure 6.2: Generated aged faces by PFAGAN-AL

6.3.3. PFAGAN-AL-LS

The addition of LSGAN to the previous work has shown an enhancement to the image being generated. Figure 6.4 depicts the result of adding least-square GAN to the previously performed experiment described in section 6.3.2 above. As shown in the figure below, it has tried to capture the shapes, improve the quality of generated images, and stabilize the training process.

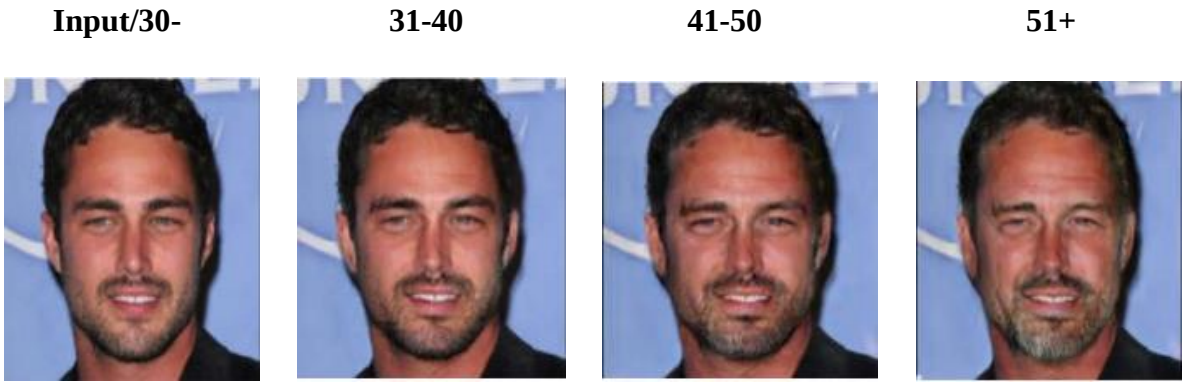


Figure 6.3: Generated aged faces by PFAGAN-AL-LS

6.3.4. IPAPFA-GAN

Finally, the proposed solution of this work (IPAPFA-GAN) constitutes constraints like mixed identity-preserving loss, adversarial loss, age estimation loss, and self-attention mechanism. A self-attention mechanism was added to PFAGAN-AL-LS which helps generator network only modifies the regions relevant to face aging. More specifically, it improved the performance of the previous for preserving identity-related information, keeping identity-irrelevant information such as background unchanged, and generating high quality images. Figure 6.5 below shows the result of IPAPFA-GAN.

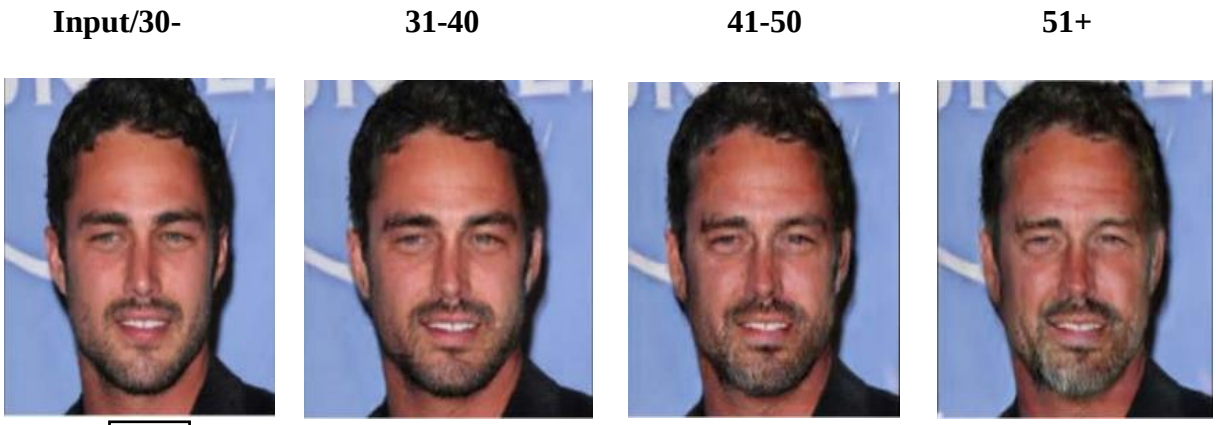


Figure 6.4: Generated aged faces by IPAPFA-GAN

Figure 6.5 below shows the input faces from the age group of 30- are aged into the age group of 51+ by the four models. The final proposed solution (IPAPFA-GAN) produces enhanced/realistic aging effects and suppresses the ghost artifacts towards a better image quality by using an attention mechanism over baseline work, PFAGAN-AL, and PFAGAN-AL-LS.

Figure 6.5: Generated aged faces by four models

Figure 6.6 below illustrates the generated aged faces under three extreme conditions (1st row- illumination, 2nd row- occlusion, and 3rd row- low quality) of face on the CACD dataset. Each input face from 30- was aged into three old age groups by IPAPFA-GAN. Previous works revisited to better clarify why our proposed solution robust under these extreme conditions. Specifically, due to various intrinsic complexities of face aging, previous works cannot produce high-quality aged faces with smooth aging effects, especially when the age gap between source age and target age becomes large, making them not robust under extreme conditions. However, IPAPFA-GAN train several generators independently to learn aging effects between two different age group to generate high-quality aged faces with realistic aging effects, which makes it robust to various extreme conditions.

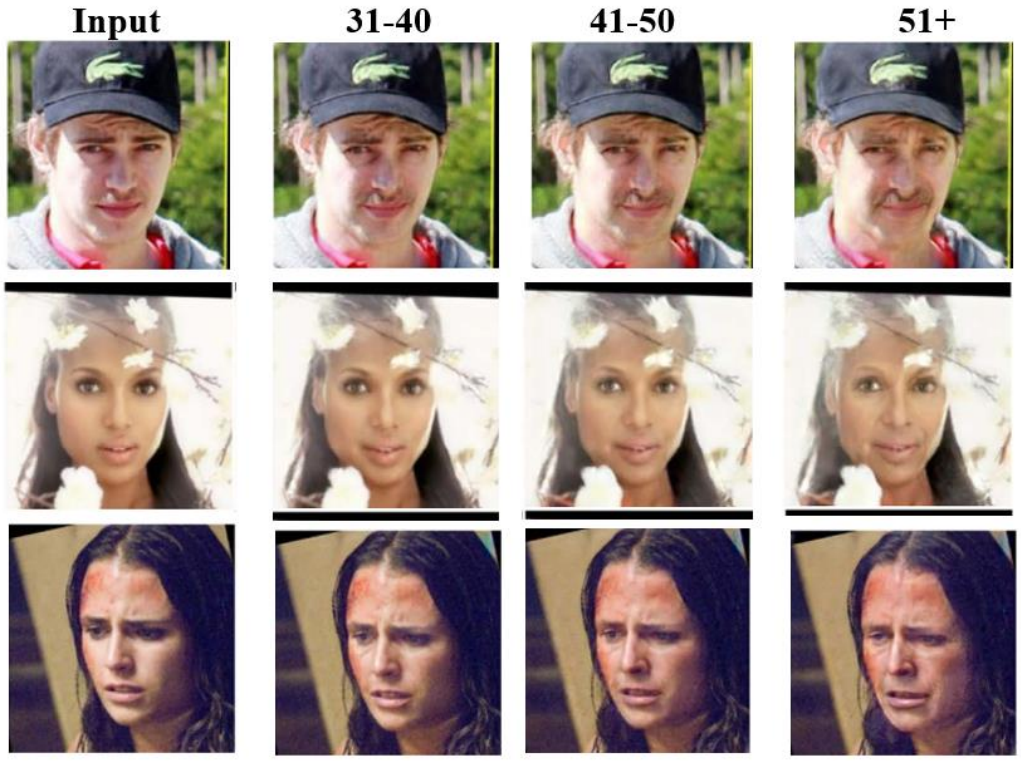


Figure 6.6: Robustness analysis of aged faces by IPAPFA-GAN

Figure 6.7 depicts the generated Ethiopian faces by the proposed model (IPAPFA-GAN).

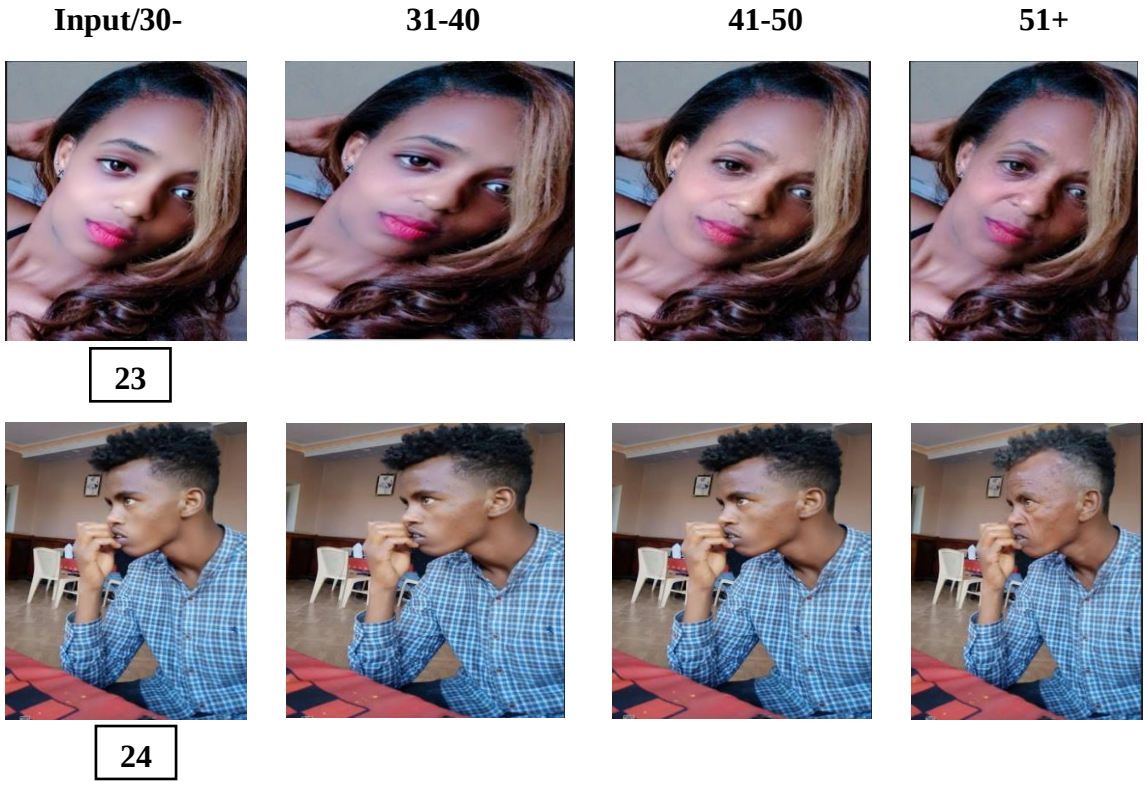


Figure 6.7: Tested aged Ethiopian faces by IPAPFA-GAN model

6.4. Evaluation Metrics

6.4.1. Inception Score

Table 6.1 below demonstrates the inception score for all experiments performed. This score simultaneously measures the images variety and whether each image distinctly looks like something. If both are true, the score will be high and if either or both are false, the score will be low. A higher score is better this means GAN can generate many different distinct images and the model tends to produce higher-image quality faces. The inception score is higher on the CACD dataset due to large variations in the dataset. The results are calculated over aged test faces for a fair comparison.

Table 6.1: Inception score for all experiments

No.	Experiments	Inception Score (IS)
1.	PFAGAN	33.39
2.	PFAGAN-AL	33.52
3.	PFAGAN-AL-LS	33.71
4.	IPAPFA-GAN	**34.23**

The bold value indicates the best results in the experiments.

6.4.2. Pearson Correlation Coefficient

Pearson correlation coefficient (PCC) is used to evaluate the performance of the aging accuracy and smoothness of different methods. Table 6.2 below shows the PCC for the experiments conducted. The higher PCC demonstrates that the higher aging accuracy, the smother aging result, and the better the model will be.

Table 6.2: Pearson correlation coefficient results for all experiments

No.	Experiments	Pearson Correlation Coefficient (PCC)
1.	PFAGAN	0.986
2.	PFAGAN-AL	0.989
3.	PFAGAN-AL-LS	0.991
4.	IPAPFA-GAN	**0.993**

The bold value indicates the best results in the experiments.

6.4.3. Verification Confidence Score

Table 6.3 below indicates the verification confidence for the performed experiments. The verification confidence score is measured between the input faces and the synthetic aged images from three age groups of the same subject. Face++ API is used for face comparing which means checking the likelihood that two faces belong to the same person. So, you will get a confidence score and threshold to evaluate the similarity. The high verification confidence score indicates that the identity information is consistently preserved.

Table 6.3: Verification confidence score for all experiments

No.	**Experiments**	Age group 31-40 41-51 51+		
		Verification Confidence Score (VCS)		
		30-		
1.	PFAGAN	96.21	94.50	89.80
2.	PFAGAN-AL	96.32	94.57	89.81
3.	PFAGAN-AL-LS	96.45	94.62	89.85
4.	IPAPFA-GAN	**97.03**	**95.12**	**90.42**

The bold values indicate the best results in the experiments.

6.5. Discussion

6.5.1. Discussion on IS results

To evaluate the quality and performance of the model's, inception score (IS) was performed as explained in section 6.4. PFAGAN-AL is implemented by replacing the pixel-wise loss of the baseline work with attention loss and scored 33.52 IS. PFAGAN-AL-LS is implemented with the addition of least-square GAN to the PFAGAN-AL and the IS is enhanced from 33.52 to 33.71. It strokes PFAGAN-AL by IS because least-square GAN improves the quality of generated images and stabilizes the training process. Finally, IPAPFA-GAN is implemented with the addition of attention mechanism to the PFAGAN-AL-LS and the IS is enhanced from 33.71 to 34.23. It strokes PFAGAN-AL-LS by IS and eliminates blurriness and accumulative artifacts to produce high-quality face images. In general, this experiment demonstrates the importance of the attention mechanism (self-attention), attention loss, and least-square GAN in this study.

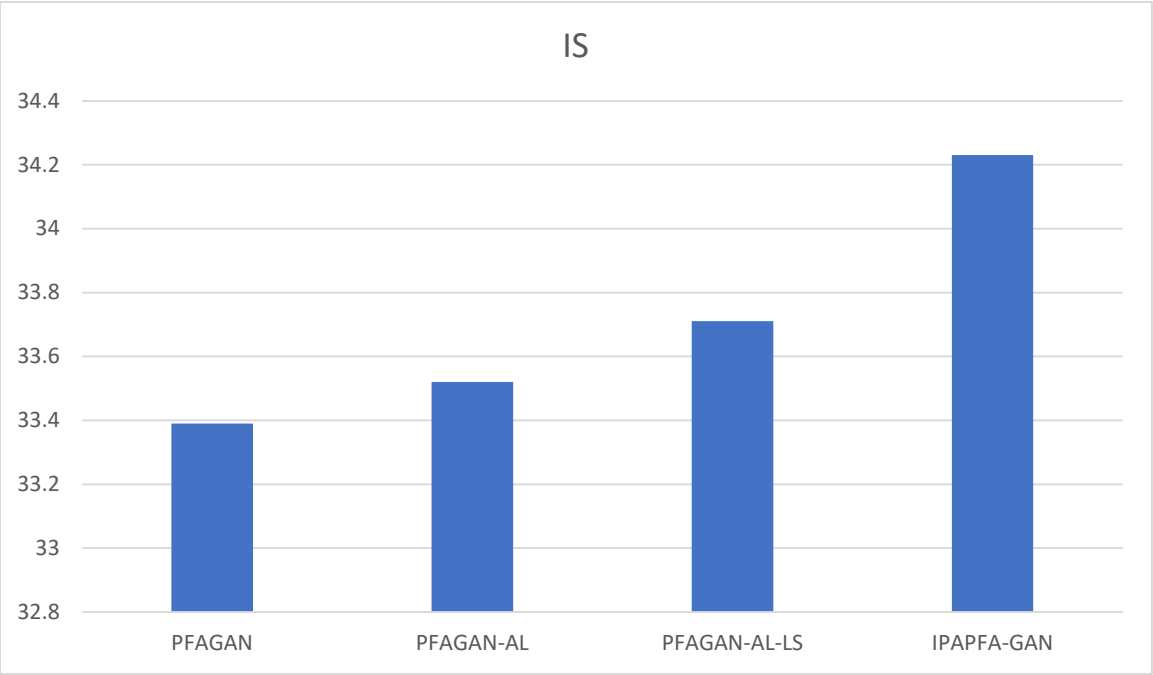


Figure 6.8: IS comparison of the four models

6.5.2. Discussion on PCC results

To evaluate the model performance in terms of aging accuracy and smoothness, pearson correlation coefficient (PCC) was used as demonstrated in section 6.4. PFAGAN-AL scored 0.989 PCC value. PFAGAN-AL-LS scored 0.991 PCC value and beats PFAGAN-AL by PCC value. The final proposed solution, IPAPFA-GAN scored 0.993 PCC value and beats both PFAGAN-AL and PFAGAN-AL-LS by PCC value. Therefore, in addition to age estimation loss, attention mechanism, attention loss, and least-square GAN play an important role in obtaining better aging accuracy and smoother aging results in this study.

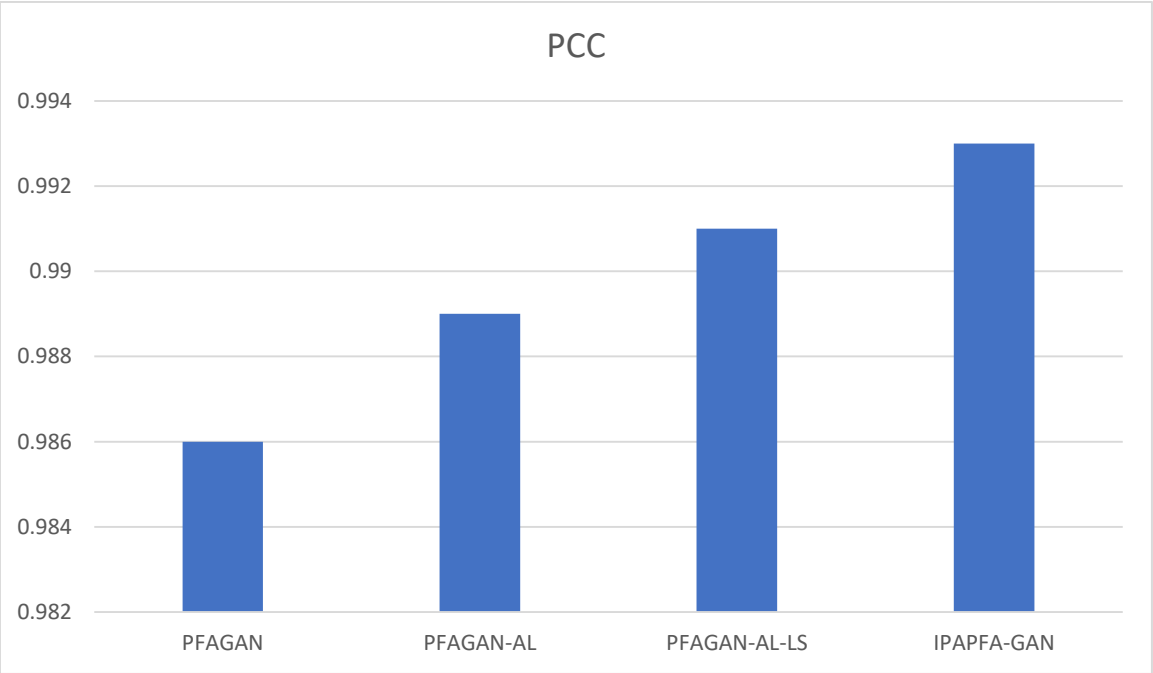


Figure 6.9: PCC comparison of the four models

6.5.3. Discussion on VCS results

A Verification confidence score (VCS) was used to evaluate the performance of the model in terms of identity preservation. Identity preservation is one of the essential requirements of face aging. A VCS is used to check the identity preservation between input faces and generated images from three age groups. PFAGAN-AL scored 96.32, 94.57, and 89.81 average verification confidence between input faces and aged face images from three age groups respectively. PFAGAN-AL-LS scored 96.45, 94.62, and 89.85 average verification confidence respectively, and beats PFAGAN-AL by average verification confidence. IPAPFA-GAN scored 97.03, 95.12, and 90.42 respectively, and beats both PFAGAN-AL and PFAGAN-AL-LS by average verification confidence. As we observe from this score, verification confidence slightly decreases as more changes appear in the face when the age gap becomes age and it is simple to understand the importance of the attention mechanism (self-attention) in preserving identity-related information.

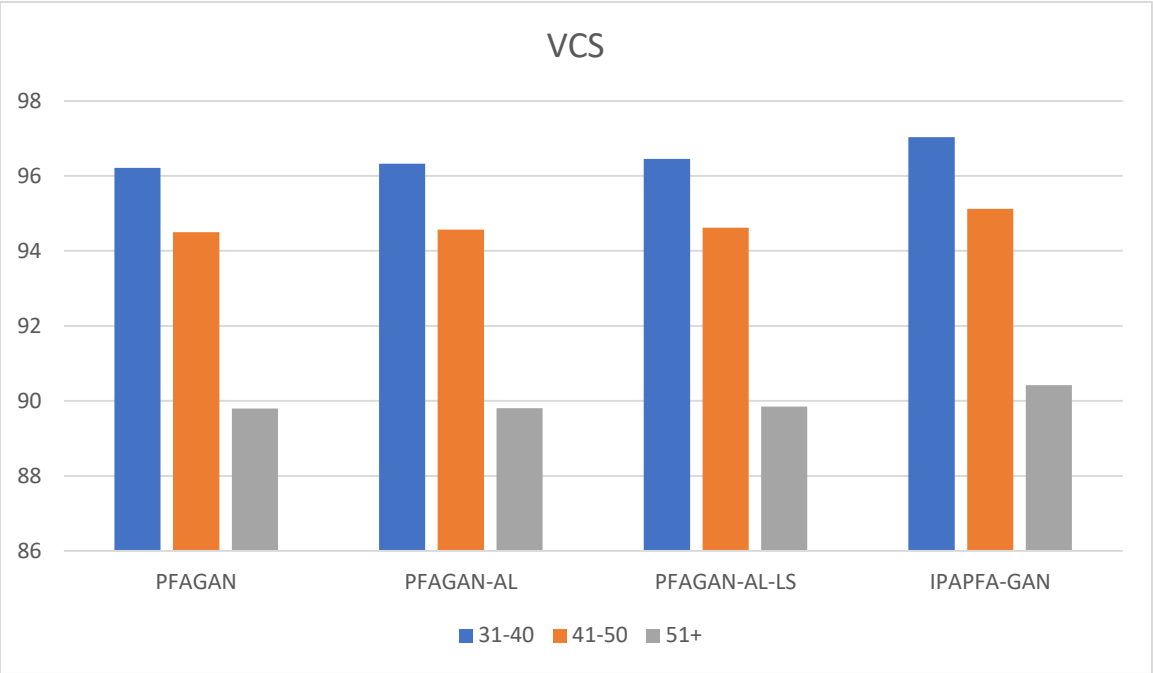


Figure 6.10: VCS comparison for the four models

CHAPTER SEVEN

7. CONCLUSION AND FUTURE WORK

7.1. Conclusion

Nowadays, image-to-image translation has shown impressive results in translating an image from one domain to another domain. It can be applied in different areas such as face aging, gender-swapping, object transformation, style transfer, season translation, and photo enhancement. Face aging is an extension of image-to-image translation where the goal is to translate a face image belonging to an age class to an image of a different age class. One of the main problems in face aging is meeting the three basic requirements of face aging at the same time. To tackle this issue, using an attention mechanism and adding a learning function to the face aging network may be an alternative solution.

This work presented identity-preserved and attention-based progressive face aging using the GAN model. This study aimed to improve the quality of the synthesized image by adding a learning function and an attention mechanism to the generator and the discriminator network of the baseline work (Huang et al., 2021). Among the experiments, the goal was to generate aged face images that are visually realistic from the input face images with small computational cost as possible. To do so, this study adds a self-attention mechanism, least-square GAN, and replaces pixel-wise loss with attention loss to the baseline work. Of course, these modifications enable IPAPFA-GAN to generate aged face images that are closer to the original, while preserving unique identities.

The experiment demonstrated the logical process of the architecture with the mathematical expression used to advance image quality. Among the four models, IPAPFA-GAN has better visual quality and performance with different evaluation metrics (IS, PCC, and VCS), as shown in the results section. PFAGAN-AL replaces the pixel-wise loss of the baseline work with attention loss and has shown better visual quality and performance than baseline work. PFAGAN-AL-LS adds least-square GAN to PFAGAN-AL for the discriminator network to improve the quality of synthesized images. Hence, it has shown better visual quality and performance than the PFAGAN-AL model. Finally, the proposed solution (IPAPFA-GAN) adds

a self-attention mechanism to the PFAGAN-AL-LS and outperforms the three models in terms identity-preservation and image quality.

As discussed in the result section replacing the pixel-wise loss with attention loss preserves identity-related information and keeps background information unchanged in the PFAGAN-AL model. So, this has answered the second research question of what is the effect of adding identity-preserving constraints in the generating aged human face images. The least-square GAN improves the quality of generated images and stabilize the training process in the PFAGAN-AL-LS model. This has answered the first research question of what is the effect of adding least-square GAN in generating aged face images. Since the attention mechanism only modifies the regions relevant to face aging, the proposed approach maintains identity-related information, and improves the visual quality of images. So, this has addressed the third research question of what is the effect of adding a self-attention mechanism in the generation of aged human face images.

Furthermore, quantitative experiments validate the effectiveness of our approach. The IPAPFA-GAN beats the three models by inception score, Pearson correlation coefficient, and verification confidence score. Finally, it is reasonable to conclude that adding an attention mechanism (self-attention), least-square GAN, and attention loss to the face aging network would yield a better performance due to they make the network focus on more important features, improve image quality, obtain a better aging accuracy and preserve identity-related information. Experimental results on CACD benchmarked dataset demonstrated the superiority of IPAPFA-GAN over the state-of-the-art cGAN-based methods in terms of identity-preservation, image quality, aging accuracy, and aging smoothness.

7.2. Future Work

As observed from the experiment, the model is mainly dependent on the attention mechanism (self-attention), least-square GAN, and attention loss. The generated output of the proposed solution mainly depends on attention mechanism performance as discussed in section 6.4. Based on this one incorporating another attention mechanism such as contextual, spatial and scale attention to the model can improve the execution of the model. To make this study applicable to different areas, the face rejuvenation task is also important. So, face rejuvenation deserves studying as future work. Taking the study in this way could make it more robust and fundamental to be utilized in a wide range of application areas.

SPECIAL ACKNOWLEDGMENT

This research project is funded by Adama Science and Technology University under the grant number:

ASTU/SM-R/554/22

Adama, Ethiopia

REFERENCES

Astawa, I. N. G. A., Radhitya, M. L., Ardana, I. W. R., & Dwiyanto, F. A. (2021). Face Images Classification using VGG-CNN. *Knowledge Engineering and Data Science*, *4*(1), 49. https://doi.org/10.17977/um018v4i12021p49-54

Bahdanau, D., Cho, K. H., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.

Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2018). VGGFace2: A dataset for recognising faces across pose and age. *Proceedings - 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018*, *June*, 67–74. https://doi.org/10.1109/FG.2018.00020

Chen, H., & Haoyu, C. (2019). Face Recognition Algorithm Based on VGG Network Model and SVM. *Journal of Physics: Conference Series*, *1229*(1). https://doi.org/10.1088/1742-6596/1229/1/012015

Choi, K. S., Shin, J. S., Lee, J. J., Kim, Y. S., Kim, S. B., & Kim, C. W. (2005). In vitro trans-differentiation of rat mesenchymal cells into insulin-producing cells by rat pancreatic extract. *Biochemical and Biophysical Research Communications*, *330*(4), 1299–1305. https://doi.org/10.1016/j.bbrc.2005.03.111

Elmahmudi, A., & Ugail, H. (2021). A framework for facial age progression and regression using exemplar face templates. *Visual Computer*, *37*(7), 2023–2038. https://doi.org/10.1007/s00371-020-01960-z

Gholamy, A., Kreinovich, V., & Kosheleva, O. (2018). Why 70/30 or 80/20 Relation Between Training and Testing Sets : A Pedagogical Explanation. *Departmental Technical Reports (CS)*, 1–6.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

H. Yang, D. Huang, Y. Wang, A. K. J. (2018a). Face aging with identity-preserved conditional generative adversarial networks. *In Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 7939–7947.

H. Yang, D. Huang, Y. Wang, A. K. J. (2018b). Learning face age progression: A pyramid architecture of GANs. *In Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 31–39.

He, F., Liu, T., & Tao, D. (2020). Why ResNet Works? Residuals Generalize. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(12), 5349–5362. https://doi.org/10.1109/TNNLS.2020.2966319

Huang, Z., Chen, S., Zhang, J., & Shan, H. (2021). PFA-GAN: Progressive Face Aging with Generative Adversarial Network. *IEEE Transactions on Information Forensics and Security*, *16*, 2031–2045. https://doi.org/10.1109/TIFS.2020.3047753

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 5967–5976. https://doi.org/10.1109/CVPR.2017.632

J. Song, J. Zhang, L. Gao, X. Liu, H. T. S. (2018). Dual conditional GANs for face aging and rejuvenation. *In Proc. Int. Joint Conf. Artif. Intell.*, 899–905.

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.

Knyaz, V. A., Kniaz, V. V., & Remondino, F. (2019). Image-to-voxel model translation with conditional adversarial networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11129 LNCS*, 601–618. https://doi.org/10.1007/978-3-030-11009-3_37

Kulpati, S. (2019). A Brief Introduction To GANs. *Sigmoid*.

Liu, Y., Li, Q., Sun, Z., Member, S., & Tan, T. (n.d.). *A 3 GAN : An Attribute-aware Attentive Generative Adversarial Network for Face Aging*. 1–16.

Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., & Smolley, S. P. (2017). Least Squares Generative Adversarial Networks. *Proceedings of the IEEE International Conference on Computer Vision*, *2017-Octob*, 2813–2821. https://doi.org/10.1109/ICCV.2017.304

Mirza, M., & Osindero, S. (2014). *Conditional Generative Adversarial Nets*. 1–7. http://arxiv.org/abs/1411.1784

Osman AM, V. S. (2018). a survey of the state-of-the-art. *ICTAS Conference on Information Communications Technology and Society*. https://doi.org/10.1109/ICTAS.2018. 8368755

Oza, M., Vaghela, H., & Bagul, S. (2019). Semi-Supervised Image-To-Image Translation. *Proceeding - 2019 International Conference of Artificial Intelligence and Information Technology, ICAIIT 2019*, *Nips*, 16–20. https://doi.org/10.1109/ICAIIT.2019.8834613

Q. Li, Y. Liu, Z. S. (2020). Age progression and regression with spatial attention modules. *In Proc. AAAI Conf. Artif. Intell.*

Rothe, R., Timofte, R., & Van Gool, L. (2018). Deep Expectation of Real and Apparent Age from a Single Image Without Facial Landmarks. *International Journal of Computer Vision*, *126*(2–4), 144–157. https://doi.org/10.1007/s11263-016-0940-3

Sermanet, P., & Lecun, Y. (2011). Traffic sign recognition with multi-scale convolutional networks. *Proceedings of the International Joint Conference on Neural Networks*, 2809–2813. https://doi.org/10.1109/IJCNN.2011.6033589

Shao, J., Science, C., & Engineering, S. (2021). *WAVELET-BASED MULTI-LEVEL GANS FOR FACIAL Concordia University*. *November*.

Sharma, N., Sharma, R., & Jindal, N. (2021). Prediction of face age progression with generative adversarial networks. *Multimedia Tools and Applications*, *80*(25), 33911–33935. https://doi.org/10.1007/s11042-021-11252-w

Shu, X., Tang, J., Lai, H., Liu, L., & Yan, S. (2015). Personalized age progression with aging dictionary. *Proceedings of the IEEE International Conference on Computer Vision*, *2015 Inter*, 3970–3978. https://doi.org/10.1109/ICCV.2015.452

T, R. (2020). Simplified arrival biometrics land at key US airports. *ScienceDirect. Biometric*

Technology Today, *2020*(10), 3.

W. Wang, Z. Cui, Y. Yan, J. Feng, S. Yan, X. Shu, N. S. (2016). Recurrent face aging. *In IEEE Conference on Computer Vision and Pattern Recognition*, 2378–2386.

Y. Liu, Q. Li, Z. S. (2019). Attribute-aware face aging with wavelet-based generative adversarial networks. *In Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 11 869–11 878.

Yadav, R., Sancheti, P., Pagar, T., Wagh, D., & Mutrak, P. A. S. (n.d.). *E-Bazzar*. *08*(02), 3–7.

Yang, H., Huang, D., Wang, Y., & Jain, A. K. (2021). Learning Continuous Face Age Progression: A Pyramid of GANs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(2), 499–515. https://doi.org/10.1109/TPAMI.2019.2930985

Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). Self-attention generative adversarial networks. *36th International Conference on Machine Learning, ICML 2019*, *2019-June*, 12744–12753.

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. https://doi.org/10.1109/LSP.2016.2603342

Zhang, S., Cheng, D., Jiang, D., & Kou, Q. (2020). Least squares relativistic generative adversarial network for perceptual super-resolution imaging. *IEEE Access*, *8*, 185198–185208. https://doi.org/10.1109/ACCESS.2020.3030044

Zhang, Z., Song, Y., & Qi, H. (2017). Age progression/regression by conditional adversarial autoencoder. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 4352–4360. https://doi.org/10.1109/CVPR.2017.463

Zhu, H., Huang, Z., Shan, H., & Zhang, J. (2020). Look globally, age locally: Face aging with an attention mechanism. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, *2020-May*, 1963–1967. https://doi.org/10.1109/ICASSP40776.2020.9054553

APPENDIX

Appendix A: Randomly generated images by IPAPFA-GAN

Epoch 1

Input/30- 31-40 41-50 51+

Epoch 5

Epoch 10

Epoch 50

Epoch 100

Epoch 195

Epoch 200

Appendix B: Sample code

Appendix B.1: Dataset preprocessing

```
from mtcnn.mtcnn import MTCNN
from PIL import Image
import numpy as np
import cv2
import os
from tqdm import tqdm
from skimage import transform as trans

#define read path
root_data_dir="C:/Users/CV-LAB/Desktop/IPAPFA_GAN-main/data"
image_root_dir=os.path.join(root_data_dir,"CACD2000")
#define store path
store_root_dir="C:/Users/CV-LAB/Desktop/IPAPFA_GAN-main/data"
store_image_dir=os.path.join(store_root_dir,"CACD2000_processed")
if os.path.exists(store_image_dir) is False:
    os.makedirs(store_image_dir)

#define some param for mtcnn
src = np.array([
 [30.2946, 51.6963],
 [65.5318, 51.5014],
 [48.0252, 71.7366],
 [33.5493, 92.3655],
 [62.7299, 92.2041] ], dtype=np.float32 )
threshold = [0.6,0.7,0.9]
factor = 0.85
minSize=20
imgSize=[120, 100]
detector=MTCNN(steps_threshold=threshold,scale_factor=factor,min_face_size=minSize)

#align,crop and resize
keypoint_list=['left_eye','right_eye','nose','mouth_left','mouth_right']

for filename in tqdm(os.listdir(image_root_dir)):
    dst = []
    filepath=os.path.join(image_root_dir,filename)
    storepath=os.path.join(store_image_dir,filename)
    npimage=np.array(Image.open(filepath))
    #Image.fromarray(npimage.astype(np.uint8)).show()

    dictface_list=detector.detect_faces(npimage)#if more than one face is detected, [0] means
choose the first face
```

```
    if len(dictface_list)>1:
        boxs=[]
        for dictface in dictface_list:
            boxs.append(dictface['box'])
        center=np.array(npimage.shape[:2])/2
        boxs=np.array(boxs)
        face_center_y=boxs[:,0]+boxs[:,2]/2
        face_center_x=boxs[:,1]+boxs[:,3]/2
        face_center=np.column_stack((np.array(face_center_x),np.array(face_center_y)))
        distance=np.sqrt(np.sum(np.square(face_center - center),axis=1))
        min_id=np.argmin(distance)
        dictface=dictface_list[min_id]
    else:
        if len(dictface_list)==0:
            continue
        else:
            dictface=dictface_list[0]
    face_keypoint = dictface['keypoints']
    for keypoint in keypoint_list:
        dst.append(face_keypoint[keypoint])
    dst = np.array(dst).astype(np.float32)
    tform = trans.SimilarityTransform()
    tform.estimate(dst, src)
    M = tform.params[0:2, :]
    warped = cv2.warpAffine(npimage, M, (imgSize[1], imgSize[0]), borderValue=0.0)
    warped=cv2.resize(warped,(256,256))
    Image.fromarray(warped.astype(np.uint8)).save(storepath)
```

Appendix B.2: Train discriminator and generator

```
def train(self, inputs, n_iter):
        opt = self.opt
        source_img, true_img, source_label, target_label, true_label, true_age, mean_age =
inputs
        self.generator.train()
        self.discriminator.train()

        if opt.image_size < opt.pretrained_image_size:
            source_img_small = F.interpolate(source_img, opt.image_size)
            true_img_small = F.interpolate(true_img, opt.image_size)
        else:
            source_img_small = source_img
            true_img_small = true_img
        g_source = self.generator(source_img_small, source_label, target_label)
        if opt.image_size < opt.pretrained_image_size:
```

```
        g_source_pretrained = F.interpolate(g_source, opt.pretrained_image_size)
    else:
        g_source_pretrained = g_source

    ########Train Discriminator###########
    self.d_optim.zero_grad()
    d1_logit = self.discriminator(true_img_small, true_label)
    # d2_logit = self.discriminator(true_img, source_label)
    d3_logit = self.discriminator(g_source.detach(), target_label)

    # d_loss = 0.5 * (ls_gan(d1_logit, 1.) + ls_gan(d2_logit, 0.) + ls_gan(d3_logit, 0.))
    d_loss = 0.5 * (ls_gan(d1_logit, 1.) + ls_gan(d3_logit, 0.))

    with amp.scale_loss(d_loss, self.d_optim) as scaled_loss:
        scaled_loss.backward()
    self.d_optim.step()

    ########Train Generator###########
    self.g_optim.zero_grad()
    ################################GAN_loss#############################
    gan_logit = self.discriminator(g_source, target_label)
    # g_loss = 0.5 * ls_gan(gan_logit, 1.)
    g_loss = ls_gan(gan_logit, 1.)

    ################################Age_Loss#############################
    age_loss = self.age_criterion(g_source_pretrained, mean_age)

    ################################Feature_loss#########################
    fea_loss = F.fea_loss(g_source_pretrained, source_img)

    ################################SSIM_loss############################
    ssim_loss = compute_ssim_loss(g_source_pretrained, source_img, window_size=10)

    ################################Attention_loss#######################
    att_loss = compute_att_loss(g_source_pretrained, source_img, window_size=10)

    ################################ID_loss##############################
    id_loss = F.mse_loss(self.extract_vgg_face(g_source_pretrained),
self.extract_vgg_face(source_img))

    total_loss = g_loss * opt.gan_loss_weight + \
                 (α_att * att_loss + α_ssim * ssim_loss + α_fea * fea_loss)  + \
                 id_loss * opt.id_loss_weight + \
                 age_loss * opt.age_loss_weight

    with amp.scale_loss(total_loss, self.g_optim) as scaled_loss:
        scaled_loss.backward()
```

```
        self.g_optim.step()

        self.logger.msg([
            d1_logit, d3_logit, gan_logit, age_loss, l1_loss, ssim_loss, id_loss
        ], n_iter)
```

Appendix B.3: Train age estimation network

```
def train_net(net, device, global_step=0):
    train_data = AgeDataset(do_transforms=True)
    n_train = len(train_data)

    logging.info(f 'Data loaded having {n_train} images')

    train_loader = DataLoader(train_data, batch_size=exp_config.age_batch_size,
                              shuffle=True, num_workers=exp_config.n_workers, pin_memory=True)
    print(train_loader.batch_sampler)
    writer =
SummaryWriter(comment='AgeEstimator_LR_{}_BS_{}'.format(exp_config.age_lr,
exp_config.age_batch_size))

    optimizer = optim.Adam(net.parameters(), lr=exp_config.age_lr, weight_decay=1e-8)
    scheduler = optim.lr_scheduler.StepLR(optimizer,
step_size=exp_config.age_lr_decay_steps,
                                          gamma=exp_config.age_lr_decay_rate)
    criterion = utils.age_criterion
    max_epochs = exp_config.age_max_epochs -
global_step//math.ceil(n_train/exp_config.age_batch_size)
    logging.info(f ' ' 'Starting training from step {global_step}:
            Epochs:          {max_epochs}
            Batch size:      {exp_config.age_batch_size}
            Learning rate:   {exp_config.age_lr}
            Training size:   {n_train}
            Device:          {device.type}
        ' ' ')
    net.train()
```

Appendix B.4: Sub-generator implementation

```
class SubGenerator(nn.Module):

    def __init__(self, in_channels=3, repeat_num=4, norm_layer='bn'):
        super(SubGenerator, self).__init__()
        layers = [
            get_norm_layer(norm_layer, nn.Conv2d(in_channels, 32, 9, 1, 4)),
            nn.LeakyReLU(0.2, inplace=True),
            get_norm_layer(norm_layer, nn.Conv2d(32, 64, 4, 2, 1)),
            nn.LeakyReLU(0.2, inplace=True),
            get_norm_layer(norm_layer, nn.Conv2d(64, 128, 4, 2, 1)),
            nn.LeakyReLU(0.2, inplace=True),
        ]
        for _ in range(repeat_num):
            layers.append(ResidualBlock(128, norm_layer))
        layers.extend([
            get_norm_layer(norm_layer, nn.ConvTranspose2d(128, 64, 4, 2, 1)),
            nn.LeakyReLU(0.2, inplace=True),
            get_norm_layer(norm_layer, nn.ConvTranspose2d(64, 32, 4, 2, 1)),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Conv2d(32, 3, 9, 1, 4),
        ])
        self.main = nn.Sequential(*layers)

    def forward(self, x):
        return self.main(x)
```