“Temporal Cycle Consistency: for a Video-to-Video Translation.”

By

Kirubel Abebe Senbeto

A Thesis Submitted to Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Office of Graduate Studies

Adama Science and Technology University

October, 2020

Adama, Ethiopia

“Temporal Cycle Consistency: for a video-to-Video Translation.”

By

Kirubel Abebe Senbeto

Advisor: Prof Yun Koo Chung

A Thesis Submitted to Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Office of Graduate Studies

Adama Science and Technology University

October, 2020

Adama, Ethiopia

APPROVAL OF THE BOARD OF EXAMINERS

We, the undersigned, members of the Board of Examiners of the final open defense by Kirubel Abebe Senbeto, have read and evaluated his thesis entitled "Temporal Cycle Consistency: for a Video-to-Video Translation" and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirement of the degree of Masters in Computer Science and Engineering (CSE).

Name	**Signature**	**Date**
Kirubel Abebe Senbeto		
Name of the Student		
Prof. Yun Koo Chung		
Advisor		
External Examiner		
Internal Examiner		
Chair Person		
Head of Department		
School Dean		
Post Graduate Dean		

DECLARATION

I hereby declare that this MSc thesis is my original work and has not been presented for a degree in any other university, and all sources of material used for this thesis have been duly acknowledged.

Name	Signature
Kirubel Abebe Senbeto	_____

This MSc thesis has been submitted for examination with my approval as a thesis by

Advisor. Name	Signature
Yun Koo Chung (Ph.D.)	_____

ACKNOWLEDGMENT

First and foremost, I would like to thank the HOLY TRINITY FATHER, SON, and HOLY SPRITE the Almighty GOD who created everything seen and unseen. Also, I love to Praise the Virgin Mary the Holy Mother of LORD JESUS CHRIST by the song of St. Yared the Ethiopian.

This thesis research is the outcome of a one-year study and hard work, but it's very hard to remember when so many people have helped me in far too many different ways. However, some should be honored for the essential assistance they have offered in the course of study.

I would like to thank Prof. Yun Koo Chung (Ph.D.) head of the Computer Vision and Robotics SIG for his patient guidance on any issue and his recommendation he has made since the beginning of this research. Dr. Mesfin Abebe (Ph.D.) has been supporting me throughout the research and made significant recommendations to the research. Another person worth mentioning is Anteneh Tilaye (M.Sc.) for his support in evaluating the outputs produced. I am also grateful for all the support and advice I have received from computer vision postgraduate students, and friends.

I would also like to acknowledge Capital Ethiopia and Aware Production which have permitted me to use respective videos as a dataset.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

2D	2-Dimensional
3D	3-Dimensional
AED	Annotation Edit Distance
AEE	Average endpoint error
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
CC	Cycle Constraint
CC+CP	Cycle Constraint Plus Feature Preserving
CC+CP+TD	Cycle Constraint Plus Feature Preserving Plus Temporal Aware Discriminator
CGAN	Conditional Generative Adversarial Network
CGI	Computer-Generated Imagery
CNN	Convolutional Neural Network
Conv-nets	Convolutional Neural
CPU	Central Processing Unit
Cycle GAN	Unpaired Image-To-Image Translation Using Cycle-Consistent
DL	Deep Learning
FCN	Fully Convolutional Networks
FID	Fréchet Inception Distance
GAN	Generative Adversarial Networks
GPU	Graphics Processing Unit
HFR	High Frame Rate
IDE	Integrated Development Environment
IoU	Intersect Over Union
IS	Inception Score
L1	Manhattan Distance or L1 Norm
L2	Euclidean Distance or L2 Norm
LSTM	Long Short-Term Memory
mIoU	Mean Intersect Over Union

ML	Machine Learning
MoCycle-GAN	Mocycle-GAN: Unpaired Video-To-Video Translation
MPI	Max Planck Institute
P1	Human Evaluation Study Protocol One
P2	Human Evaluation Study Protocol Two
Pix2Pix	Image-To-Image Translation With Conditional Adversarial Nets
RC	ReCycle GAN
RC+TD	ReCycle GAN Plus Temporal Discriminator
ReCycle GAN	ReCycle GAN: Unsupervised Video Retargeting
RGB	Red Green Blue
RNN	Recurrent Neural Network
TPU	Tensor Processing Unit
VAE	Variational Autoencoder

LIST OF NOTATIONS AND SYMBOLS

Notation	***Meaning***
A	Indicate a given video is domain A
B	Indicate a given video is domain B
D_x	Discriminator X
D_y	Discriminator Y
$D_x^{t:3}$	Discriminator consider three frames
f_t	Optical flow between two frames
$\approx$	Indicate retranslation back to original domain
$\sim$	Indicate translation to another domain
G_{AB}	Generator, network map form domain A to domain B
G_{BA}	Generator, network map form domain B to domain A
$O_{x_t}^n$	Object n in frame x_t
X	Sample video from domain A
x_t	Sample frame from domain X
$X \to Y \; or \; (Y \to X)$	Translation from domain X to domain Y or (Y to domain X)
Y	Sample video from domain B
y_t	Sample frame from domain Y
z	Noise data sample

ABSTRACT

Generative Adversarial Network (GAN) is a deep learning method that is developed for synthesizing data. Area of applications for which it can be used are image-to-image translations, Video-to-video translation, and video retargeting. However, to train those models there is a need for large amounts of complex paired data which is hard to find. To collect datasets, especially when we need a paired dataset is time-consuming and expensive. One way to overcame this problem is to collect datasets in one domain and translate it to another domain using translation techniques to make it a paired dataset. Various research has leveraged enormous in image translation by the use of GANs on an unpaired dataset. As far as video translation is concerned, current GAN-based approaches do not entirely leverage space-time knowledge in videos. This research examines the idea of using GANs for the utilization of spatial-temporal information in a video by extending the unpaired video-to-video translations model (ReCycle GAN) to enhance spatial-temporal video translation. In particular, previous methods suffer from Object disappearance, Object dislocation, and flickering Artifacts. To Mitigate these issues, this work proposes to add feature preserving loss and temporal aware discriminator to the Cycle GAN and ReCycle GAN to generate more temporal consistent videos. Extensive qualitative and quantitative assessments demonstrate the notable success of the proposed system against existing methods. Average human evaluation study has shown that this research excels at 60% compared to Cycle GAN and 35% on ReCycle GAN. This paper concludes that adding feature preserving constraints and temporal aware discriminator does improve temporal coherency of generated output video.

Keywords: *Cycle GAN, ReCycle GAN, Spatio-Temporal information, Unsupervised Video-to-Video translation*

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

Computer Vision is measured among the most fascinating fields in computer engineering and artificial intelligence. The chase of providing machines with a sense of sight that is even better than that of humans is keeping researchers busy and motivated. There is an extensive range of problems with active research within the field of computer vision, such as facial recognition, object classification, scene recognition, and domain transfer. In this thesis, the focus is on domain transfer.

Unsupervised video retargeting is the transference of sequential content information form one domain to another while preserving the style of the target domain. Such domain transfer could be served in numerous areas including motion translation from one person to another person and video colorization – monochrome video to color, and day time video to night time video. Recent works[1]–[4] use the generative adversarial network for retargeting and image-to-image translation problems. This study aims to extend ReCycle GAN by Bansal et al. [1] to improve frame to frame continuity (motion consistency) by introducing additional constraints to the network.

Image transfer is a subproblem of domain transfer that aims to translate or map domain to domain. Such domain transfer could be served in numerous areas, including from classical language translation to motion translation from one person to another person and video colorization. Since this work uses Images and video as input data, we can say that image transfer is a process of repainting a given image by style image while preserving original contents (for example, Figure 1-1).

In the year 2018, several artificial intelligence news [5] headlines were screaming about a paint drawn 100% by AI sold $432,500 (fascinating, isn't it?). But because of image transfer data scientist does not need to buy a hundred-thousand-dollar painting for decorating his living room while he can have one when he is home sitting in front of his laptop, marvelous.

$$Input\ Image\ +\ Style\ Image\ ->\ Output\ Image\ (Styled\ Input)$$

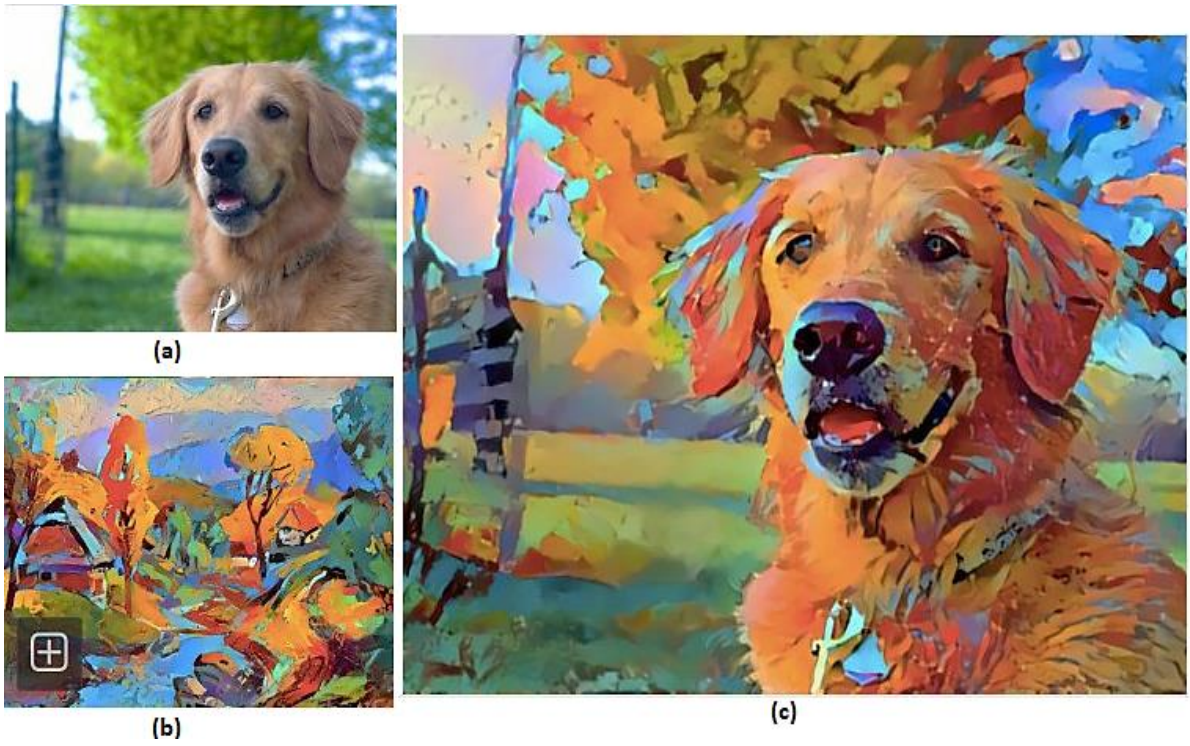


Figure 1-1(a) input image. (b) style image. (c) output image.

Source: Adapted from [6]

Perhaps, the first successful neural style transfer paper was published in 2015 by Gatys et al. [7]**.** After this work, many researchers came with a more realistic synthetic image. Pix2Pix [3] introduces with a supervised image-to-image translation, but Pix2Pix needs paired data for training which is expensive and unlikely –needs paired data examples from both domains to learn. Other finest GAN paper Cycle-GAN by Zhu et al. [2] presents the unsupervised image to image transfer to overcome the Pix2Pix problem due cycle consistency –If I take an input image of a horse, feed it to the translation network it generates a zebra image then take the output image as an input again run the second transformation. I expect to get the same horse image I started with. Cycle-GAN place foundation for unsupervised image transfer problem in computer vision.

Video-to-video translation is a natural extension of an image-to-image translation (since the video is a sequence of images). Recent[1], [8]–[10] works use the generative adversarial network for retargeting style transfer and, images to image translation problems. This work

aims to extend video-to-video translation to the improved frame to frame continuity (motion consistency) by introducing additional constraints to the network.

In order to clearly understand this thesis research question, we need to have a brief introduction to the following topics. A more detailed discussion will be held in the proceeding section.

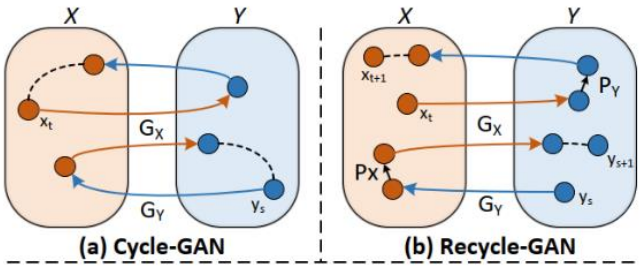


Figure 1-2 Cycle Gan vs Recycle GAN

Source: Adapted from[1]

1.1.1. Generative Adversarial Networks

GAN (Generative Adversarial Networks) fit into the conventional algorithms called Generative models. The term 'generative' refers to the fact that these networks can learn how to produce data samples that are similar to real ones in the training dataset [11]. It is a sub-set of ML which aims to study algorithms that learn the data distribution of the given data, deprived of specifying a target value. This method builds upon the success of using deep neural networks in content generation.

Generative adversarial networks are collected of two networks work against each other in a zero-sum game framework [11]. The first network is called a Generator. The goal is to produce new data close to that of samples from real datasets. The Generator could act as a human art forger, which creates fake works of art.

The second network is entitled, Discriminator. This model aims to recognize if an input data is '***real***' — goes to the original dataset — or if it is '***fake***' — generated by a falsifier generator network. In this scenario, a Discriminator is corresponding to the law enforcement agent (or an art expert), which tries to spot artworks as truthful or fraud. Successful training of a GAN requires reaching an equilibrium state between two opposing objectives, unlike

CNN or Long Short-Term Memory (LSTM) where the training objective is to minimize or maximize the value of a single loss function.

Conditional GAN: The conditional GAN [12] is an extension of the [11] original vanilla GAN, by introducing a conditioning variable into both generator and discriminator network. So instead of generating random data, the newly introduced condition variable would allow generating a particular data distribution specified by the conditioning variable. Mainly, the random noise input to the generator has been concatenated with a variable specifying the condition to generate the fake data, meaning to generate the fake data cGAN use random noise and newly introduced conditional variable.

Video-to-video transfer: Video-to-video transfer is a domain transfer problem that aims to transfer sequential content information form one domain to another while preserving the style of the target domain. Current approaches for domain transfer categories broadly into three classes. Early techniques use classical computer vision mechanism work specifically designed for particular body parts such as the human face [13] they lack generalization and does not work well if there is occlusion. The second approach use paired image-to-image translation such as Pix2Pix -in an image it takes a pixel then converts by another pixel. Isola et al. [3] use conditional GAN [12], learn a mapping between paired input to the output image. The third category is unsupervised and unpaired data domain transfer like Cycle-GAN [2], which enforces cycle consistency for the unpaired image.

The recent state of artwork work ReCycle-GAN by Bansal et al. [1] motivated by [2] proposes video retargeting via spatiotemporal constraint though directly synthesizing future frames via temporal predictor to preserve temporal continuity. Bansal et al., clams video-to-video translation is still under constraint since their work result shows of video-to-video transfer has very flickering output. This thesis work proposed to extend Bansal et al., work to improve temporal continuity between adjacent consecutive frames by introducing additional temporal cycle consistency constraints also proposes to introduce Spatiotemporal video-to-video translation for better realistic results.

1.2. Motivation of the Study

Recent deep learning achievement is due to the abundance of the enormous amount of data available nowadays. However, still, there is a big problem to collect data especially when there is a need for paired data set (such as day and night) since capturing datasets in two (or more) completely different environments is a cumbersome task. Domain transfer could be the mechanism to overcome such problems. After the first paper on GAN by Goodfellow et al. [11], GAN has been used in a wide range of areas for numerous applications and image-to-image translation maybe is the significant one. Advancement in GAN enables scientists and researchers to create fake images indistinguishable from real ones[14][15]. This advancement is applicable in automated content generation such as multimedia content generation, photo editing, game graphic/scenery design, and cinematic effects.

By extending the image translation idea, various researchers propose a number of approaches for the video-to-video translation network to learn both spatial and temporal domains but failed [16] to Achieve the potential found by image translation networks. Generally, currently used video-to-video translation networks, prone to object disappearance problems, and arbitrary strange motion on the generated videos make translated videos more unrealistic. This work tries to solve these problems by extending the solutions presented in previous works.

1.3. Statement of the Problem

Problem formulation: Inspired by recent work Recycle-GAN in the unpaired video-to-video translation, the notion of a research problem is as follows. Assume two videos archives in source and target domain $X = \{x\}_{t=1}^{T}$ and $Y = \{y\}_{x=1}^{T}$ respectively, cycle constraint enables an image-to-image translation in mutually frontward and backward mapping. There are two mapping functions $G_{AB}\ and\ G_{BA}$ mapping from domain $X \to Y$ and $Y \to X$ correspondingly form target domain to source and vice versa. $G_{AB}(x_t) = \widetilde{x_t}$ where x_t is input video frame at time t and $\widetilde{x_t}$ is a synthetic frame in X domain same is true in Y domain. Cycle consistency constraint $G_{BA}(\widetilde{x_t}) = \widetilde{\widetilde{x}_t}$ so then $\widetilde{\widetilde{x}_t} \approx x_t$ as well as $G_{BA}(\widetilde{y_t}) = \widetilde{\widetilde{y}_t}$ so then $\widetilde{\widetilde{y}_t} \approx y_s$.

Besides the preservation of cycle consistency in each frame, this work explores mapping temporal consistency between consecutive frames in both domains. Meaning considering an

optical flow between x_t and x_{t+1} is f_t and optical flow between $\tilde{\tilde{x}}_t$ and $\widetilde{\widetilde{x_{t+1}}}$ is $\tilde{\tilde{f}}_t$, then, temporal cycle consistency need to enforce motion consistency via minimizing the difference between f_t and $\tilde{\tilde{f}}_t$. Recycle-GAN [1] claims "*video-to-video translation is under constraint*". this work proposes toward adding temporal cycle consistency constraint to the extended video-to-video translation to see more coherence in its result. To do so, an extensive experimental attempt was made with the purpose of answering the research questions.

1.4. Research Questions

This work intends to answer the following research questions.

RQ1. What are the appropriate temporal consistency constraints for the unpaired video to video translation?

RQ2. Can adding temporal cycle consistency constraints improve temporal coherency for video translation?

RQ3. What is the effect of temporal discriminator regarding improving unsupervised video-to-video translation?

1.5. Objectives of the Study

1.5.1. General Objective

The general objective of the study is to incorporate a temporal cycle consistency constraint for video-to-video translation to the ReCycle GAN network.

1.5.2. Specific Objectives

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

- To identify learning constraints to improve temporal coherency for video to video translation.
- To embed feature preserving loss to Cycle-GAN.
- To modify the discriminator network to make it aware of temporal information.
- To add learning constraints to the Cycle-GAN and ReCycle-GAN networks.

» To Evaluate the performance of the model using inception score and human evaluation study.

1.6. Scope of the Study

The scope of the thesis work within the given time and resource includes: -

» Translate a given domain video (sequence of image) to another domain.
» Add learning constraints to Cycle-GAN and the ReCycle-GAN network.
» Blend spatial information to temporal information to improve the consistency of video-to-video translation.

1.7. Limitation of the Study

This thesis work does not cover the following due to time and resource limitations.

» The one-to-many video-to-video translation is not a part of this work. The network is trained to translate from one domain image to another domain, which is a one-to-one correspondence.
» The video does not zoom in or out throughout the whole process.
» This work does not consider the object occlusion problem.

1.8. Organization of the Thesis

The remainder of this thesis is organized as follows:

Chapter Two: discusses the background literature and related works regarding image-to-image translation, video retargeting, and video-to-video translation. This chapter also elasticities the theoretical framework of Deep Learning and Generative Adversarial Network.

Chapter Three: features the research methodology, including different methods and techniques used to develop the solution and select the appropriate one. Data collection method, design tools, prototype development framework and platforms, and evaluation methods are also discussed

Chapter Four: will cover points about the proposed learning function in detail. Model Architecture and pseudocode for implementation, and learning functions mathematical correspondent descriptions have been made.

Chapter Five: Explains how the desired proposed solution is implemented. and the working environment setup, cycle-GAN implementation, with training implementation described using snip code.

Chapter Six: The obtained testing result from Temporal Cycle consistency for a video-to-video translation model is presented and Compare with the other related work in order to have the best judgment.

Chapter Seven: Concludes the research and provides directions for possible future work.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORKS

2.1. Introduction

In the early days, the software was hard-coded with commands and rules represented in a mathematical formula. Those methods short-handed as they attempt to simulate sophisticated real-world situations and data. To solve this scarcity another software development paradigm emerges which asks whether the machine can understand as humans by having cognitive taught called Artificial intelligence or AI. AI has made a lot of improvements in recent years, allowing AI applications completely capable of gathering and extracting data to learn from the knowledge sequence called machine learning [17].

Machine learning (ML) is a vast research area with diverse learning capacities and it keeps to grow. From different learning capabilities of machine learning, unsupervised learning is one of them. This learning type is the task of clustering unorganized data to organize them based on the information they composed. Another kind of machine learning is supervised learning, unlike the unsupervised approach, for every data input there is a corresponding output label (x, y), the network task is to learn how to map from input x to it label y in testing time (R.B. data is paired dataset). Semi-supervised learning is another kind of machine learning approach when some section data is labeled and the rest section is unlabeled. Even ML plays very tremendous work, in recent days, but it still fails to process complex data like image and video. So as to work with such complex data types Deep Learning is an alternative which subfield of machine learning.

Deep learning (DL) is another mechanism of learning from data in sequential filter layers while the previous approaches learn from data representation. DL filter present data information one in terms of another which build a hierarchical ordered of features. These features enable us to extract high-level features. The first emerged deep learning was the artificial neural network ANN [18]. Convolutional Neural Network (CNN) was introduced by Yann LeCun [19] in 1989 to recognize handwritten digits but the lack of a large dataset and low computing capability at a time limit its popularity. CNN bloom after Alex-Net [20] notch a significant win in the ImageNet contest in 2012 using the CNN image classifier network. Alex et al work opens many scientists and researchers eye to the power of deep

learning. Today high-performance models and networks are designed for face detection, object classification, and recognition.

CNN models score a high success on classification models, classification models are tasked to predict label y for given input x. In other sense, Generative models are the Multiplicative reciprocal of classification networks. The most impressive new emerging deep learning approaches are Generative models. these models are trained to learn the essence of the data distribution to generate fake samples that are similar to real ones. GANs application is a very broad image-to-image translation, and video-generation can be some examples of its application area.

GAN introduced by Goodfellow et.al in 2014 [11] the model consists of two stand-alone DL networks named Generator and Discriminator model which have fight one another in an adversarial relationship[1]. Hence the discriminator plans to distinguish fake samples $p(z)$ from the real input data $pdata(x)$.

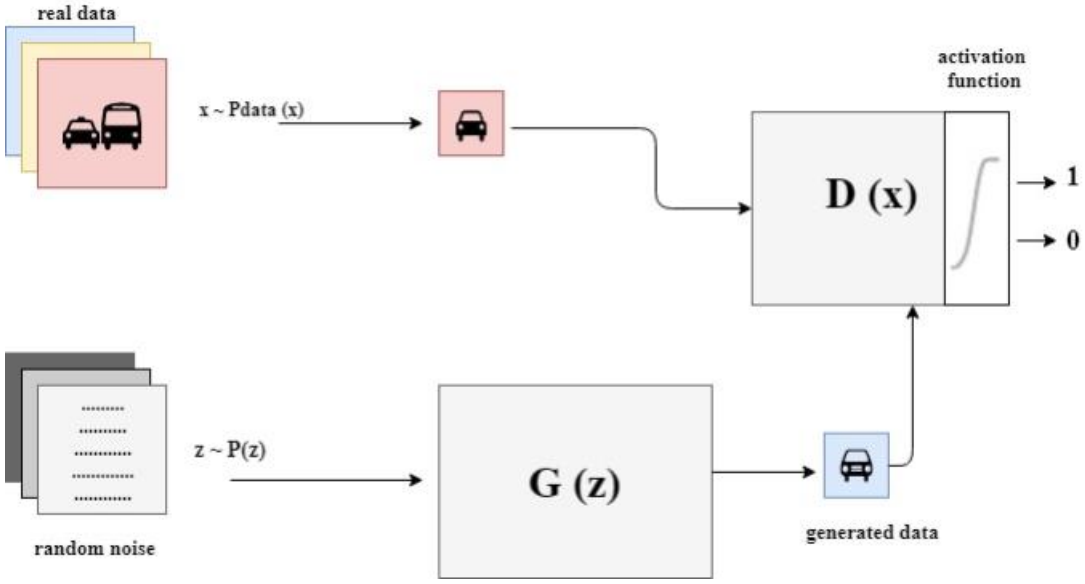


Figure 2-1 GAN framework structure GAN: Discriminator $D(x)$ and Generator $G(z)$

[1] Some authors see GAN in other perspective rather adversarial: collaboration of two networks to mimic a give real data distribution.

Oppositely the generator network is all about generative fake samples as good as the real ones. During training, both models update their weight to improve until Nash-equilibrium is reached. The next section discusses the detail inside of GAN.

2.2. Inside GAN

Let's see the detail inside of GAN. As discuss GAN in the previous section, GAN consists of two independent networks Generator and Discriminator as shown in Figure 2-1, which are represented by differentiable functions concerning each network's input and parameters. The discriminator is defined by a function $D(x)$ where x (observed variable) is the input which is a real dataset. $D(x)$ 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 $G(z)$ accepts 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 fed into the discriminator network $D(x)$, which attempts 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 formulated as [11]:

$$\mathrm{L_{adv}} = \min_{G} \max_{D} E_{x\epsilon X}(\log D(x)) + E_{x\epsilon Z}(\log(1 - D(G(z)))) \qquad \textit{eq.(2.1)}$$

The generator can be seen as a kind of reverse CNN. It takes an z-dimensional vector of noise and upsamples it to an image using transposed convolution(transconv) to be specific transconv is a convolutional upsampling. Conceptually, the discriminator in GAN provides guidance to the generator on what images to create implicitly in the training process. Now we can discuss how to train the GAN network.

2.2.1. GAN Training

Machine learning is all about generalization in which the model learns from real-world examples so that it can predict the test set accurately. No difference for GAN, training is all about the process of learning to mimic the real dataset samples. Unlike many deep learning models, training is a bit tricky [21] so let us dive into it. However, before that, let sees an adversarial conflict between discriminator and generator.

The Discriminator's goal is to be as precise possible (binary classification). For the real examples $x, D(x)$ seeks to get as real as possible to 1 (label for the positive class). Meaning $x_{fake}, D(x_{fake})$ attempts to converge 0 as possible (label for the negative class).

The Generator's goal is the reverse. It tries to find a way to fool the discriminator by producing fake example x_{fake} that are alike from the real data in the training dataset. Mathematically, the Generator strives to produce fake examples x_{fake} such that $D(x_{fake})$ is as close to 1 as possible.

Table 2-1 Generator goal vs discriminator goal

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

Now let's back to GAN and see pseudocode for training GAN (*R.B* its iterative process)

I. 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: $G(z) = x_{fake}$.
   c. Compute the classification losses for $D(x)$ and $D(x_{fake})$, and backpropagate the total error to update $\theta^{(D)}$ to minimize the classification loss.

II. Train the Generator:
   a. Take a mini-batch of random noise vectors z and generate a mini-batch of fake examples: $G(z) = x_{fake}$.
   b. Compute the classification loss for $D(x_{fake})$, and backpropagate the loss to update $\theta^{(G)}$ to maximize the classification loss.

Unlike other deep learning training notice that in step 1, the Generator's parameters are not updated intact while training the Discriminator. Similarly, the Discriminator's parameters intact while in the Generator session[22]. The reason GAN allows updates only to the biases and weights of the network being trained is to isolate all deviations to only the constraints that are under the network's control. This guarantees that separately generator and discriminator get relevant signals about the updates to make, without interacting from the other's updates meaning each two players taking turns to update their weights. This process continues until the Nash equilibrium.

GAN is based on the adversarial game between two networks. In short, if the Generator wins the Discriminator loses and vice versa if the other wins. In-game theory, the Generative network converges when the generator and the discriminator hit the Nash equilibrium. This is the optimum point for the GAN loss min-max function eq(2.1). Regarding GAN at Nash equilibrium discriminator no longer able to distinguish between real and fake samples so it randomly classifies ($accuracy = 50\%$).

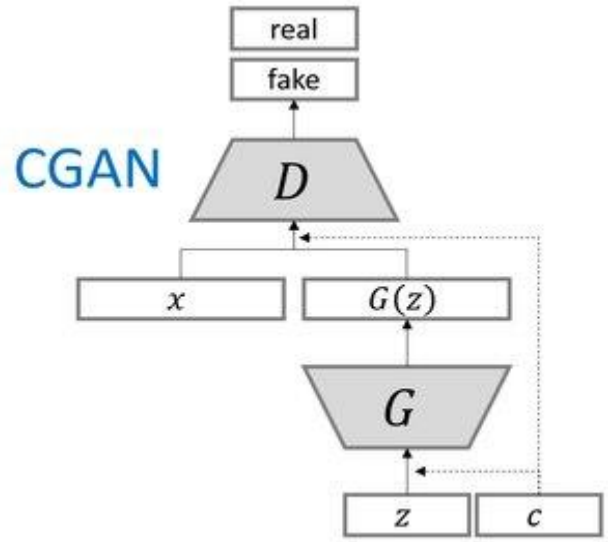


Figure 2-2 Conditional GAN Architecture

Source: Adapted from[22]

2.2.2. Conditional GAN

Even though GAN models are able to produce new random possible examples for sample data, there is no way to identify the types of images generated. However, in order to imitate

the original data set and images, the network seeks to define the composite relation between the latent space inputs in the generator.[11], [23].

Mirza et al. propose The conditional generative adversarial network, or cGAN [12] for short, which is a type of GAN that involves the conditional generation of images by a generator model. Image generation can be based on the label of the class [2]. It requires the Generator network to produce only the target class of frames of a given form by a conditional variable. The conditional variable C is fade to the generator and discriminator networks as shown in Figure 2-2 above. This work unlocks opportunities for many fascinating research topic like image-to-image translation, style transfer and video retargeting [1], [3]. The next section will discourse about Image-to-image translation.

2.3. Image-to-Image Translation

Let start with Abto software AI software company from Europe say about the image-to-image transfer when they announce their research product *"you may hear A magician can make his trick with just a wave of a magic wand, but its old news. Here in our lab, our engineers can make their magic with just one click! Interested, how the same winter landscape would look in summer"* [24] I was wondering too, winter to summer absolutely fascinating.

Recent advancements in GANs [11] empowers image-to-image transfer models to create realistically looking [2], [3], [25], [26] adapted image (Figure 2-4 B show image-to-image transfer from sunny to rainy). The image-to-image translation aims to learn a mapping function between the input image and out image in different domains. When we talk about Image-to-Image basically learning involves the precise modification of an image while preserving contain information and it requires large datasets of paired images that are complex to prepare, meaning the dataset should contain images that are one to one correspondence as shown in Figure 2-4. The primary difficulty in the image-to-image translation is they need paired data set for training, but in reality, doing so is very expensive and not scalable, but some work achieves good results. Pix2Pix[3] is one of them, which is a conditional Generative model by Isola et al. train in a supervised manner using a paired

[2] conditioning variable *C* could be any type of information. Like Image, tabular information or….

dataset that fits into a supervised image-to-image translation. Pix2Pix as the name indicates it learn to map pixel from the first image to the second one.

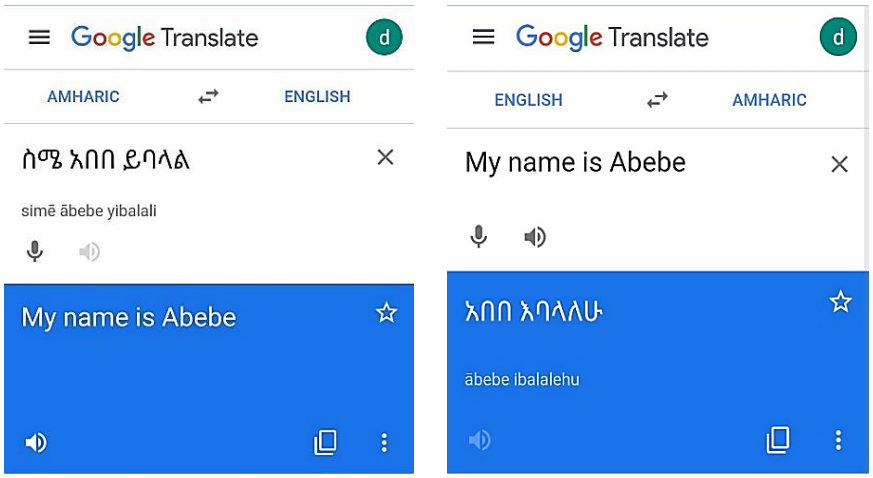


Figure 2-3 Amharic to English language translation using google translator.

Because in reality pair datasets are very rare and expensive Zhu et al. [2] came up with Cycle GAN which was invented to learn bidirectional mapping in the absence of paired training data via Cycle consistency loss. Cycle Consistency loss utilizes to learn transformation between two domains in a frontward and backward fashion. Cycle consistency constraint is not a new idea; in fact, it is very old news in natural language processing. The following example gives a simple illustration. Assume using language translation from Amharic (አማርኛ) to English in both directions. When the user input "ስም አበበ ይባላል።" the model should generate "My name is Abebe" perhaps if the user translates "My name is Abebe" to Amharic back again it should generate the original text "ስም አበበ ይባላል". Meaning the difference between the original text and regenerated text should be minimum. Google translator has been used to demonstrate this example, as shown in Figure 2-3. The loss of the network in this instance would be the difference between "ስም አበበ ይባላል" and "አበበ እባላለሁ". The general architecture of Cycle GAN contains two generators and discriminators for each domain. Where one generator translates from domain A to B while the others do the reverse. Let us see it in a bit detail using Table 2-2.

Table 2-2 Cycle GAN generator and discriminator operation.

G_{AB}	Translate from A to B	$x -> \tilde{x}$
G_{BA}	Translate from A to B	$y -> \tilde{y}$
D_A	Classify real A and fake $\tilde{B}$	1 for x, 0 for $\tilde{x}$
D_B	Classify real B and fake $\tilde{A}$	1 for y, 0 for $\tilde{y}$

x and y are a real image from domain A and B respectively.

while $\tilde{x}$ and $\tilde{y}$ are generated images from G_{A_B} and G_{B_A} respectively. R.B $\tilde{x}$ is in domain B where $\tilde{y}$ in domain A

The loss function of the network could be formulated as: $min \sum \left|\left| x - G_{A_B}\left(G_{B_A}(x)\right)\right|\right|$ meaning translate a given image are x and reconstructed image $\tilde{\tilde{x}}$ the difference should be the minimum ($x \approx \tilde{\tilde{x}}$) . $\tilde{\tilde{x}}$ Input image x translated to another domain and retranslated back to its original domain. Ahead of image transformation across domain video-to-video translation is an additional extension.

Figure 2-4 (A) pair shoe dataset sample from Pix2pix, (B) Sunny to Rainy translation
Source: Adapted from [3]

2.4. Video-to-Video Translation

Video-to-video translation is a natural extension of an image-to-image translation. Translating video points toward learning the appearance of objects in a scene and realistic motion movement between successive frames. A straightforward way to video-to-video translation carry out the image-to-image translation in each frame of input videos without considering those frames that have a relation between them. This approach is non-trivial

since this is key issues that underlie the flickering [4], [16] effect in the output video. To overcome the flickering effect, Chen et al. [4] consider temporal information along with spatial information. Specifically, they exploit previous frame optical flow to warp the current frame towards imposing temporal constraints. Let see what temporal information and different mechanism to extract. But before that it worth a brief discussion about the problem in current approaches.

2.5. Problems in Translation Networks

As discussed in previous sections, video-to-video translation is an immediate extension of image translation, so every limitation of image translation is extended correspondingly. Furthermore, Object disappearance, Object dislocates, and Artifacts [9], [27]–[31] are the most common problems for video translation.

Let say there are two generators G_{AB} and $G_{BA}$to translate from one domain to another domain and two discriminators D_A and D_B, where G_{AB} trained to translate from A to B and $G_{AB}$from B to A. and discriminators $D_A$and D_B to classify between real and fake in both domains. Video X and Y are sample videos from respective domain A and B.

$$X = \{x_1, x_2, x_3 \dots , x_n\} \text{ where } x_i \text{ are the i}^{\text{th}} \text{ frame of video } X.$$

Each frame may contain various objects. $\{O_{x_t}^1, O_{x_t}^2, O_{x_t}^3 \dots O_{x_t}^n\} \epsilon x_t$ Objects in a frame can be seen as a group of connected pixels.

- » Artifacts: An image frame artifact is any element that occurs in the picture that is not present in the initial picture set.
- » Tide Spatially to the input: The optimizer is required to learn a solution that is strongly similar to the input due to the reconstruction loss on the input itself.
- » Object disappearance: is a problem object O_i in a given video frame x_t in domain A shall also appear in translated appear $\widetilde{x_t}$ in another domain image. meaning if a car appears in x_t is should also appear in $\widetilde{x_t}$. Mathematically,

$$if \{O_{x_t}^i\} \epsilon x_t, then \{O_{\widetilde{x_1}}^i\} \epsilon \widetilde{x_t} where: \widetilde{x_t} = G_{AB}(x_t)$$

» Object dislocation[3]: happen when an object O_i in frame x_t from a domain A changes its position when translated in $\widetilde{x_t}$ domain B. Object dislocation also can be seen as an abrupt object movement. Mathematically,

$$if\ \{O^i_{x_t}\} \epsilon\ x_t \& \ locate\ [(a1, b1) - (a2{:}\ b2)]\ then, \{O^i_{\widetilde{x_1}}\} \epsilon\ \widetilde{x_1}\ should\ locate\ in\ [(a1, b1) - (a2{:}\ b2)]$$

$$where{:}\ \widetilde{x_t} = G_{AB}(x_t), a\ \&\ b\ are\ spatial\ location\ in\ x\ and\ y\ direction$$

The problems described above are appropriate for the problem of translation, where Spatio-temporal information is not leveraged fully.

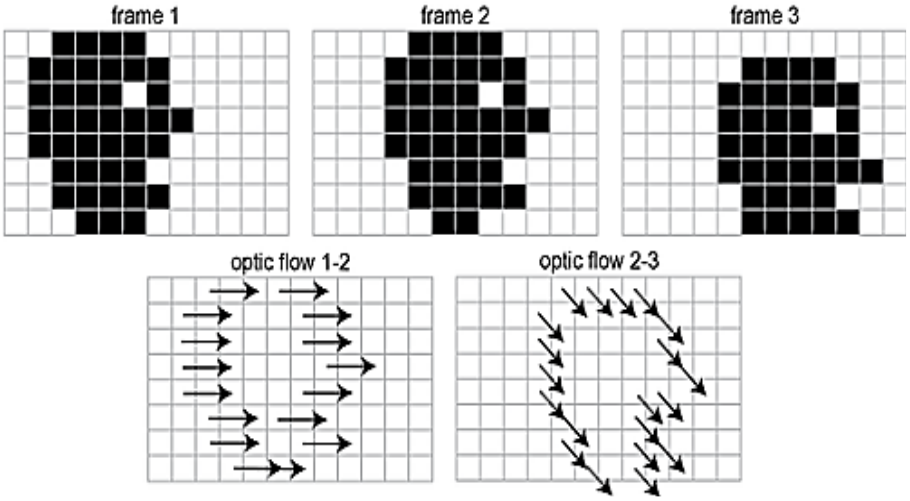


Figure 2-5 Detection of the optical flow in three consecutive images.

Source: Adapted from [32]

2.6. Temporal Information

Temporal refers to time-domain, wherein a computer vision case, it can be seen as a relation between the sequence of frames in the video, while Spatial refers to RGB space. Spatiotemporal or Spatio-temporal is used in the study of information as data is gathered over time and space. Straight forward approaches generally fail because they cannot consider both domains. Temporal information for video can describe a phenomenon in a

[3] Object dislocation in a situation like face to face translation wouldn't be a question.

particular pixel location with position change in time. For a video-to-video translation, there are various options to represent motion information. The next section would discuss those topics for illustration. The extraction of time knowledge can be split into two separate groups. One explicit temporal information extraction: this kind of network operates in such a way that the model extracts temporal information directly, such as optical flow and pose estimation. Then the model imposes temporal information on the generated frame. The other tacit does not explicitly collect temporal knowledge but aims to learn temporal dimension through specially modeled learning layers of the model. Examples could be 3D Conv-nets, RNNs, and temporal constraint models. Indeed, some of the works have been done to blend the above techniques, such as Park et al. [33].

2.6.1. Optical Flow

Optical flow is the change of structured pixels with specific intensity in successive images, or in other words, optical flow is the motion of objects among successive frames, caused by the relative movement among the object and camera. This make optical flow an ideal for encoding temporal information [1], [33], [34].

Figure 2-5 shows three sequence images, and in the next row shows the Optic flow between the modification in these images over a vector field. The research underlines the precise, pixel-wise estimation of optical flow, which is a computationally challenging task.

Nowadays, better computational resources and Recent advancements in Deep learning enable researchers to estimate optical flow [35], [36]. Generally, such approaches take two video frames as input to output the optical flow (color-coded image), which may be expressed as $(u, v) = f(x_{t-1}, x_t)$ where u is the motion in the x direction, v is the motion in the y direction, and f is a neural network that takes in two consecutive frames x_{t-1} (frame at time $= t - 1$) and x_t (frame at time $= t$) as input.

Computing optical flow with deep neural networks [35], [36] requires vast amounts of training data which is principally hard to obtain. This is because marking optical flow video footage requires a detailed finding of the precise motion of each point of a frame to the precision of the subpixels. To address the issue of labeling training data, many research works, [35]–[37] used computer graphics to simulate massive realistic worlds. Since virtual worlds are produced by complex computer instruction, the motion of each and every point

of an image in a video sequence is known. Some examples of such include MPI-Sintel [38], an open-source CGI movie with optical flow labeling rendered for numerous sequences, and Flying Chairs [37], a dataset of numerous chairs hovering around random backgrounds both generated from the virtual world using Computer Graphics.

2.6.2. Pose Estimation

Human pose estimation can be framed as the problem on the localization of key points like eye, nose, elbows, wrists, etc. in images or videos frequently referred to as human joints. It is also known as the exploration of the overt position of all articulated poses in space. Basically, pose estimation translate used in transferring motion from a deriving video to derived object in a video.

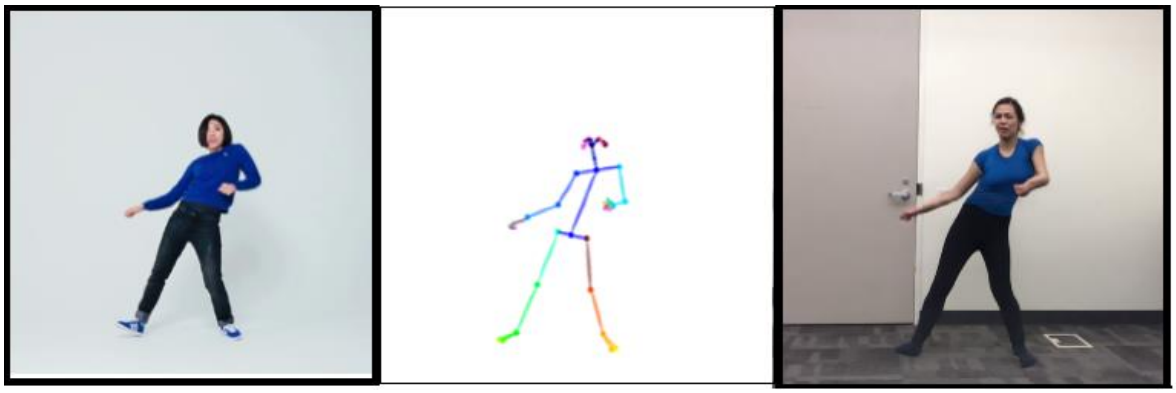


Figure 2-6 Pose extraction to transfer pose.

(left) driving video, (middle) extracted pose from driving video and, (right) generated video using estimated pose.

Source: Adapted from[39]

Mainly human pose estimation is used in transferring motion from one person to another as used [39][40], to transfer motion between different domain videos specifically for animating static image by driving motion as shown in figure 2-6 [41] and facial expression transfer [42] between source and the target person.

There are two types of pose estimation classical and deep learning; the former is all about represents an object by a group of "parts" organized in a deformable configuration, and Later, ConvNets was commonly embraced as their core building block. They largely replace hand-crafted features & graphical models perhaps this approach has returned drastic advances on standard benchmarks.

2.6.3. 3D Convolutional Tensor

The 3D convolutional tensor mechanism is one of the orthodox methods, which basically does not consider temporal information explicitly. Since it considers presenting a video scene [27] as a 3-dimensional tensor meaning it takes the whole video as input, and the network eventually learns the relation between consecutive frames to preserve temporal consistency implicitly. In due course, this approach is not used frequently because of two fundamental reasons requires a high-efficiency machine, and the network becomes an entirely black box, which means hard to tune parameters principally done in training deep learning models.

2.6.4. Recurrent Temporal

Recurrent neural networks or RNNs are a type of neural network inherently ideal for analyzing data from time-series, and other sequential figures make it ideal for video analysis. Possibly it overcomes the black-box nature of 3D Conv nets by adding an additional parameter to tune the network. Recent works consider using LSTM (Long Short Time Memory.) which takes into account all previous frames as an input to minimize temporal residual error [34].

2.7. Related Works

In the previous section, the temporal information (motion information) extraction mechanism is discussed. However, since video consists of both temporal and spatial information, let's discuss various related works mechanisms to get the advantage over an early approach (spatial only). Hence instead of applying Spatial information only (meaning split a video as a sequence of images and apply for domain transfer on each, then stitch them back) by assuming frame constraint has no relation [2]. This approach is non-trivial since the key issues which motivate the problem listed in Section 2.6. the results output video [4], [16].

To overcome the flickering effect, Chen et al. [4] consider temporal information along with spatial information. Specifically, they exploit previous frame optical flow to warp the current frame to impose temporal constraints, but this paradigm prone to occlusion and fast illumination change (since optical flow does not consider newly introduced pixels in the given frame scene). Unsupervised Video-to-Video Translation paper by Bashkirova et al.

[27] models temporal information using a 3D convolutional layer embedded on Cycle-GAN, their result shows better color preservation and less artifact compared to Cycle-GAN but, the model black-box nature makes it hard to train. Nevertheless, the result shows a lack of robustness to different video lengths.

Video-to-Video Translation with Global Temporal Consistency [34] by Wei et al. further extend the optical flow frame warping network, the authors present a mechanism focusing on the video-level consistency by residual error based on two-channel discriminator to minimize the total Mean absolute (L1) distance between the optical flow map of consecutive frames eventually this approach failed on longer video and result failed in fast motion videos since the network constrained to minimize temporal difference along with the whole video.

Another fine work by Chen et al. [8] MoCycle-GAN introduces motion guided Cycle GAN to transfer estimated motion between domains. This work is explicitly designed in two way network by splitting spatial, and temporal information separately. The spatial network is Cycle GAN and the temporal network is a motion translation network. This work also relives the temporal cycle constraint for motion reconstruction. Even if an explicit motion translation network is a blessing, the model parameter increased enormously. Another pitfall of this work is the network relay on cycle constraint to content translation.

The current state of artwork [1] ReCycle-GAN further extends cycle consistency constraint by intercorporate it with a temporal predictor network to predict over spatiotemporal predictor. though directly synthesize future frame via temporal predictor to preserve temporal continuity is still under constraint. The under constraint problem in ReCycle GAN work result in undesirable motion on generated video.

Another recent quality work by Park et al. [33] proposes an optical flow warping ground truth and content loss on frame mechanism to guarantee the consistency to overcome the temporal flickering and motion inconsistency between frames. Temporal flow consistency is another count of this work, which basically excellent if the two domains are similar in nature, but has no much impact on slightly different motion videos.

2.8. Summary of Related Work

the following table illustrates a summary of previous works on the video-to-video translation[4].

Table 2-3 Previous works summary on the video-to-video translation.

Related papers	***Dataset (Data collection)***	***Architecture***	***Temporal information modeling.***	***Temporal constraint applied on***	***Evaluation metrics used***	***Limitation***
Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks[2]	Cityscapes, Horse to Zebra, Apple to Orange, Summer to Winter Yosemite.	Cycle Consistency Constraint.	No temporal Information is considered.	-	FCN, IS	Framewise image-to-image translation.
Unsupervised Video-to-Video Translation [27]	Volumetric MNIST, GTA segment to video and MRI-to-CT	3D Cycle-GAN	The network implicitly learns from input video (3D-Conv- net)	-	Human evaluation, pixel accuracy, and L2 error between original and retranslated image	3D tensor fails for temporal learning consistency between frames, fixed-length video.

[4] These present papers are only substantial papers that directly relate to thesis work, and all are from 2017 onwards.

Video-to-Video Translation with Global Temporal Consistency[34]	DAVIS 2017	RNN based Cycle-GAN, and RNN based Discriminator for global temporal consistency	Optical flow, temporal residual error minimizer	Generator + Discriminator Network	Peak Signal to Noise Ratio, Region Similarity, and Contour Accuracy	Complex architecture hard to train. Inappropriate for in videos contain fast object motion. Not work for long videos.
MoCycle-GAN: Unpaired Video-to-Video Translation MoCycle-GAN [8]	Flower video and viper dataset	Cycle-GAN with motion translator-based motion cycle consistency	Optical flow with motion translator network	Generator Network	Human evaluation, IoU, pixel accuracy, Average class accuracy	Explicit motion translator, and no content translation
Recycle-GAN: Unsupervised Video Retargeting: ReCycle-GAN [1]	Viper, face, and flower datasets (more than 10,000 images)	Cycle-GAN with recurrent temporal predictor	Recurrent temporal predictor (Pix2Pix)	Generator Network	Human evaluation, IoU, pixel accuracy, Average class accuracy, IS	Temporal predictor fails to correctly predict, and no content translation

Preserving Semantic and Temporal Consistency for Unpaired Video-to-Video Translation [33]	Viper dataset	Cycle-GAN with flow estimator network and consistency warping network	optical flow base temporal fuse with spatial for improving occlusions problem	Generator Network, Use [43] to further reduce the Temporal warping error.	mIoU, fwIoU, and pixel accuracy	Input domain videos shall have very similar content.

As shown in the above table, researchers design various architectures in previous work to learn a mapping from a domain to domain transfer in an unsupervised manner. However, those works count on Cycle-GAN to contain translation, hence cycle loss does not assure semantically coherence outputs. As a result, existing video to video translation approaches are prone to disappearance and artifacts problems in generated video. On another hand, early works focus only on improving constraints on generator networks, which shorthanded to appropriate consideration of temporal information that can modeled using discriminator network. This lead object dislocation problem in generated videos. Therefore, to mitigate the above issues imbedding content translation constraint that is responsible for translating is essential. Even more the discriminator network must be exploited regarding temporal modeling to improve the translation network performance.

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1. Chapter Overview

The thesis research questions were outlined in Chapter one, along with a mathematical formulation and an overview of the method used to investigate the associated plans. This chapter provides further details of the methodology, dataset, and experimental metrics to answer the research questions.

The following approaches and procedures are used to accomplish the goals of this study.

3.2. Dataset

This study uses a deep learning approach to solve video-to-video translation problems in an unsupervised manner, so data is an essential part of the study. Images of a face (Obama-trump), Viper and, flowers are used for both training and testing stages as used in [1]. In addition to inference the result of this thesis, a local dataset called አዲስ (Adiss) has been collected.

- ***Obama-Trump***: is a dataset for style transfer and video retargeting. This dataset contains a sequence of images of Obama and Trump making an interview (though at a different time and talking about completely different topics). Each frame is 256 x 256 and about 8617 video frames are included.
- ***Flower Dataset***: is a recently released dataset for video translation. This dataset contains the time-lapse videos which depict the flourishing or fading of several flowers but lacking any synchronization. The resolution of the respective videos is 256 × 256—this work use flower-to-flower for domain transfer between dissimilar types of flowers.
- ***Viper Dataset***: is a prevalent visual perception benchmark to facilitate both low-level and high-level vision tasks -semantic segmentation and optical flow. It comprises videos from a realistic virtual world game (i.e. GTA V), which are composed while driving, riding, and walking in various ambient circumstances (day, sunset, snow, rain, and night). Each frame (resolution: 1920 × 1080) is annotated with pix-level labels, for video-to-labels and labels-to-video, viper could be a benchmark for

evaluating the translations between videos and segmentation label maps, and day ↔ sunset. For this study, the frame resolution is Demote to (resolution: 256 × 256).

» ***አዲስ (Adiss) Dataset***: is a local dataset for video retargeting containing two very popular politicians Prime minister Abiy Ahmed and Debretsion Gebremichael making an interview and press briefing. The frame size is 256 x 256, and around 5000 video frames are included. (a 3-minute video from the internet has been used to make the dataset)

Table 3-1 Training dataset Sample (from Flower, Obama – Trump, and Adiss datasets)

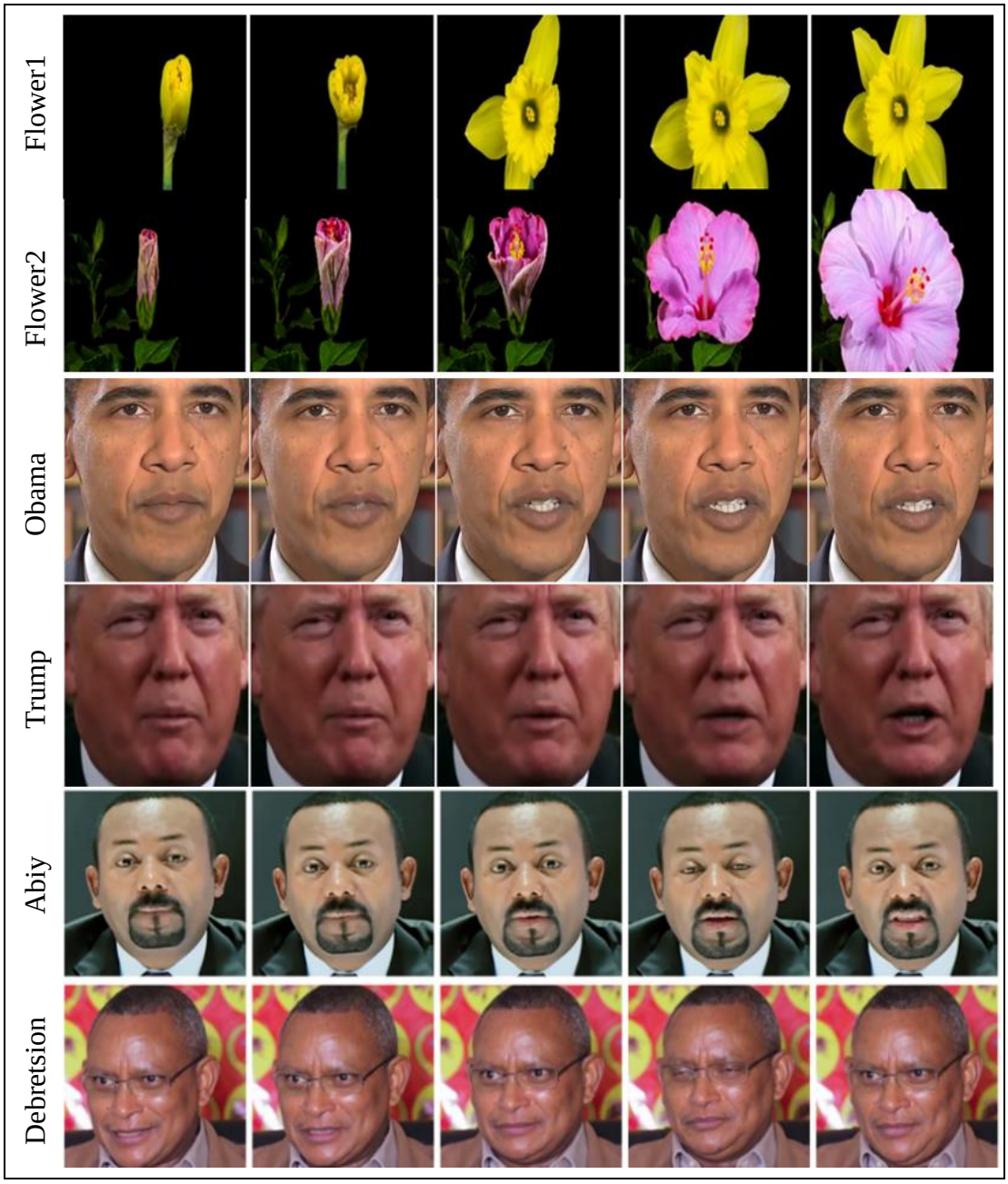
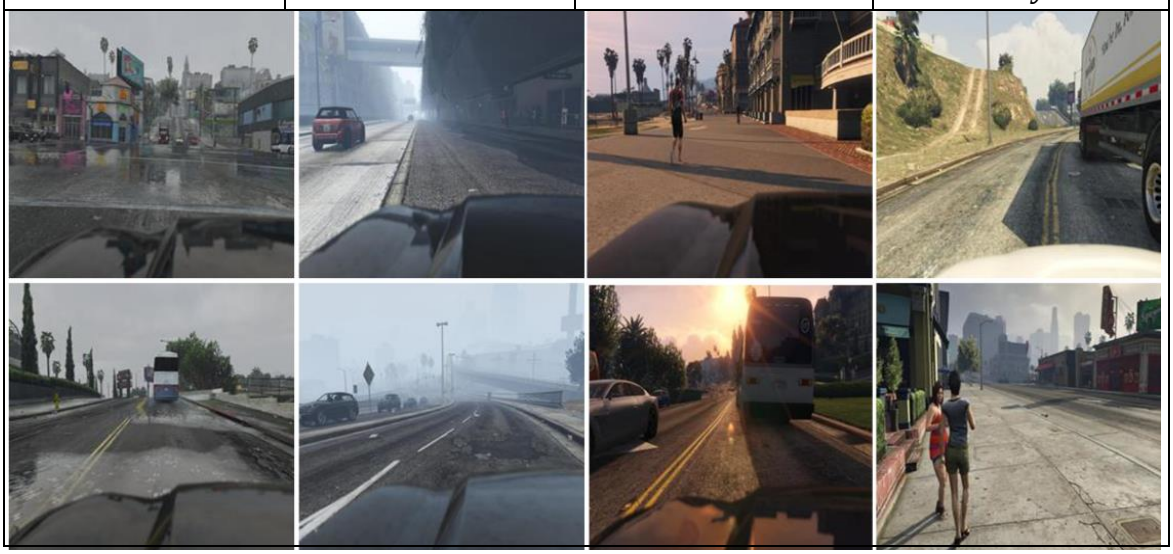


Table 3-2 Viper dataset sample examples

Rain	***Snow***	***Sunset***	***Day***
			

3.3. Development Tools

For this research, numerous types of development tools are used to design and implement the proposed thesis work. The development tools section gives a description and justification of these development tools. 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.

3.3.1. Design Tools

Design tools are mediums that are used for the creation, presentation, and interpretation of design concepts. Edraw Max [44] is used to design in the proposed system. It is a lightweight and powerful graphic design tool for creating professional-looking flowcharts, network diagrams, flowchart diagrams, and others. This tool is selected because [44].

- » It has lots of high-quality shapes, example, and template,
- » Easily visualizes complicated details through a broad range of graphics.
- » It works well with MS Office.

3.3.2. Prototype Development Framework

TensorFlow: TensorFlow is an open-source software library optimized for maximum-performance numerical modeling and processing. Its modular architecture can be easily

implemented on a range of platforms such as Central Processing Units (CPUs), Graphical Processing Units (GPUs), Tensor Processing Units (TPUs). It can also be mounted on personal computers, clusters, handheld devices, and edge devices. TensorFlow supports artificial learning, deep learning, and versatile numerical computing [45] The following diagram displays the power score of the deep learning system based on application, popularity, and interest [46].

The following diagram demonstrates the power score of the deep learning system based on application and popularity. As shown in Figure 3-1, TensorFlow is by far the most used and popular deep learning framework.

- Makes fast and rapid prototyping;
- Embraces all Convolution networks and recurrent networks, as well as variations of each.
- User-friendly, modular, and extensible.
- It can run efficiently on GPU or CPU.

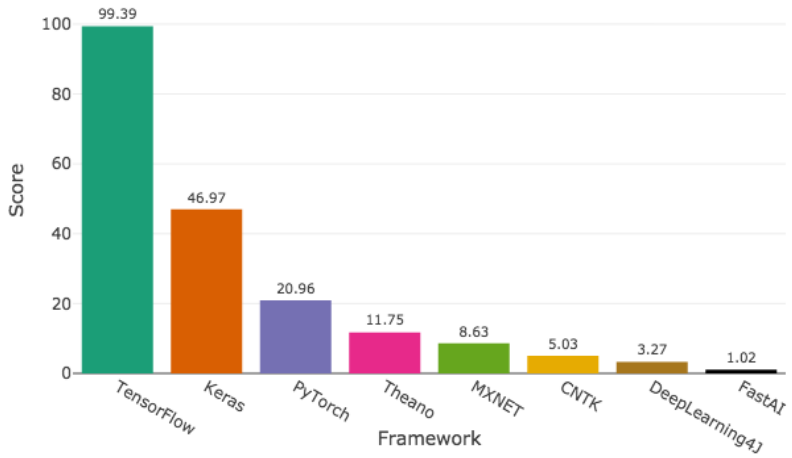


Figure 3-1 Deep Learning Framework comparison.

Source: Adapted from [47]

OpenCV: OpenCV is an open-source computer vision software library intended to provide a shared infrastructure for image processing and computer vision applications [48]. It has Python, Java, C++, and MATLAB interfaces and supports nearly any operating system as well. OpenCV was developed for image processing, meaning that and feature and data structure was developed with the image processing engineers in mind.

MATLAB Deep Network Designer: MATLAB deep network designer [49] is an application developed by MATLAB which developed for easy design, visualize, and train deep learning networks using drag & drop simple user interactive mechanism. This tool is a relief for AI developers, especially for complex network deep architectures and GAN networks. This even further helps developers to track and debug errors on the early design stage.

3.4. Baseline Works

To validate this study model's effectiveness, the implemented model is equated with models that dwell on translating video with GAN. Since architecture is based on Recycle-GAN and takes input as unpaired video data, Cycle-GAN [2] and Recycle-GAN [1] are taken as baselines for the experiments.

- Cycle-GAN [2] converts images using two generators, with the assumption of cycle consistency. This work uses it to translate the video frames and make comparisons in order to understand the Spatio-temporal constraint effect better.
- Recycle-GAN [1] uses two generators and two predictors for video retargeting. It puts forward a recycle loss to work with cycle loss and recurrent loss for content conversion and style preservation, considering the temporal detail.

The purpose of contrasts Cycle-GAN and ReCycle-GAN is to show the substantial improvements achieved by our model in terms of spatial-temporal knowledge.

3.5. Feature Extraction Network

Several state-of-the-art Deep CNN based architectures have been proposed over the last decades for the classification of images. These different modern CNN based feature extraction architecture include ResNet[50], Inception[51], Xception [52], EfficientNet[53], and others.

Figure 3-2 shows the top1 accuracy vs the number of parameters. The x-axis represents the number of trainable parameters. EfficientNet-B7. Another architectures like VGG have more than 155 million parameters and ResNet has round 60 million. EfficientNet-B7 has a smaller number of parameters and operations compared to most architectures, which makes it able to run fast on different devices that have less computing. Perhaps, Other architectures like ResNet-50 have a fewer number of parameters and operations as shown in the figure above, but their accuracy fails as well. Based on the above observation, EfficientNet-B7[53]

is used in this research for feature extraction in the task of computing feature maps of a given image.

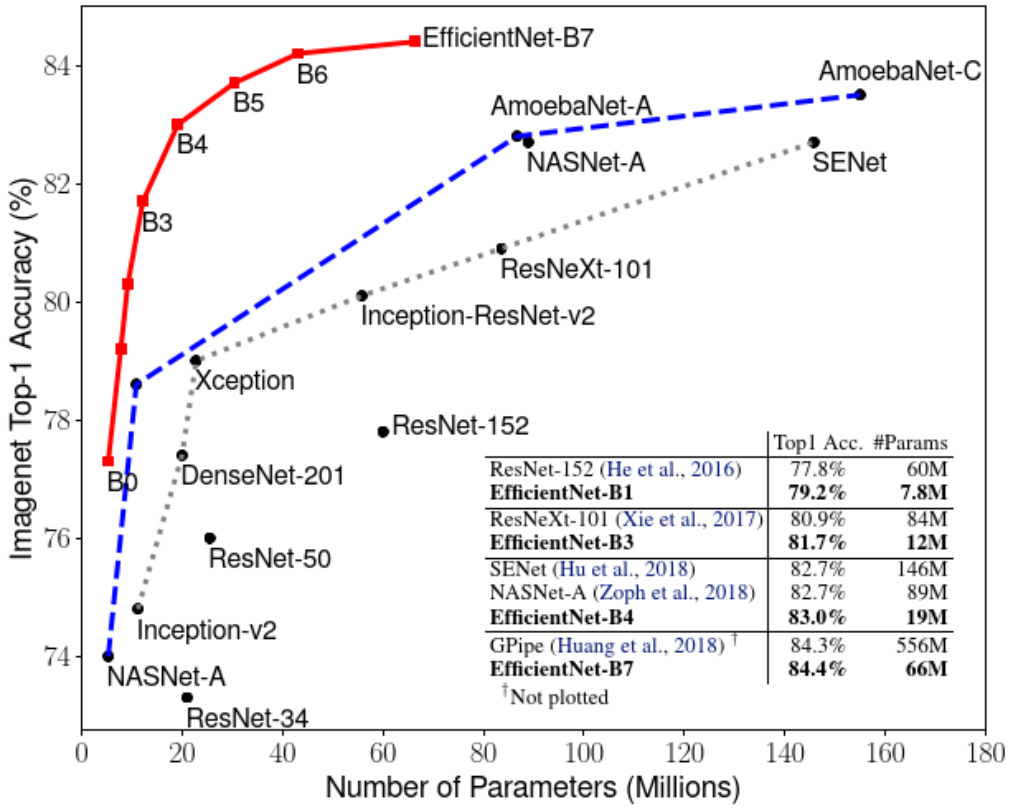


Figure 3-2 Benchmark Analysis on EfficientNet-B7

Source: Adapted from[53]

3.6. Temporal Discriminator Network

Temporal information is a motion relation between the sequence of frames as discussed in chapter two. Recent works do not consider the power of discriminator networks regarding temporal information. An ablation study has been conducted to decide the number of frames the discriminator should consider for classifying between fake and real. The detailed results are found in Appendix A.

The study performed on temporal discriminator with one frame, with two frames, with three frames, with four frames, and with five frames but the training fails with five frames because of memory insufficiency problem. From the result (AEE), a Temporal discriminator with three input frames performs well than comparative models. So, in this study temporal discriminator with three frames has been used.

3.7. Evaluation Methods

The result has been analyzed to describe the video-to-video translation models performance on a test data set. The dataset is split into different training and testing set using different test sizes. The algorithm is evaluated using the test set. One big problem with GANs is that there is no robust way to beyond visual inspection. The next subsection present is a qualitative analysis and a quantitative metric to evaluate this work.

3.7.1. Human Evaluation Study

This evaluation method uses 10 to 15 volunteers to assess whether the given video is real or fake after he/she sees real and fake videos to determine whether or not the generated data is any good. The average score value is evaluated as per the figure of entities. Motivated by the ReCycle-GAN human evaluation study, this thesis work uses two protocols. First, the input video, synthesized videos of other approaches, and this work result are seen simultaneously for the participants.

They are asked which one has higher consistency, better smoothness, and better continuity between video frame sequences. Second, only synthesized fake videos are seen simultaneously for the participants, and they are asked which one has higher consistency and looks more natural Translation. The higher the human evaluation score means, the better the performance of the network.

3.7.2. Inception Score

The inception score is a commonly used evaluating algorithm for GANs. It uses a pre-trained inception V3 network (trained on ImageNet) to extract the features of both generated and real images. The inception score [54] for short IS, measures the variety and the quality of the created images. The superiority of the model is good if it has a high inception score.

$$IS(G) = exp(E_{x \sim p_g} D_{KL}(p(y|x)||P(y))) \quad \text{eq.(3.1)}$$

$x \sim p_g$ indicates that x is an image sampled from p_g, $D_{KL}(p||q)$ is KL divergence between the distributions p and q, $p(y|x)$ is the conditional class distribution. KL divergence measure of how one probability distribution is different from a second, reference probability distribution

CHAPTER FOUR

4. PROPOSED LEARNING FUNCTION

4.1. Chapter Overview

This chapter presents the proposed solution to video-to-video translation problems for improving temporal consistency. The generated video should be able to have better consistency between a succession of frames. This chapter can be ideally portioned into three major sections; the first introduces model Architecture to translate a given domain image into another domain—the second deals with Network optimizing loss functions. The last explains training Pseudocode to train the model.

4.2. Model Architecture

The model architecture has directly adopted from the architecture defined in *"Unpaired Image-to Image Translation using Cycle-Consistent Adversarial Networks"* [2] and *"Recycle-GAN: Unsupervised Video Retargeting"* [1] for learning domain translation.

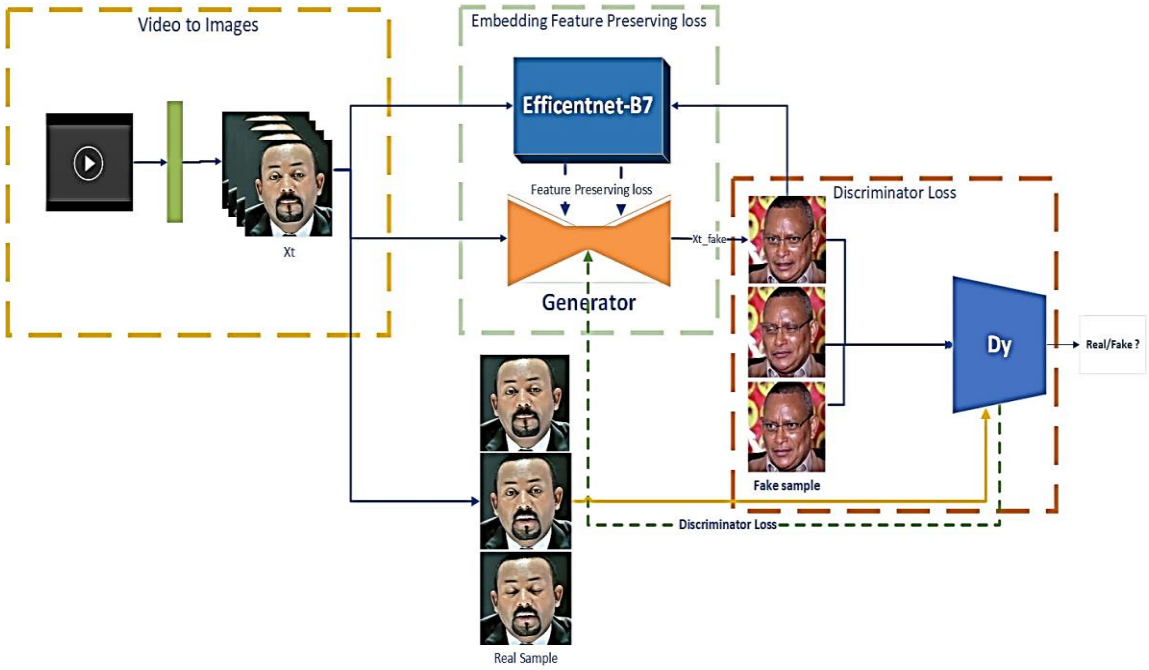


Figure 4-1 Model Architecture.

As shown in Figure 4-1, adjustments have been made to the discriminator network, and additional losses have been applied to the generative network; Section 5.4 addressed the depth implementation detail of the model architecture.

4.3. Model Learning Functions

The key objective of this thesis work is to optimize the use of space-time knowledge. In order to address the research query, this work adds loss functions, and change the discriminator network so that it can address temporal coherency to the Cycle-GAN and ReCycle-GAN: Since model architecture is based on Cycle-GAN and Recycle-GAN.

As discussed on the problem statement, the network seeks to transform a series in time domain images from the source domain, $X = \{x_1, x_2, x_3 \dots, x_n\}$, to a sequence of domain changed images, $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \tilde{x}_3 \dots, \tilde{x}_n\}$, with the exclusion of problems listed in Section 2.5. The function is then to acquire the mapping of $G_{AB} : \mathrm{X} \rightarrow \mathrm{Y}$. Note that model architecture uses sequential unpaired image data as input during training.

Because this work follows the GAN architecture, the vanilla adversarial loss is also used in this thesis work, called ℓ_{GAN}. And the cycle consistency loss ℓ_{cycle} and $\ell_{identity}$ in Cycle-GAN [2] is adopted. Besides, the recurrent loss $\ell_{recurrent}$ and the recycle loss $\ell_{recycle}$ in Recycle-GAN [1] are also leveraged. Meanwhile, this work introduces constrain $\boldsymbol{\ell_{featurePreserving}}$ to impel the model and improve the whole translation. Furthermore, the GAN loss changed accordingly to the temporal discriminator used. The full loss function of this model is as follows:

$$\min_{G,P} \max_{D} \boldsymbol{\ell} all(G,P,D) = \ell_{GAN}(G_{AB}, \boldsymbol{D_x^{t:3}}) + \ell_{GAN}(G_y, \boldsymbol{D_y^{t:3}}) + \alpha\ell_{cycle}(G_{AB}, G_{BA}) + \alpha\ell_{cycle}(G_{BA}, G_{AB}) + \mu\ell_{identity}(G_{AB}, G_{BA}) + \mu\ell_{identity}(G_{BA}, G_{AB}) + \beta\ell_{recurrent}(P_x) + \beta\ell_{recurrent}(P_y) + \gamma\ell_{recycle}(G_{AB}, G_{BA}, P_x) + \gamma\ell_{recycle}(G_{AB}, G_{BA}, P_y) + \theta\boldsymbol{\ell_{featurePreserving}}(G_{AB}) + \theta\boldsymbol{\ell_{featurePreserving}}(G_{BA}) \quad \textit{eq.(4.1)}$$

Where $\alpha, \mu, \beta, \gamma$ and θ are used hyperparameter of learning.

Indeed, the network needs more learning constraints, the aim of which is to demonstrate a significant consistency. Let's look in detail at all loss constraints. Keeping in mind that the translated video should preserve contain information but perhaps not the style**.** It should be close to the real image in another domain. The translator network should consider this constraint while learning in the training process.

Cycle Loss: Only unpaired samples are used independently in the respective videos during learning, without the need for paired input results. To fix this, the consistency of cycle continuity is necessary and leveraged by the process, which can be written as[2]:

$$\min_{G_{AB}, G_{BA}} L_{cyc}(G_{AB}, G_{BA}) = \sum_{t} \left[\left[\left\| G_{BA}\left(G_{AB}(x)\right) - x \right\|_1 \right] + \left[\left\| G_{AB}\left(G_{BA}(y)\right) - y \right\|_1 \right] \right] \quad \textit{eq.(4.2)}$$

where: G_{AB} and G_{BA} are generators, x and y are samples frames from both domain datasets.

Cycle consistency is a loss function asks a question to answer "the original image and the twice-translated (reconstructed image.) image are the same"? If this fails, we may not have a coherent mapping A-B-A. Meaning the original image A and the retranslated image A2B2A mean square distance should be minimum.

Identity Loss: Perhaps the most straightforward loss, identity loss ensures that the network retains the overall color structure of the image. So, adding a regularization concept lets us keep the tint of the photo in line with the original shot. Imagine that as a way to guarantee that the network can still recreate the original image even after adding several filters [2].

Identity loss is introduced to diminish translation of the images already in domain A to the Generator from G_{BA}, because the Cycle-GAN should understand that they are already in the correct domain. Meaning translating Amharic text to Amharic using English to Amharic translator; since the input is Amharic, the network should make no change.

$$\min_{G_{AB}, G_{BA}} L_{identity}(G_{AB}, G_{BA}) = \sum_{t} \left[\|G_{AB}(x) - x\|_1 \right] + \left[\|G_{BA}(y) - y\|_1 \right] \quad \textit{eq.(4.3)}$$

where: G_{AB} and G_{BA} are generators, x and y are samples frames from both domain datasets.

So, the full loss would be CycleGAN $= GAN\ loss + cycle\ \ loss + Identity\ Loss$

$$L\left(G_{AB}, G_{BA}, D_x, D_y\right)$$

$$= l_{GAN}\left(G_{AB}, D_y, X, Y\right) + l_{GAN}\left(G_{BA}, Dx, Y, X\right) + \alpha l_{cyc}\left(G_{AB}, G_{BA}\right) + \beta l_{identity}\left(G_{AB}, G_{BA}\right)$$

where: G_{AB} and G_{BA} are generators, D_x,and $D_y$are discriminators, respectively both domain X and Y are samples from both domain datasets.

The cycle-loss and identity-loss were extended to various temporal domains. However, these works consider only the spatial information in 2D images and completely disregard the temporal information for modeling, which is also extended by video translation.

Feature Preserving Loss: Indeed, classic cycle-consistency does not essentially assurance the transformation to be semantically consistent. This is, as a result, it does not consider any semantic correspondence during the translation, and thus the system can accomplish textbook cycle-consistency (i.e., L_{cyc}= 0) only if the inverse mapping recovers the original contents, regardless of how incorrect the forward mapping was. This assumption causes object disappearance problem hence, the network doesn't consider content translation to another domain.

$$\min_{G_{AB}, G_{BA}} L_{Fpreserving}(G_{AB}, G_{BA}) = \sum_{t} \Big[\Big[\| mNET(G_{AB}(x)) - mNET(x) \|_1 \Big] + \Big[\| mNET(G_{BA}(y)) - mNET(y) \|_1 \Big] \Big] \qquad eq.(4.4)$$

where: G_{AB} and G_{BA} are generators, x and y are samples frames from both domain datasets $mNET$ stands for a pre-trained EfficientNet-B7 feature extractor.

By adding the above loss, the network is inspired to minimize the Object Disappearance problem list in section 2.5 to have consistent semantics earlier and afterward the translation. This thesis work uses EfficientNet-B7 [53] as a feature extractor that enforces the content information that appears in the original image also should appear on translate. For example, if a person and a dog appear in image A so does in translated image A2B, albeit the style modification. (i.e. EfficientNet-B7 current state of classification algorithm tested on Image-net Dataset)

Recurrent Loss: To handle video data, the temporal ordering of the sequential frames must be taken advantage of. In Recycle-GAN [1], we adopt a recurrent temporal P_x predictor to predict frames in the future based on the past frame details. The repeated deficit is as follows:

$$\min_{P_x} l_{recurrent}(Px) = \sum_t \|x_{t+1} - Px(x_{t-1}^t)\|_1 \qquad \textit{eq. (4.5)}$$

where: $Px(x_{t-1}^t)$ is a prediction of Px given x_{t-1} and x_t as concatenated input.

***Recycle Loss*:** Merging image generator [2] and temporal prediction network. The recycle loss[1] across domains and time can be described as:

$$\min_{P_x G_y P_y,} l_{recycle}(G_{AB}, G_{BA}, P_x) = \sum_t \left\|x_{t+1} - G_{AB}(P_y(G_{BA}(x_{t-1}^t)))\right\|_1 \qquad \textit{eq. (4.6)}$$

where: G_{AB} and G_{BA} are generators, x_{t+1} and y_{t+1} are samples frames from both domain datasets, $Px(x_{t-1}^t)$ is a prediction of Px given x_{t-1} and x_t as concatenated input.

4.4. Temporal Discriminator

To improve visual quality further, a discriminator takes three consecutive generated images to decide whether it is real or fake. The discriminator architecture and the output stay the same with Patch GAN [3]; instead, the differences are just the input and the number of channels.

$$\mathrm{L_{adv}} = \min_G \max_D E_{x\epsilon X}(\log(D(x_t, x_{t-1}, x_{t-2})) + E_{x\epsilon Z}(\log(1 - D(G(y_t), G(y_{t-1}), G(y_{t-2})))) \qquad \textit{eq.(4.7)}$$

where:D discriminator and G is the generator, x_{t-2}, x_{t-1}, x_t and y_{t-2}, y_{t-1}, y_t are samples frames from both domain datasets.

Rather than differentiating between single frames, the discriminator network is designed in a way to observes three constitutive of synthesized frames and three constitutive of the real frames. This approach makes the discriminator network optimal because it takes into account the temporal nature of the video generation issue, such as Object Dislocation.

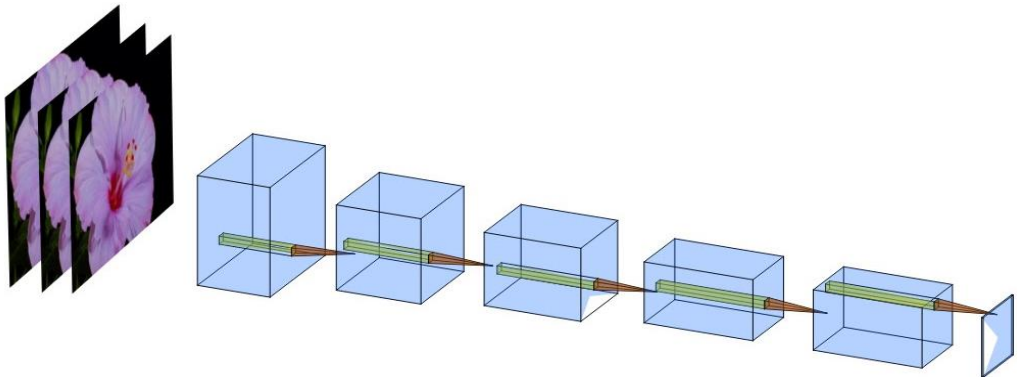


Figure 4-2 Temporal Discriminator Network

4.5. Training Pseudocode

Training algorithms for this thesis work have been introduced in this section as this study compares earlier research, Cycle-GAN, and ReCycle-GAN. Their training algorithms have also present in Appendix B.

Algorithm 1: Cycle GAN with feature preserving loss and temporal discriminator

Input: $\boldsymbol{x, y}$

Output: Generated frame $\boldsymbol{\tilde{x}, \tilde{y}}$

1 $Take\ a\ sample\ mini-batch: x, x_{t-1}, x_{t-2}, y, y_{t-1}, y_{t-2}$

2 Train D:

3 $Translate\ A, B: \tilde{x} = G_{AB}(x),\ \tilde{y} = G_{BA}(y)$

4 $\boldsymbol{D_A(x, x_{t-1}, x_{t-2}), D_B(y, y_{t-1}, y_{t-2}), D_A(y, \widetilde{y_{t-1}, y_{t-2}}), D_B(\ x, \widetilde{x_{t-1}, x_{t-2}})}$

5 $dloss = \frac{1}{4} * \sum((D_A(\overset{t:3}{x}), D_B(\overset{t:3}{y}))\ , (D_A(\overset{t:3}{\tilde{x}}), D_B(\overset{t:3}{\tilde{y}}))))$

6 $update\ \Theta^{(dloss)}\ to\ minimize\ classification\ loss.$

7 Train G:

8 $\tilde{\tilde{x}} = G_{AB}(\tilde{x}),\ \tilde{\tilde{y}} = G_{AB}(\tilde{x}), xI = G_{BA}(x), yI = G_{AB}(y),$
$\widetilde{\widetilde{x\_pred}} = G_{BA}(Py(G_{AB}(x_{t-1}), G_{AB}(x_{t-2}))),$
$\widetilde{\widetilde{y\_pred}} = G_{AB}(Px(G_{BA}(y_{t-1}), G_{BA}(y_{t-2})))$

9 $cycle\_loss = \frac{1}{2}(mae((x - \tilde{\tilde{x}}), (y - \tilde{\tilde{y}})))$

10 $identity\_loss = \frac{1}{2}(mae((x - xI), (y - yI)))$

11 $\boldsymbol{feature\_preserving\_loss = \frac{1}{2}(mae((mNet(x) - mNet(\tilde{x})), (mNet(y) - mNet(\tilde{y}))))}$

12 $\boldsymbol{D_A(\tilde{y}), D_B(\tilde{x}): d\_loss = \frac{1}{2}\sum(DB(\ x, \widetilde{x_{t-1}, x_{t-2}}), DA(y, \widetilde{y_{t-1}, y_{t-2}}))}$

13 $update\ \Theta^{(\text{d\_loss,cycle\_loss,identity\_loss,}feature\_preserving\_loss)}\ min\ loss.$

The bold indicates the added constraints.

Algorithm 2: ReCycle GAN with temporal discriminator

Input: $x, x_{t-1}, x_{t-2}, y, y_{t-1}, y_{t-2}$

Output: Generated frame $\tilde{x}, \tilde{y}$

1 $Take\ a\ sample\ mini-batch: x, x_{t-1}, x_{t-2}, y, y_{t-1}, y_{t-2}$

2 Train P:

3 $A, B:\ x\_pred = P_x([x_{t-1}, x_{t-2}])$

4 $A, B:\ y\_pred = P_y([y_{t-1}, y_{t-2}])$

$Compute: p_loss = \frac{1}{2} mea\left((x, x_pred), (y, y_pred)\right)$

5 $update\ \Theta^{(p_loss)}\ to\ minimize\ prediction\ loss$

6 Train D:

7 $Translate\ A, B:\ \tilde{x} = G_{AB}(x),\ \tilde{y} = G_{BA}(y)$

8 $\boldsymbol{D_A(x, x_{t-1}, x_{t-2}), D_B(y, y_{t-1}, y_{t-2}), D_A(y, \widetilde{y_{t-1}, y_{t-2}}), D_B(x, \widetilde{x_{t-1}, x_{t-2}}),}$

9 $dloss = \frac{1}{4} * \sum((D_A(x^{t:3}), D_B(y^{t:3})), (D_A(\tilde{x}^{t:3}), D_B(\tilde{y}^{t:3})))$

10 $update\ \Theta^{(dloss)}\ to\ minimize\ classification\ loss.$

11 Train G:

12 $\tilde{\tilde{x}} = G_{AB}(\tilde{x}),\ \tilde{\tilde{y}} = G_{AB}(\tilde{x}), xI = G_{BA}(x), yI = G_{AB}(y),$

13 $\widetilde{\widetilde{x\_pred}} = G_{BA}(Py([G_{AB}(x_{t-1}), G_{AB}(x_{t-2}))),$
$\widetilde{\widetilde{y\_pred}} = G_{AB}(Px([G_{BA}(y_{t-1}), G_{BA}(y_{t-2}))$

14

15 $cycle\_loss = \frac{1}{2}(mae((x - \tilde{\tilde{x}}), (y - \tilde{\tilde{y}})))$

16 $Recycle\_loss = \frac{1}{2}(mae((x - \widetilde{\widetilde{x\_pred}}), (y - \widetilde{\widetilde{y\_pred}})))$

17 $identity\_loss = \frac{1}{2}(mae((x - xI), (y - yI)))$

18 $\boldsymbol{D_A(\tilde{y}), D_B(\tilde{x}): d\_loss = \frac{1}{2}\sum(D_B(x, \widetilde{x_{t-1}, x_{t-2}}),\ D_A(y, \widetilde{y_{t-1}, y_{t-2}}))}$

19 $update\ \Theta^{(\text{d\_loss,cycle\_loss,identity\_loss,}Recycle\_loss,feature\_preserving\_loss)}\ min\ loss.$

The bold indicates the added constraints.

As discussed in the preceding section, the previous approach does not consider content translation, which leads to object dislocation and object disappearance problem. However, this work emphasizes minimizing the content difference between fake and real images, as shown in training pseudocode Table 4-1 line-11. On the other hand, the proposed method also modifies the patch-GAN discriminator to make it a temporal aware network, line-4, and line-12 in Table 4-1 and line-8 and line-18 in Table 4-2, which enforces the network care about the relation among consecutive three frames.

CHAPTER FIVE

5. IMPLEMENTATION OF THE PROPOSED SOLUTION

5.1. Chapter Overview

In this chapter, the implementation of the proposed solution is described. The working environment, cycle-GAN implementation, and experimental class conducted are discussed.

5.2. Working Environment.

This section explains the hardware stack that has been used to implement GAN experiments in addition to describing the hardware stack.

- Laptop: The Laptop computer is used for developing a network architecture.
  - Operating system: Windows 10
  - Processor: Intel ® Core™ i7-2300QM CPU @ 2.00GHz
  - Graphics: Intel ® Graphics 3000
  - Primary Memory (RAM): 8.00 GB
  - System Type: 64-bit Operating System, x64-based Processor
- Desktop: The desktop computer is used for developing a video for video translation.
  - Operating system: windows 10
  - Processor: Intel ® Core™ i5-4580 CPU @ 3.29GHz x 4
  - Graphics: Intel ® HD Graphics 4600
  - GPU: GeForce RTX 2070 Super 6 GB RAM
  - Primary Memory (RAM): 8.00 GB
  - System Type: 64-bit Operating System, x64-based Processor

Visual studio code and Jupyter notebook are used as a development IDE, with python interpreter 3.6 on a laptop computer. For implementing the proposed domain transfer problem OpenCV 3.7. Furthermore, TensorFlow-GPU 2.2 was used. In the next section list of experiments class conducted for evaluating the proposed hypothesis are discussed.

5.3. Environmental Setup

In this thesis work, different software and IDEs have been used.

Anaconda: in an application used to install the up-to-date version of python with its different module and IDEs, for implementing the proposed solution an anaconda application version 1.9.7 with 64-bit support used.

Jupyter Notebook: is the most popular and handy IDE among AI and deep learning researchers to work with python. This thesis work uses Jupiter note-book 6.0.0.

5.4. Cycle-GAN Implementation

A Cycle-GAN is made up overall of two GAN architectures: a generator and a discriminator. The generator architecture contains two separate models, Generator AB, and Generator BA. In the same manner, the discriminator architecture contains an additional two architectural models, Discriminator A, and Discriminator B. Table 5-1 shows convolutional layers of Cycle-GAN architecture.

The generator network is an encoder-decoder category network. It takes an image as an input with the shape (256,256,3), and outputs generated image with the same shape. Based on base work Cycle GAN two generator networks are defined. The generators had consisted of 15 layers. Four convolution layers followed by nine residual blocks and two deconvolutional layers - deconvolution means transposed 2-D convolution. The LeakyReLU activation was on all layers except the last layers output layer in the same manner Instance normalization was used in every layer besides the last one.

```
G_A2B = module.ResnetGenerator(input_shape=(256,256, 3))
G_B2A = module.ResnetGenerator(input_shape=(256,256, 3))
```

The discriminator network is equivalent to the discriminator's architecture in a Patch GAN network[3]. Basically, it takes an image of the shape of (256, 256, 3) and predicts whether the image is real or fake. This network composed of 5 convolutional layers denotes a 4 × 4 filter convolution-instance normalization with LeakyReLU layer and stride 2. After the last layer, apply a convolution to produce a 1-dimensional output. The slope of leaky in leakyReLU was 0.2.

```
D_A = module.ConvDiscriminator(input_shape=(256,256, 3))
D_B = module.ConvDiscriminator(input_shape=(256,256, 3))
```

Table 5-1 Cycle GAN architecture convolutional layers

Layer	***Generators***
1	Convolutional-(Filters-32, Kernel size-7, Stride-1), Instance normalization, LeakyReLU
2	Convolutional-(Filters-64, Kernel size-3, Stride-2), Instance normalization, LeakyReLU
3	Convolutional-(Filters-128, Kernel size-3, Stride-2), Instance normalization, LeakyReLU
4-12	Residual block-(Filters-128, Kernel size-3, Stride-1), Instance normalization, LeakyReLU
13	Convolutional-(Filters-64, Kernel size-3, Stride-0.5), Fractionally strided, Instance normalization, LeakyReLU
14	Convolutional-(Filters-32, Kernel size-3, Stride-0.5), Fractionally strided, Instance normalization, LeakyReLU
15	Convolutional-(Filters-3, Kernel size-7, Stride-1), Instance normalization, Tanh
Layer	***Discriminators***
1	Convolutional-(Filters-64, Kernel size-4, Stride-2),LeakyReLU with slope 0.2
2	Convolutional-(Filters-128, Kernel size-4, Stride-2), Instance normalization, LeakyReLU with slope 0.2
3	Convolutional-(Filters-256, Kernel size-4, Stride-2), Instance normalization, LeakyReLU with slope 0.2
4	Convolutional-(Filters-512, Kernel size-4, Stride-2), Instance normalization, LeakyReLU with slope 0.2

The generator's objective is to diminish the adversarial loss function against an adversary discriminator, which constantly tries to maximize it. Similar to other network types of GAN is no different. The learning function has to be explicitly defined in order for the network to learn to translate the image.

```
self.combined = tf.keras.Model(inputs=[img_A, img_B],
     outputs=[valid_A, valid_B,reconstr_A,
reconstr_B,img_A_id, img_B_id,img_A_id, img_B_id])

#define loss function
d_loss_fn, g_loss_fn =
gan.get_adversarial_losses_fn(adversarial_loss_mode)
cycle_loss_fn = tf.losses.MeanAbsoluteError()
identity_loss_fn = tf.losses.MeanAbsoluteError()
G_loss = (A2B_g_loss + B2A_g_loss) + (A2B2A_cycle_loss +
B2A2B_cycle_loss) * cycle_loss_weight + (A2A_id_loss +
B2B_id_loss) * identity_loss_weight
```

Almost all the configurations were taken from the Cycle-GAN paper and the implementations of its authors on GitHub.

5.5. Temporal Predictor Network Implementation

This thesis work uses Recycle-GAN temporal predictor network Px and Py for video retargeting, which is identical to the Pix2Pix [3] generator network. However, the input layer has been modified to receive two successive previous images.

```
inputs1 = tf.keras.layers.Input(shape=[256,256,3])
inputs2 = tf.keras.layers.Input(shape=[256,256,3])
#(bs, 256, 256, channels*2)
inputs = tf.keras.layers.concatenate([inputs1, inputs2])
```

The temporal predictor network predicts the next frame based on two previous frames taken as input. Like every neural network, the temporal predictor network is similar and has been explicitly defined in the network.

```
#input temporal predictor network
from temporal_predictor import Generator
Px = Generator(inputs)
Py = Generator(inputs)
```

```
P_optimizer = keras.optimizers.Adam(learning_rate = 2e-4,
beta_1 = 0.5)
#A_1, A_2, B_1 and B_2 are the previous two frames in Domain
A and Domain B
@tf.function
def train_P(A, A_1, A_2, B, B_1, B_2):
    with tf.GradientTape() as pt:
        A_p = Px([A_1, A_2],training = True)
        B_p = Py([B_1, B_2],training = True)
        xl1_loss = P_loss_fn(A, A_p)
        Px_loss = xl1_loss * LAMBDA
        yl1_loss = P_loss_fn(B, B_p)
        Py_loss = yl1_loss * LAMBDA
        P_loss = (Px_loss + Py_loss)* args.cycle_loss_weight
#update gradient weight
   P_grad = pt.gradient(P_loss, Px.trainable_variables
+Py.trainable_variables)
   P_optimizer.apply_gradients(zip(P_grad,
Px.trainable_variables + Py.trainable_variables))
    return A_p, B_p, {'Px_loss': Px_loss,  'Py_loss':
Py_loss}
```

As shown in the code snip, *train_p* function takes six argument variables. A, A_1, and A_2 are in domain A and the rest in domain B. Since Pix2Pix needs paired dataset A_1 and A_2 concatenated as an input, the network predicts A_p, which is predicted frame-based given inputs. The loss is the L1 distance between A_P and A, which is used to update the gradient weights.

5.6. Feature Preserving Loss Implementation

As discussed in the previous sections, feature preserving loss aims to minimize content information deference between real and the translated fake images. To do so Efficientnet-b7 pre-trained model is imported. Since the aim is to extract the feature map of the input image,

the last four layers are removed, as shown in code snip. Using a pre-trained EfficientNetB7 model, the new Tensorflow model has created.

```
import efficientnet.tfkeras as ef # pretrained EfficientNet
#remove the last four layers
base_model =
ef.EfficientNetB7(input_shape=(256,256,3),include_top=False)
x = base_model.layers[-4].output
mNet = tf.keras.Model(inputs = base_model.input, outputs=x)
```

Another function *get_content_feature* is define was then defined to measure the content of two pictures. in order to compute feature map new, which returns a feature map of the input images. Then compute feature preserving loss between the real image and fake image pair sets has been computed then the loss has been used to update network weight. Meaning the content loss would be the L1 distance between *M_B, M_B2A, and M_A, M_A2B*.

```
def get_content_features(a,b):
     return mNet(a), mNet(b)
M_A, M_A2B = get_content_features(A, A2B)
M_A_A2B = identity_loss_fn(M_A, M_A2B)
M_B, M_B2A = get_content_features(B, B2A)
M_B_B2A = identity_loss_fn(M_B, M_B2A)
```

5.7. Temporal Discriminator Network Implementation

This work also uses additional temporal aware discriminator network. As discussed in the presiding section, it takes three images to discriminate whether the images are real or fake.

```
def build_discriminator(n):
     inputA = tf.keras.Input(shape = (256,256,3))
     inputB = tf.keras.Input(shape = (256,256,3))
     inputC = tf.keras.Input(shape = (256,256,3))
     #concatinate inputs
     h = tf.keras.layers.Concatenate()([x, y, z])
```

```
    d1 = conv2d(h, 64, 4, 2)
    d2 = conv2d(d1, 128, 4, 2)
    d3 = conv2d(d2, 256, 4, 2)
    d4 = conv2d(d3, 512, 4, 2)
    d5 = conv2d(d4, 1, 4, 1)
    x = tf.keras.Model(img,d5,name=n)
    return x
```

The discriminator architecture and the output stay the same with patchGAN [3]; instead, the differences are just the concatenated input and the number of channels. This enforces the network to strictly focus on the relation among generated images and its relation with two previous images.

```
    A_d_logits = D_A((A,A_1,A_2), training=True)
    B2A_d_logits = D_A((B2A,B2A_1,B2A_2), training=True)
    B_d_logits = D_B((B,B_1,B_2), training=True)
    A2B_d_logits = D_B((A2B,A2B_1,A2B_2), training=True)
```

5.8. Experiment Class

To evaluate the essence of temporal information for video translation testing the initial hypothesis is mandatory. Five different classes of experiments are conducted, as shown below in Table 5-2 for each dataset group. The first three experiments focus on video translation on flower and viper datasets, while the rest two are basically for video retargeting on Obama-Trump and (አዲስ) Adiss datasets. The first class (CC) is all about vanilla Cycle-GAN image translation on a given sequence of images, considering the spatial domain only. The second is regarding consider using feature preserving loss. The third one includes temporal discriminator build up on the second experiment. The fourth experimental class uses vanilla ReCycle-GAN aiming video retargeting, and the last one merges ReCycle-GAN with temporal discriminator which become a total of ten experiments.

Table 5-2 Lists of experimental classes.

Notation	***Experiment***	***Training Epochs***	***Dataset***	***Model used***
CC	Baseline CycleGAN	200 epochs, 20 epochs	Flower dataset and Viper dataset	Cycle-GAN
CC+CP	CycleGAN baseline generator trained with additional feature preserving loss	200 epochs, 20 epochs	Flower dataset and Viper dataset	Cycle-GAN and EfficientNet-B7
CC+CP+TD	Cyclegan baseline generator trained with additional feature preserving loss and temporal discriminator network	200 epochs, 20 epochs	Flower dataset and Viper dataset	Cycle-GAN, flownet2, and Temporal aware discriminator.
RC	Baseline ReCycle-GAN	30 epochs	Obama trump dataset and Adiss Dataset	ReCycle-GAN
RC+TD	ReCycle-GAN with temporal discriminator	30 epochs	Obama trump dataset and Adiss Dataset	ReCycle-GAN & temporal Discriminator

CHAPTER SIX

6. RESULTS AND DISCUSSIONS

6.1. Chapter Overview

Previous chapters identified the methodologies that were selected to experimentally investigate the research propositions—this section reports on the outcomes of the experimental stage. The data collected and information are analyzed concerning the principal research goal posed in this thesis: How to preserve temporal consistency for a video-to-video translation? Moreover, this thesis work proposes a hypothesis that *"adding temporal consistency constrain would improve temporal consistency between successive frames."*.

6.2. Video-to-Video Translation

The video-to-video translation takes a video from the scene as an input to generate an equivalent video in other domains with the consideration of preserving temporal information. This work conducts different training experiments to explain the qualitative and quantitative outcomes of comparing the baselines on which the study is based tested on different datasets. This research work uses the inception score (IS) and a Human evaluation study to evaluate the experimental outcome. Using the training algorithms mentioned in the segment. 4.2.

The models compared in the evaluation are shown in Table 5-2. As discussed in the evaluation metrics, the Human evaluation study follows two protocols. The first question ask which video looks more real showing generated videos only, which are labeled P1 in the next section. The second question request which one looks more realistic and natural translation by showing real input video and the fake generated videos side by side and the result labeled P2. In fact, P1 and P2 answer to a different aspect of the result. Since P1 asks how good the generated video regardless of how the translation is good or not. On the other hand, P2 does not consider the quality of the generated video only, but also how realistic the translation is compared to the original input video. All training parameter was set to ten except identity loss which is five ($\alpha = \theta = \beta = \gamma = 10, \mu = 5$). The next section confers results and discussion on the experiment output found on each dataset.

6.2.1. Flower-to-Flower

Figure 6-3 demonstrates this method's synthesized frames on the flower dataset. The experiment takes around 2 days for training (200 epochs). The videos in this dataset show the blooming of different flowers, which is a relatively slow process, meaning the shifts between adjacent frames are relatively small. The proposed algorithm can generally preserve the consistency of a sense and content information based on the given input video. The translated flower in each target field retains a continuity for much of the time, with input flower at a different domain. Table 6-1 and Figure 6-1 shows the inception score and Human evaluation study of experimental runs of the network.

Table 6-1 IS score and Human evaluation study Result on flower Dataset

Methods	Flower			
	IS		Average Human evaluation study per domain[5]	
Real dataset	1.165±0.030	1.248±0.055		
	Domain A	Domain B	Domain A	Domain B
CC	1.022±0.002	1.102±0.031	25%	3.1%
CC + CP	1.023±0.009	1.122±0.184	9.4%	0%
CC + CP + TD	**1.138±0.041**	**1.162±0.025**	**65.6%**	**96.9%**

Bold values indicate the best results in the experiments.

Quantitative Experimental result demonstrates the proposed model score better quality and variation among the result it generates; this indicates the generated frames are quite unique. But CC and CC+CP model produces redundant frames comparatively; this is maybe because of the models lack of shape awareness.

Another experiment conducted Human evaluation score show, this thesis model is qualitatively better based on the human perceptual view in all P1, P2, and Average score per class. On the other hand, even though the Inception score of CC+CP outperforms vanilla Cycle-GAN, the Human evaluation study shows CC excel CC+CP. Indeed, The CC+CP network even amplifies the flickering effect.

[5]Human Evaluation User study for flower translation found at: https://forms.gle/dG3jo9iVskvXxLWLA

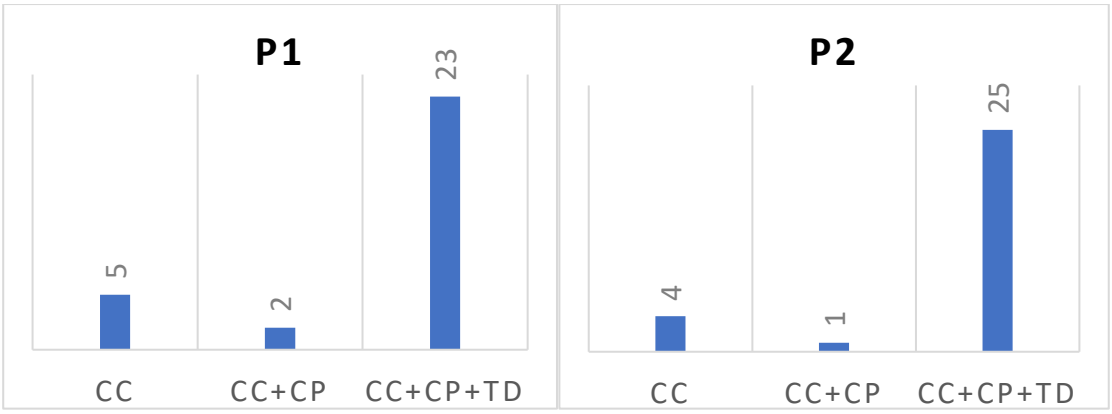


Figure 6-1 Human evaluation study on flower dataset

(Left) human evaluation study found after showing fake videos only to participants, (right) human evaluation study based on generated videos with the corresponding real video input. Higher values indicate the best results in the experiments.

Even if, temporal continuity can be preserved by model CC+CP+TD. Figure 6-3 shows that the result generated contains many artifacts because of the vanishing gradient problem. This indicates, training the model does not really have much weight changes after some epochs (meaning the gradient becomes very low-approximate to zero, multiplication of a small number further minimizes model weight) so any gradient update does change almost nothing in backpropagation. As a result, the discriminator network of CC+CP+TD becomes too complicated to be tricked by the generator network.

As seen in Figure 6-2 below, the discriminator loss becomes slightly similar to zero, and the generator loss will escalate to one. All the generator generated images are known as false, in other words, the discriminator network quickly bits the generator. Since ideal GAN training intentions for nash equilibrium to balance the generator and the discriminator network, the loss indicates quite the opposite. To mitigate the vanishing gradient and gradient explosion problem, gradient penalty had been applied to the discriminator network of CC+CP+TD; this improves the generated output video quality, furthermore, the loss of the network indicates gradient penalty had further stabilized the whole training process as shown in figure 6-4 below. (however, for a fair comparison gradient penalty result has not included in the Human evaluation study and Inception score report.). From the above observation, this work performs advantages over Cycle-GAN (CC) and Cycle-GAN with

Feature preserving loss (*CC+CP*), due to the improvements of video continuity and stability brought by the spatial-temporal constraint.

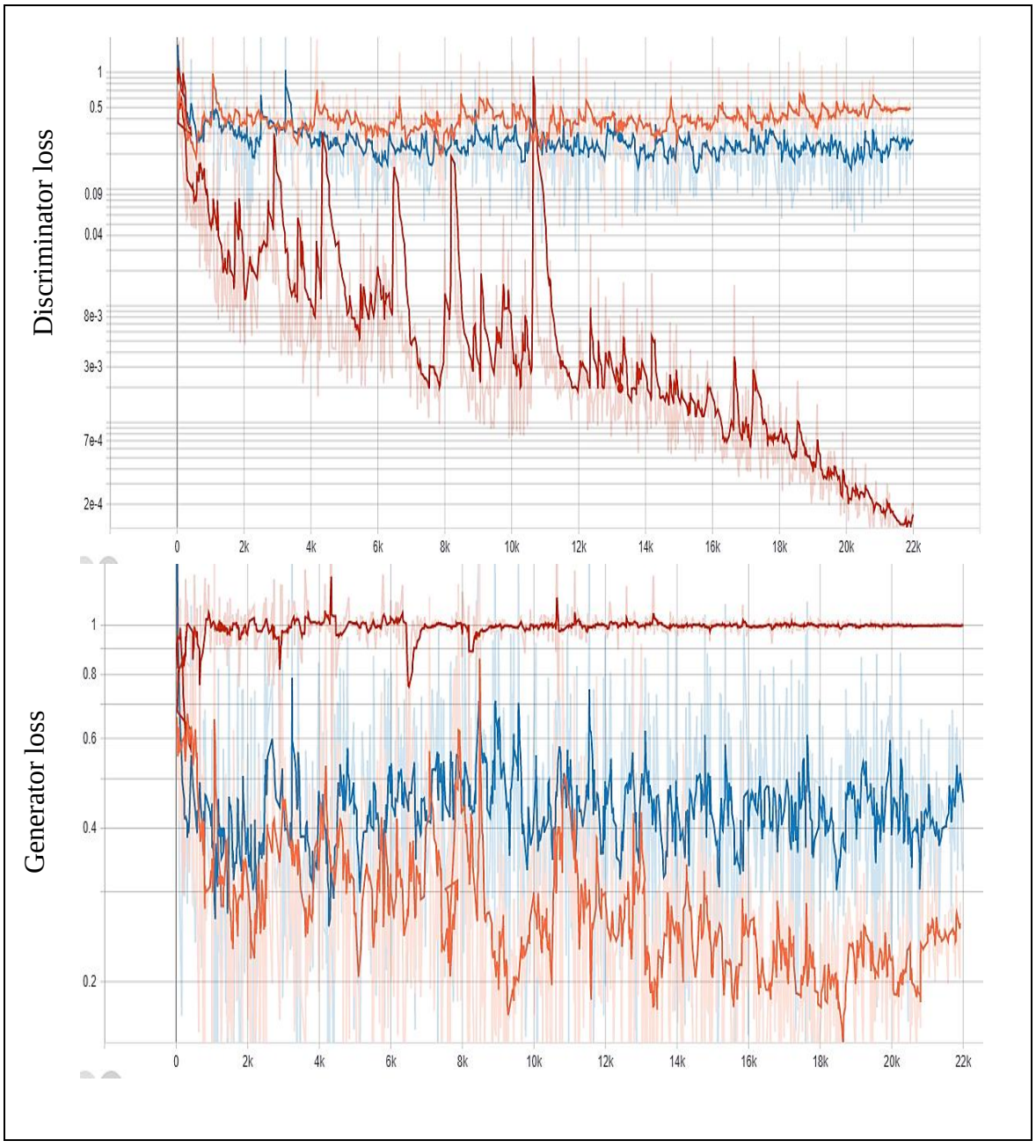


Figure 6-2 Training loss, vanishing problem on CC+CP+TD

(Top) Discriminator network, and (bottom) Generator network, (red): CC+CP+TD, (blue): CC+CP, (orange): CC. The CC+CP+TD discriminator loss quickly fails to zero results the generator loss to explode to one.

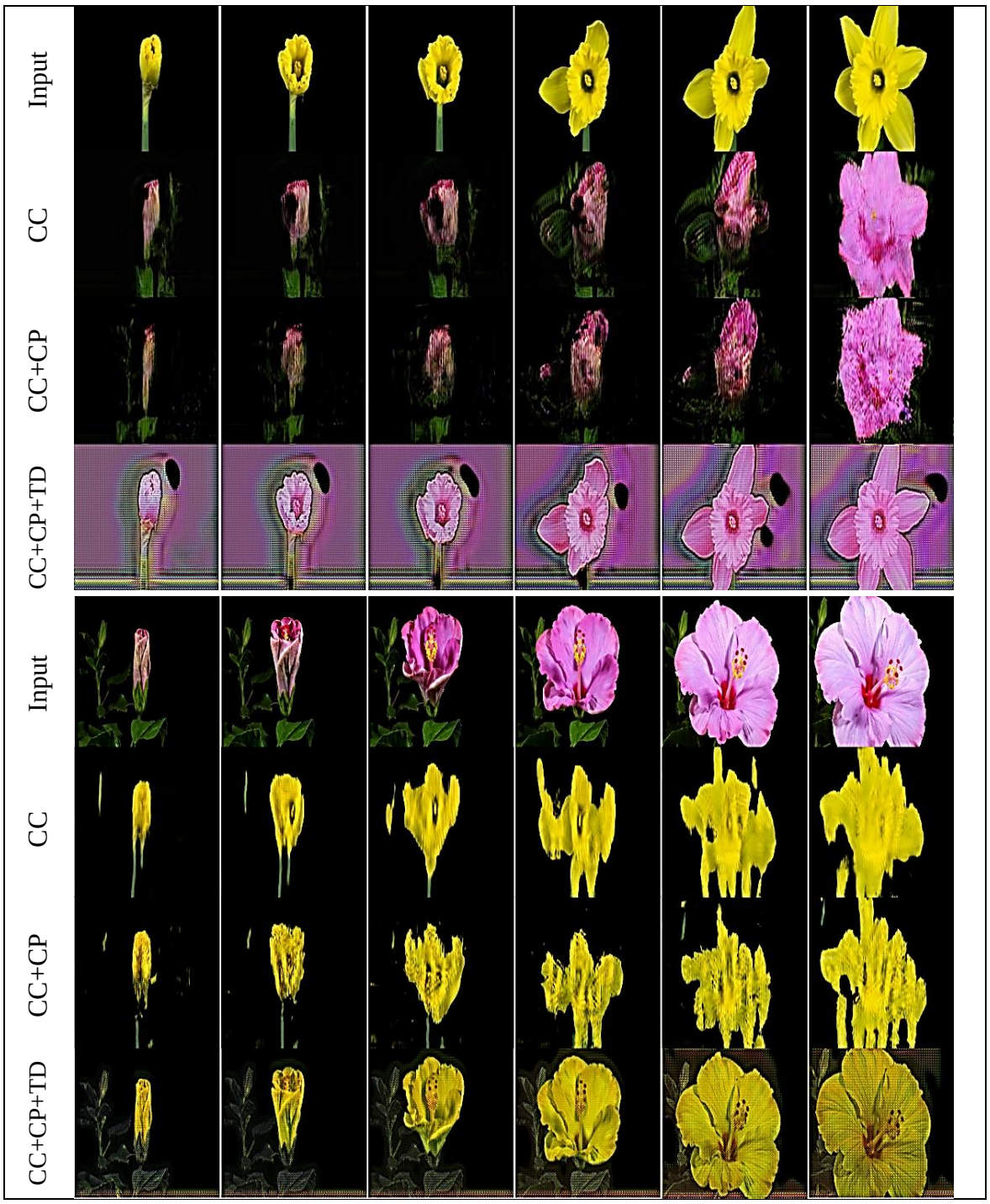


Figure 6-3 Flower to flower translation result

The real images labeled Input that the synthetic images are based on. The second-row is the result of Cycle-GAN, Third-row shows Cycle-GAN with Feature preserving loss, and the last includes temporal discriminator.

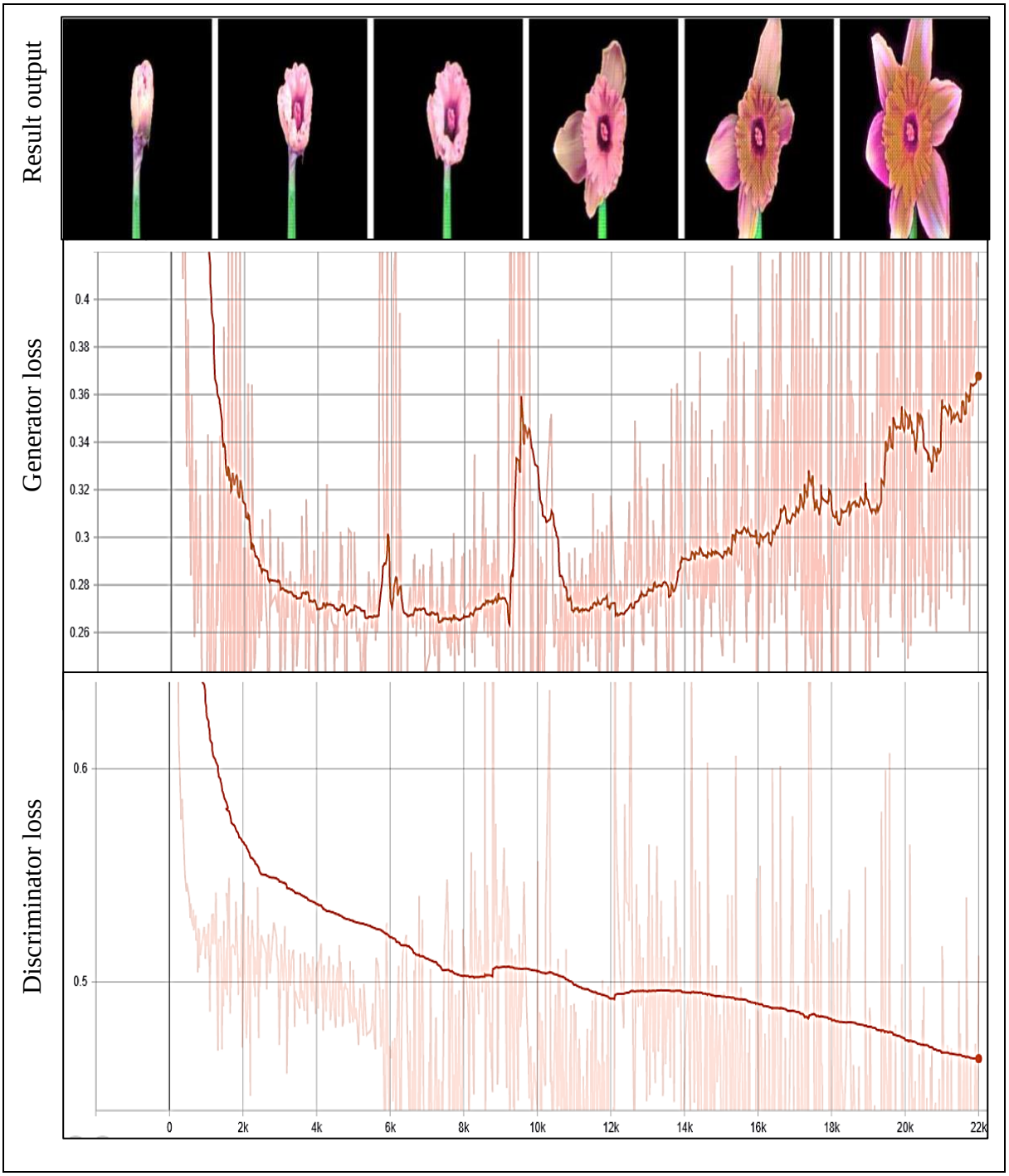


Figure 6-4 CC+CP+TD with gradient penalty

(Top) output sample (middle) generator loss, and (bottom) discriminator loss, the x-axis tells the number of instances and the y-axis indicates the loss.

6.2.1. Sunset-to-Day

Similar to flower-to-flower translation, this experiment uses the same training setup, except uses a subset of viper dataset target to translate day time video to sunset video and vice

versa. It takes around 1.16 sec per image and a total of 3.75 days, train for 20 epochs. The task of sunset-to-day is shown to explain the influence of our exploitation of the proposed solution of this thesis; therefore, more focus on the video quality improvements shown in Figure 6-5 and Figure 6-7.

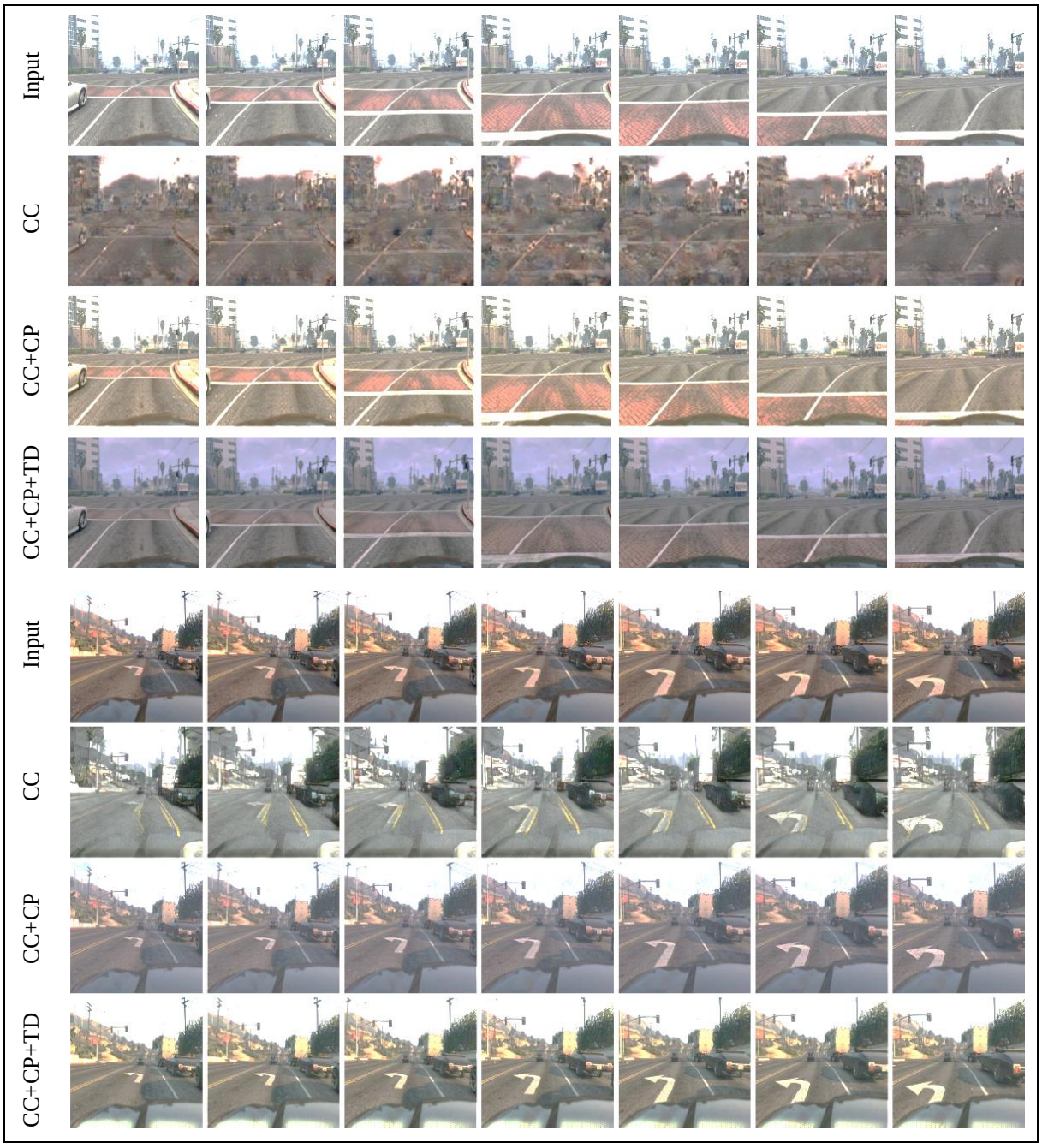


Figure 6-5 Sunset-to-day translation output result

The real images labeled Input that the synthetic images are based on. The second-row is the result of Cycle-GAN, Third-row shows Cycle-GAN with Feature preserving loss, and the last includes temporal discriminator.

Table 6-2 IS score and Human evaluation study Result on Viper Dataset

Methods	Day to Sunset			
	IS		Average Human evaluation study per domain [6]	
Real data	*3.56s±0.21*	*3.81±0.44*		
	Day	Sunset	Day	Sunset
CC	2.50±0.17	2.71±0.19	0%	5%
CC + CP	3.09±0.07	**3.64±0.26**	40%	40%
CC + TD	**3.23±0.13**	3.61±0.11	**60%**	**55%**

Bold values indicate the best results in the experiments.

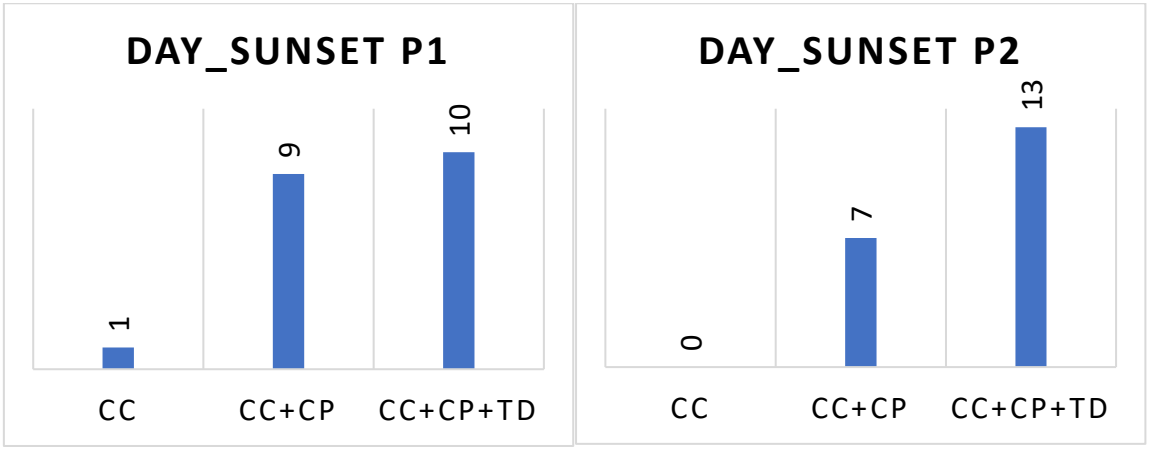


Figure 6-6 Human Evaluation Study on Viper dataset

(left) a human evaluation study found after showing fake videos only to participants, (right) a human evaluation study found after showing fake videos with the corresponding real video input. Higher values indicate the best results.

Eventually, this work positively improves visual quality, as confirmed in IS and Human evaluation study results (Table 6-2). This experiment may tell us a great deal about our method because the network can convert complex datasets successfully, while compared to the flower dataset. In the experiments CC+CP almost catches CC+CP+TD as shown in the above result report, CC+CP performs much better in the sunset-to-day datasets than the flower translation, as shown in Table 6-2 above. Even more, CC+CP performs as good as this thesis work as Figure 6-6 shows (P1 result show this work exceed CC+CP by only one

[6] Human Evaluation User study for Day to Sunset found at: https://forms.gle/xbkt9aFxA4YmNFnH6

vote). I suppose it is because the Efficientnet-B7 feature extracting network trains on ImageNet. I did not think there were substantial flower blooming (in fact, it only contains 1197 images of flowers, which is around 0.0084% of the entire dataset.) instances in training so the network might not be able to extract adequate features in flower video.

Figure 6-7 Comparison between Cycle-GAN with this thesis work on Sunset to Day

(Left) input images, (center) Cycle-GAN, (right) this thesis work, CC+CP+TD can preserve the detailed content and color information than CC.

For this reason, the network performed badly due to this cause. CC+CP+TD was impacted by the vanishing of the gradient problem in the tiny dataset as seen in the flower to flower translation, and maybe the viper dataset is big enough as the pixelated and the artifacts problem in the flower translation has diminished even better. The Human evaluation study scores tell that a majority of the participants prefer our synthesized videos than those comparative models 55% over Cycle-GAN and 17.5% on Cycle-GAN with feature

preserving loss. Figure 6-7 shows that CC does not retain detailed information, but this model generates a decent result positively toward content translation compared with CC. As shown in the above figure, black artifacts hover the building (red box), the car shape altered a bit (yellow box), and the color of the road line (green box) change as the result of Cycle-GAN. But the proposed model can preserve the color and shape of the input frame better than comparative work.

6.2.1. Obama-to-Trump

Unlike in the previous two video-to-video translation experiments which designed for translating spatial information, and content from input video to generated one, video retargeting in another way, aim to translate video motion from source to driving video.

In this experiment, Obama to trump translation using the Recycle-GAN and ReCycle-GAN with temporal discriminator has been evaluated. The result shows both approaches are capable of assessing the stylistic facial gestures of Donald Trump and Barack Obama[7] *(Please note that the photos are very minimal in their representation*). Nevertheless, mouth motion slightly Differ, as shown in the above figure 6-8. For example, in Trump to Obama translation, the RC+TD model fetches trump mouth movement more reasonably than the comparative model ReCycle GAN. Training takes around 4.12 days for 30 epochs.

The IS result shows this thesis work ReCycle-GAN with temporal discriminator (RC+TD) advantages over ReCycle-GAN(RC). (i.e. Content preserving network cannot be implemented since this work focuses on video Retargeting). In comparison with that of the ReCycle-GAN, our network increases the IS with a slight advantage as shown below in Table 6-3, and it also outperforms human evaluation study by 40% of base work.

RC model has been suffered from image flipping problem as shown below in Figure 6-10, which truly inherited from the convolutional neural network [55]. (But for fair comparison in Figure 6-8, RC output images have been aligning according to the input.). However,

[7] Please check the website for better comparison: https://sites.google.com/astu.edu.et/kirubelabebe/temporal-cycle-consistency-constraint-for-video-to-video-translation-result

RC+TD could maintain input video alignment which indicates the temporal discriminator network has better orientation awareness in its model weight.

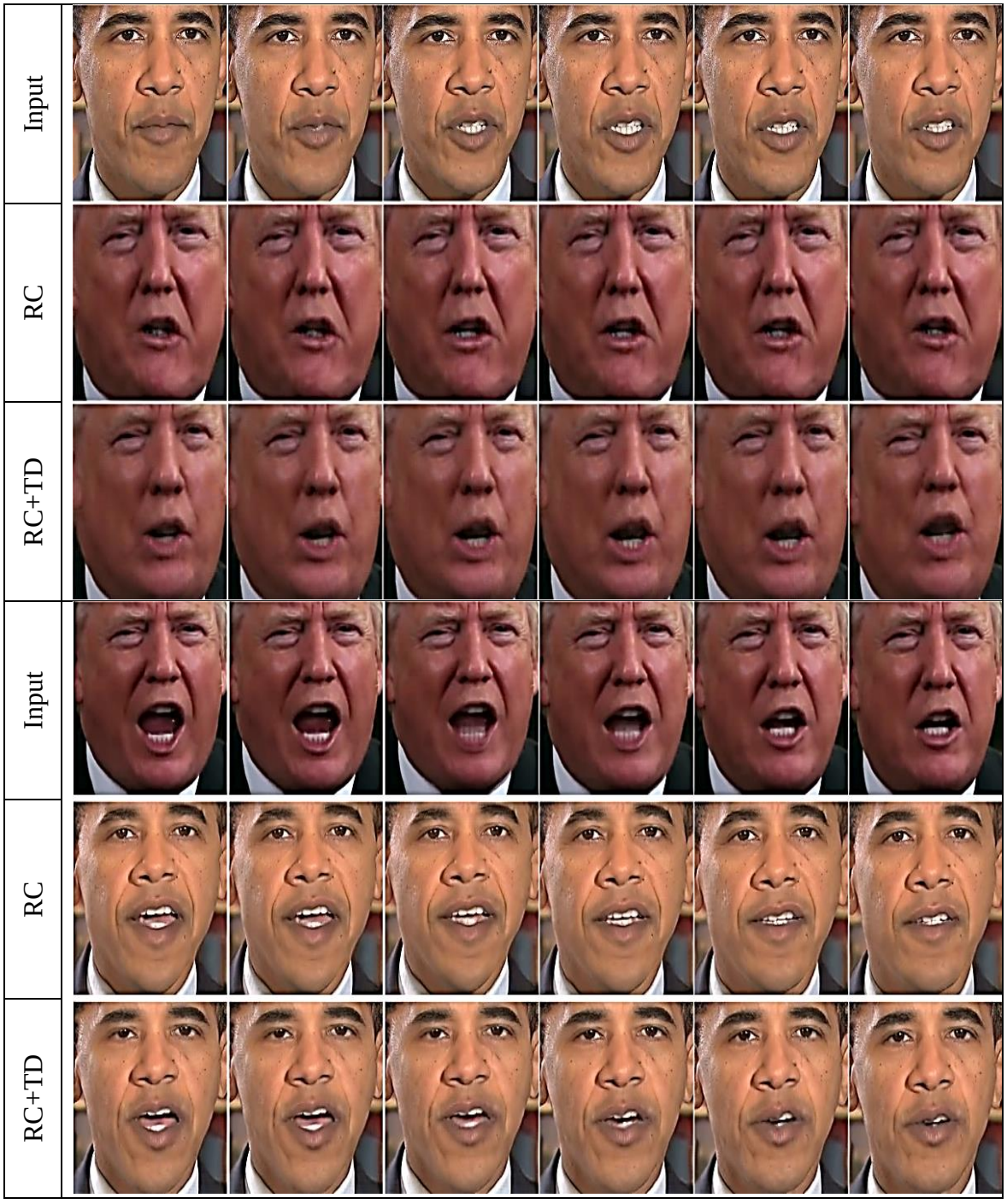


Figure 6-8 Obama to trump translation result

Row label as six sequential inputs are the inputs to the network, and the rests are the corresponding output of the network. The top three are Obama to Trump, and the bottom ones are the reverse translation.

Table 6-3 Obama to trump inception score and human evaluation study.

Methods	Obama to Trump			
	IS		Average human evaluation study per domain [8]	
Real data	1.283±0.102	1.069±0.274		
	Obama	Trump	Obama	Trump
RC	1.035±0.120	1.048±0.010	40%	20%
RC + TD	**1.041±0.013**	**1.068±0.011**	**60%**	**80%**

Bold values indicate the best results in the experiments.

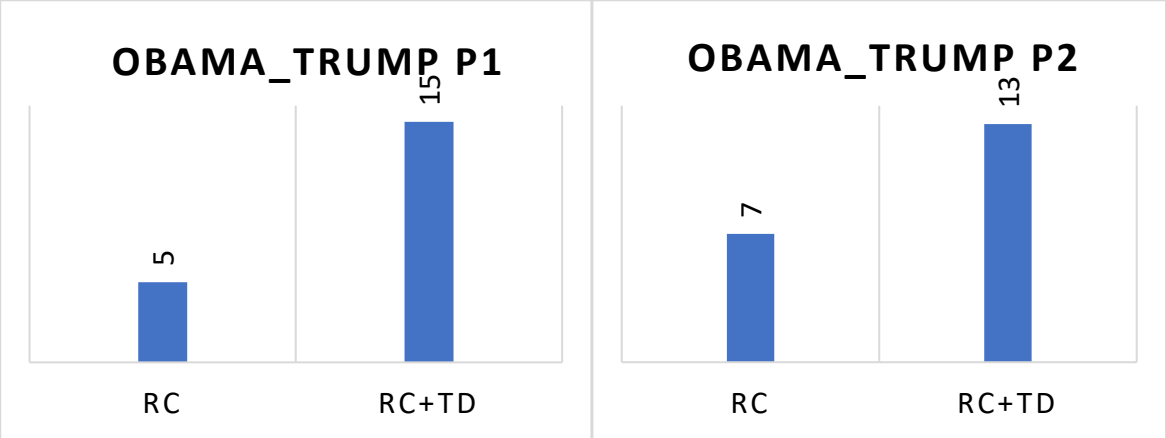


Figure 6-9 Human evaluation study on Obama-Trump dataset

(left) A human evaluation study found after showing fake videos only to participants, (right) A human evaluation study found after showing fake videos with the corresponding real video input. Higher values indicate the best results in the experiments

Figure 6-10 RC Trump Generated image sequences

Generated video sequences from the RC model result flipped Trump out.

[8] Human Evaluation Study for Trump to Obama found at: https://forms.gle/ydauVZbixebUJVYJA

On the other hand, the temporal discriminator network positively impacts video retargeting based on qualitative and quantitative evaluations discussed above; however, since Temporal Discriminator forces RC+TD model, the distance between successive frames become very small and similar, doing so limit motion change which sometimes makes the result too static and unreal. On the other hand, the temporal discriminator network positively impacts video retargeting based on qualitative and quantitative evaluations discussed above; however, since Temporal Discriminator forces RC+TD model, the distance between successive frames become very small and similar, doing so limit motion change which sometimes makes the result too static and unreal.

6.2.1. Abiy-to-Debretsion

This experiment extends face to face to implicate performance of the model on the local dataset Adiss (አዲስ) which relatively is a very complex datasets contain full face including hair, eyeglass, colorful background, and unaligned face direction. As shown in the generated output of RC and RC+TD, both models almost perfectly capture the head movement of the input video. But both models failed to mimic lip movement well.

Table 6-4 Abiy-to-Debretsion translation result

Methods	Abiy-to-Dibretsion			
	IS		Average human evaluation study per domain [9]	
Real data	*1.250±0.055*	*1.475±0.047*		
	Debretsion	Abiy	Debretsion	Abiy
RC	1.171±0.054	**1.314±0.033**	35%	35%
RC + TD	**1.218±0.030**	1.307±0.031	**65%**	**65%**

Bold values indicate the best results in the experiments

The quantitative result of RC+TD improves the inception score on Debretsion's fake video with a significant amount. on the other side, the thesis work lost by base work with a slight margin fake Abiy video. The human evaluation study claims, participants were confident about choosing a better one among the result found from RC and RC+TD since the fake

[9] Human Evaluation User study for Day to Sunset found at: https://forms.gle/u2qi7DwBnyJCeW2d9

Abiy video result was simply recognizable as fake. The complex nature of the Adiss(አዲስ) dataset in several aspects such as it does not consider face keypoint alignment and face size normalization impact network convergence. Additionally, from the result in figure 6-11 below its clear, the network basically emphases on translating head motion rather than lip movement on the input video.

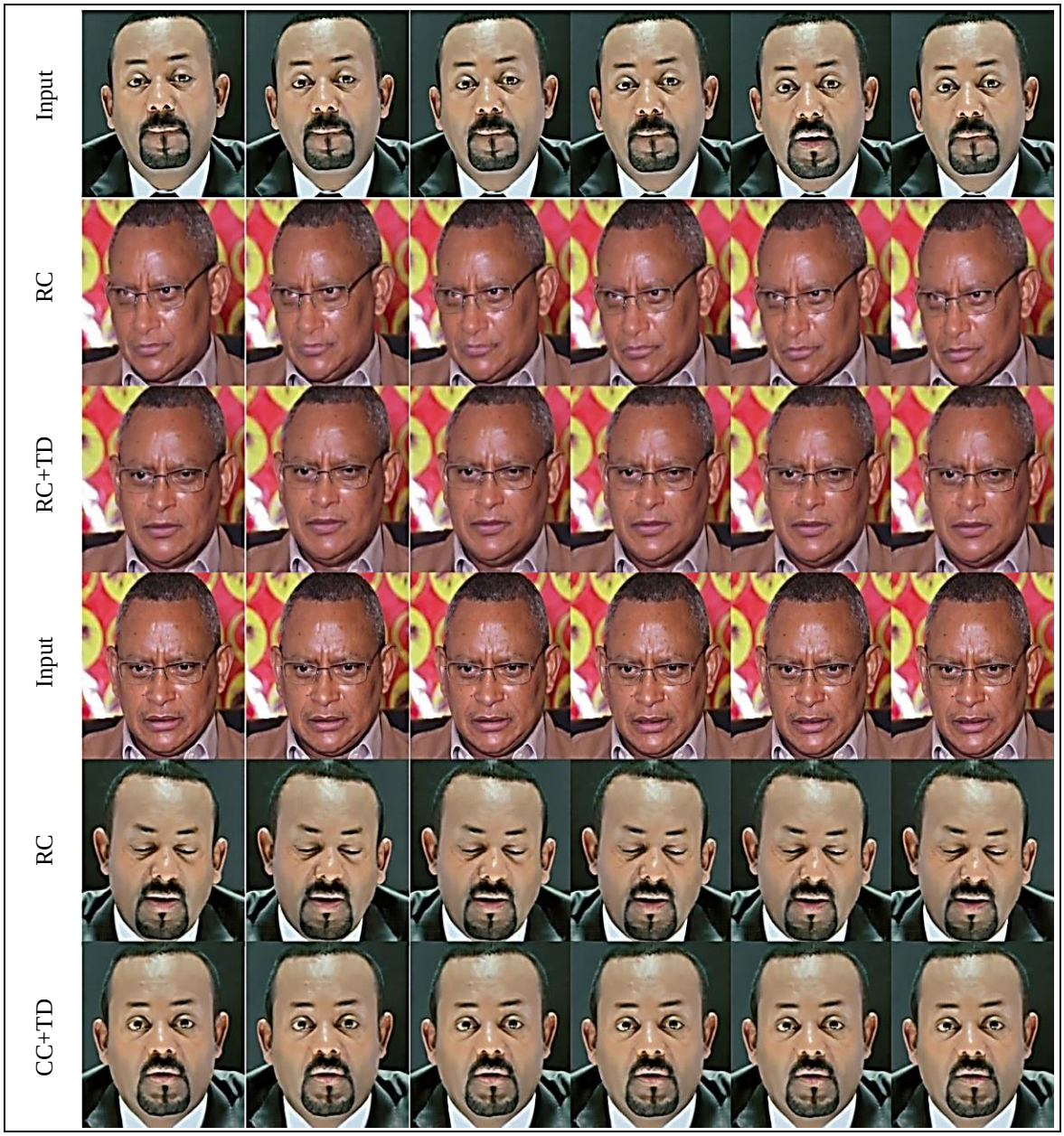


Figure 6-11 Abiy to Debretsion translation result.

Row label as six sequential inputs are the inputs to the network, and the rests are the corresponding output of the network. The top three are Abiy to Debretsion, and the bottom ones are the reverse translation.

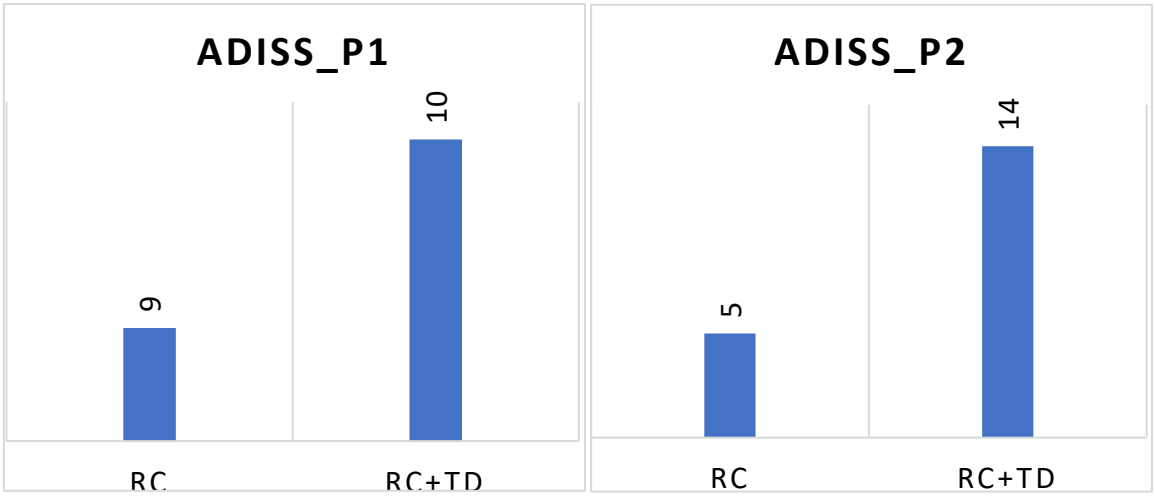


Figure 6-12 Human evaluation study on adiss dataset

(left) A human evaluation study found after showing fake videos only to participants, (right) A human evaluation study found after showing fake videos with the corresponding real video input. Higher values indicate the best results in the experiments

Even if both networks typically produce an unsatisfactory result the user study indicates this thesis model has a slight edge in both P1, P2 scores. Furthermore, the RC+TD Average Human evaluation study per domain surpasses ReCycle-GAN by 30%.

6.3. Results Discussion

6.3.1. Video-Translation Summary

Based on quantitative and qualitative results, on flower-to-flower translation and sunset-to-day, the simplest model CC, which only trained only on the cycle-loss, has the lowest score among the models, indicating that the (with respect) cycle GAN architecture is not complex enough for the video-to-video translation. Since CC only considers the spatial domain, the translation lacks knowledge of the temporal domain. The CC+CP model shows a non-significant improvement in the flower dataset but performs relatively well on the Viper dataset, which shows dependability on Efficientnet-B7. CC+CP+TD outperforms the baseline work Cycle-GAN by almost 61% and 49.95% on Cycle-GAN with feature preserving loss.

6.3.2. Video-Retargeting Summary

Inception Score on RC and RC+TD indicate optimistic progress of the model applied, which illustrate the robustness of this thesis work in order to learn better spatial-temporal information from the video and to produce better content consistency. In reality, both models are capable of learning the style of the input video and speaker action, but the outcomes are far from flawless. Based on the ReCycle GAN (RC) model generated result, it lacks shape and orientation knowledge as shown on the Obama-to-trump translation in which case the model flip trump face horizontally. Evidently, ReCycle-GAN with temporal discriminator output in the human evaluation analysis reveals an average 30 percent increase in the model performance.

CHAPTER SEVEN

7. CONCLUSION AND FUTURE WORK

7.1. Conclusion

Video-to-video translation is a natural extension of an image-to-image translation. Translating video points toward learning objects appearance in a scene and realistic motion movement between successive frames. A straightforward way to video-to-video translation carry out the image-to-image translation in each frame of input videos without considering those frames that have a relation between them. This approach is non-trivial since the underlying flickering problem effect is in the output video.

The purpose of this study was to improve temporal coherence for the video-to-video translation by adding constraints to the GAN network learning function trained on the unpaired dataset, which starts on the ReCycle-GAN claim. Among the investigation, the goal was to generate as visually realistic video as possible. To do so, this thesis adds Feature preserving loss, and Temporal aware discriminator to the baseline works. Indeed, these changes make the proposed model very aware of the perpetual spatial-temporal information changes in the video.

Different from early approaches, which focus only on the generated image, look real or fake based on Spatial information only. this approach enforces the discriminator network to emphasize not only on the spatial domain to judge real or fake but also check temporal coherency between the generated image and its preceding two frames. To decide the number of frames the discriminator should consider the ablation study had conducted and indicate three frame discriminators found better one among compared others.

Object disappearing appears to be another issue in recent works, so this thesis introduces a loss-preserving constraint to minimize the distance between the extracted Efficientnet-B7 features on the generated fake image and the original input. The implemented model Combines the above two losses to preserve temporal information. In fact, this work has been impacted by the Efficientnet-B7 model weight which makes it dependent, furthermore, it impacted by dataset type and size such as seen on the flower to flower translation. The model becomes unstable and collapses by vanishing gradient problems. Even more, it produces artifacts in the generated fake images. However, applying a gradient penalty to the

discriminator network improves the vanishing gradient problem, and a better result is generated.

Compared with baseline works Cycle-GAN [2], and ReCycle-GAN [1], qualitative and quantitative experimental findings indicate. The Cycle-GAN model, trained only on Cycle Loss, has the lowest evaluation score among this thesis work. Hence, it suggests that Cycle-GAN architecture is not complex enough for the video-to-video translation. Since it considers the spatial domain only, the translation lacks information on the temporal domain. Another version Cycle-GAN with content preserving loss also was very dependent on the feature extraction model used.

In the case of video retargeting experiments, the RC model can capture the style and content of a video as well as the proposed model. However, lousy alignment result has been seen in the generated trump video result as well in the fake Abiy video, in which the eye was closed for the entire period. RC+TD shows better shape aware retaining tendency from the experiments, which helps to better quality generated videos and applying temporal discriminator network retain better flow as shown ablation study.

Experiments show more significant variation among the result, both qualitatively and quantitatively. This thesis's achievement is that it excels in the human assessment analysis by 60 percent Cycle-GAN and 35 percent in the ReCycle GAN. This Research work concludes that adding temporal cycle consistency constraints to video-to-video translation does improve temporal coherency. It has been shown by the inception score and Human evaluation study experimental results. Further, these changes make the proposed model to learn better spatial-temporal information between consecutive video frames. so that, temporal coherency does improve. The temporal discriminator network also positively impacts the video-to-video translation. regarding the orientation and color preservation between translated corresponding original input.

7.2. Future Work

The thesis method does not come without limitations. As observed in the experiment, the model is strongly dependent on the EfficientNET-B7 outputs, and since the feature preserving loss is not designed to be consistent across frames. Generated output depends essentially on the feature extraction network's performance on a specific training dataset, as

discussed in 6.2.1. This naturally leads to inconsistency in the results produced. One approach to resolving this issue is a retune feature extraction network on the training dataset.

Furthermore, rather than a simple concatenation of output images in a manner to make temporal ware, the discriminator network could be modeled in a much efficient approach using the transformer network [56] so further research could be work to extend in a better approach. Although the number of participants in human evaluation is a very small range from 10 to 15 persons, further evaluation of human study would improve for better judgment.

Finally, this thesis's work paves a way to a different learning path: letting it learn using the discriminator network with considering a substantially long sequence of frames, focus on how to construct synthetic intermediate frames between successive frames, could increase the video's frame rate. HFR or (high frame rate) videos will increase the movement representation and consequently provide better pictures to increase the audience's accuracy. Perhaps it could need considerably more than that of two frames as used in this thesis work.

*** * ***

This research was funded by grant number **ASTU/SM-R/091/19** from Adama Science and Technology University.

Adama, Ethiopia.

REFERENCES

[1] A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh, “Recycle-GAN: Unsupervised Video Retargeting,” *CoRR*, vol. abs/1808.0, 2018.

[2] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,” *CoRR*, vol. abs/1703.1, 2017.

[3] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-Image Translation with Conditional Adversarial Networks,” *CoRR*, vol. abs/1611.0, 2016.

[4] D. Chen, J. Liao, L. Yuan, N. Yu, and G. Hua, “Coherent Online Video Style Transfer,” in *CoRR*, 2017, vol. 2017-Octob, pp. 1114–1123.

[5] “What Happens Now That An AI-Generated Painting Sold For $432,500?” [Online]. Available: https://www.forbes.com/sites/williamfalcon/2018/10/25/what-happens-now-that-an-ai-generated-painting-sold-for-432500/#f7702aca41ca. [Accessed: 12-Dec-2019].

[6] “Attempts on Real Time Style Transfer – mc.ai.” [Online]. Available: https://mc.ai/attempts-on-real-time-style-transfer/. [Accessed: 22-Sep-2020].

[7] L. Gatys, A. Ecker, and M. Bethge, “A Neural Algorithm of Artistic Style,” *J. Vis.*, vol. 16, no. 12, p. 326, Aug. 2016.

[8] Y. Chen, Y. Pan, T. Yao, X. Tian, and T. Mei, “Mocycle-GAN: Unpaired video-to-video translation,” *MM 2019 - Proc. 27th ACM Int. Conf. Multimed.*, pp. 647–655, Aug. 2019.

[9] S. Tulyakov, M.-Y. Y. Liu, X. Yang, and J. Kautz, “MoCoGAN: Decomposing Motion and Content for Video Generation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1526–1535.

[10] H. Liu, C. Li, D. Lei, and Q. Zhu, “Unsupervised video-to-video translation with preservation of frame modification tendency,” *Vis. Comput.*, 2020.

[11] I. Goodfellow *et al.*, “Generative Adversarial Nets (NIPS version),” *Adv. Neural Inf. Process. Syst. 27*, 2014.

[12] M. Mirza and S. Osindero, “Conditional Generative Adversarial Nets,” Nov. 2014.

[13] C. Cao, Q. Hou, and K. Zhou, “Displaced dynamic expression regression for real-time facial tracking and animation,” in *ACM Transactions on Graphics*, 2014, vol. 33, no. 4.

[14] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks,” *CoRR*, vol. abs/1812.0, 2018.

[15] T. C. Wang, M. Y. Liu, J. Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, “High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8798–8807.

[16] H. Huang *et al.*, “Real-time neural style transfer for videos,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-Janua, pp. 7044–7052.

[17] Jake VanderPlas, *Python Data Science Handbook*. O’Reilly Media, Inc.

[18] R. Rojas, “Neural Networks: A Systematic Introduction. ,” *Springer New York, NY, USA -Verlag New York, Inc.*, 1996.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[20] G. E. H. Alex Krizhevsky, Ilya Sutskever, “ImageNet Classification with Deep Convolutional Neural Networks,” *ILSVRC2012*, pp. 1–1432, 2007.

[21] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN,” Jan. 2017.

[22] J. L. and V. Bok, *GANs in Action: Deep learning with Generative Adversarial Networks*. Manning Publications Co., 2019.

[23] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with

deep convolutional generative adversarial networks," in *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 2016.

[24] "Image-to-Image Translation: Machine Learning Magic that Converts Winter Photos Into Summer - Abto Software, Lviv, Ukraine." [Online]. Available: https://www.abtosoftware.com/blog/image-to-image-translation. [Accessed: 03-Mar-2020].

[25] Y. Choi, M.-J. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation," *CoRR*, vol. abs/1711.0, 2017.

[26] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, "Temporal Cycle-Consistency Learning," Apr. 2019.

[27] D. Bashkirova, B. Usman, and K. Saenko, "Unsupervised Video-to-Video Translation," no. Nips, 2018.

[28] K. Vougioukas, S. Petridis, and M. Pantic, "Realistic Speech-Driven Facial Animation with GANs," *Int. J. Comput. Vis.*, 2019.

[29] A. R. Kosiorek, H. Kim, I. Posner, and Y. W. Teh, "Sequential attend, infer, repeat: Generative modelling of moving objects," *Adv. Neural Inf. Process. Syst.*, vol. 2018-Decem, no. NeurIPS, pp. 8606–8616, 2018.

[30] B. Kratzwald, Z. Huang, D. P. Paudel, A. Dinesh, and L. Van Gool, "Improving Video Generation for Multi-functional Applications," 2017.

[31] M. Chu, Y. Xie, J. Mayer, L. Leal-Taixé, and N. Thuerey, "Learning Temporal Coherence via Self-Supervision for GAN-based Video Generation," Nov. 2018.

[32] C. Militello, L. Rundo, and M. C. Gilardi, "Applications of imaging processing to MRgFUS treatment for fibroids: a review," *Transl. Cancer Res.*, vol. 3, no. 5, pp. 472–482, 2014.

[33] K. Park, S. Woo, D. Kim, D. Cho, and I. S. Kweon, "Preserving semantic and temporal consistency for unpaired video-to-video translation," *MM 2019 - Proc. 27th*

ACM Int. Conf. Multimed., pp. 1248–1257, Aug. 2019.

[34] X. Wei, S. Feng, J. Zhu, and H. Su, "Video-to-video translation with global temporal consistency," *MM 2018 - Proc. 2018 ACM Multimed. Conf.*, pp. 18–25, 2018.

[35] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-Janua, pp. 1647–1655.

[36] D. Sun, X. Yang, M. Y. Liu, and J. Kautz, "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8934–8943.

[37] A. Dosovitskiy *et al.*, "FlowNet: Learning optical flow with convolutional networks," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 2758–2766, 2015.

[38] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, vol. 7577 LNCS, no. PART 6, pp. 611–625.

[39] J. P. Bennett, "Everybody Dance Now!," *J. Phys. Educ. Recreat. Danc.*, vol. 77, no. 1, pp. 6–7, Jan. 2019.

[40] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, "GANimation: One-Shot Anatomically Consistent Facial Animation," *Int. J. Comput. Vis.*, no. January, 2019.

[41] S. Webber, M. Harrop, J. Downs, T. Cox, N. Wouters, and A. Vande Moere, "Everybody Dance Now: Tensions between participation and performance in interactive public installations," *OzCHI 2015 Being Hum. - Conf. Proc.*, pp. 284–288, 2015.

[42] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "Animating Arbitrary Objects via Deep Motion Transfer," 2018.

[43] W. S. Lai, J. Bin Huang, O. Wang, E. Shechtman, E. Yumer, and M. H. Yang, "Learning blind video temporal consistency," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11219 LNCS, pp. 179–195.

[44] "Edraw Max - Excellent Flowchart Software & Diagramming Tool." [Online]. Available: https://www.edrawsoft.com/edraw-max/. [Accessed: 02-Jun-2020].

[45] "TensorFlow." [Online]. Available: https://www.tensorflow.org/. [Accessed: 02-Jun-2020].

[46] J. Hale, "Deep Learning Framework Power Scores," 2019. [Online]. Available: https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a.

[47] "AI deep learning frameworks ranking 2018 | Statista." [Online]. Available: https://www.statista.com/statistics/943038/ai-deep-learning-frameworks-ranking/. [Accessed: 22-Sep-2020].

[48] "OpenCV." [Online]. Available: https://opencv.org/. [Accessed: 02-Jun-2020].

[49] "Design, visualize, and train deep learning networks - MATLAB." [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/deepnetworkdesigner-app.html. [Accessed: 05-Jun-2020].

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016.

[51] C. Szegedy *et al.*, "Going deeper with convolutions," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1–9, 2015.

[52] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-January, pp. 1800–1807.

[53] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 10691–10700, May 2019.

[54] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved Techniques for Training GANs,” *Adv. Neural Inf. Process. Syst.*, pp. 2234–2242, Jun. 2016.

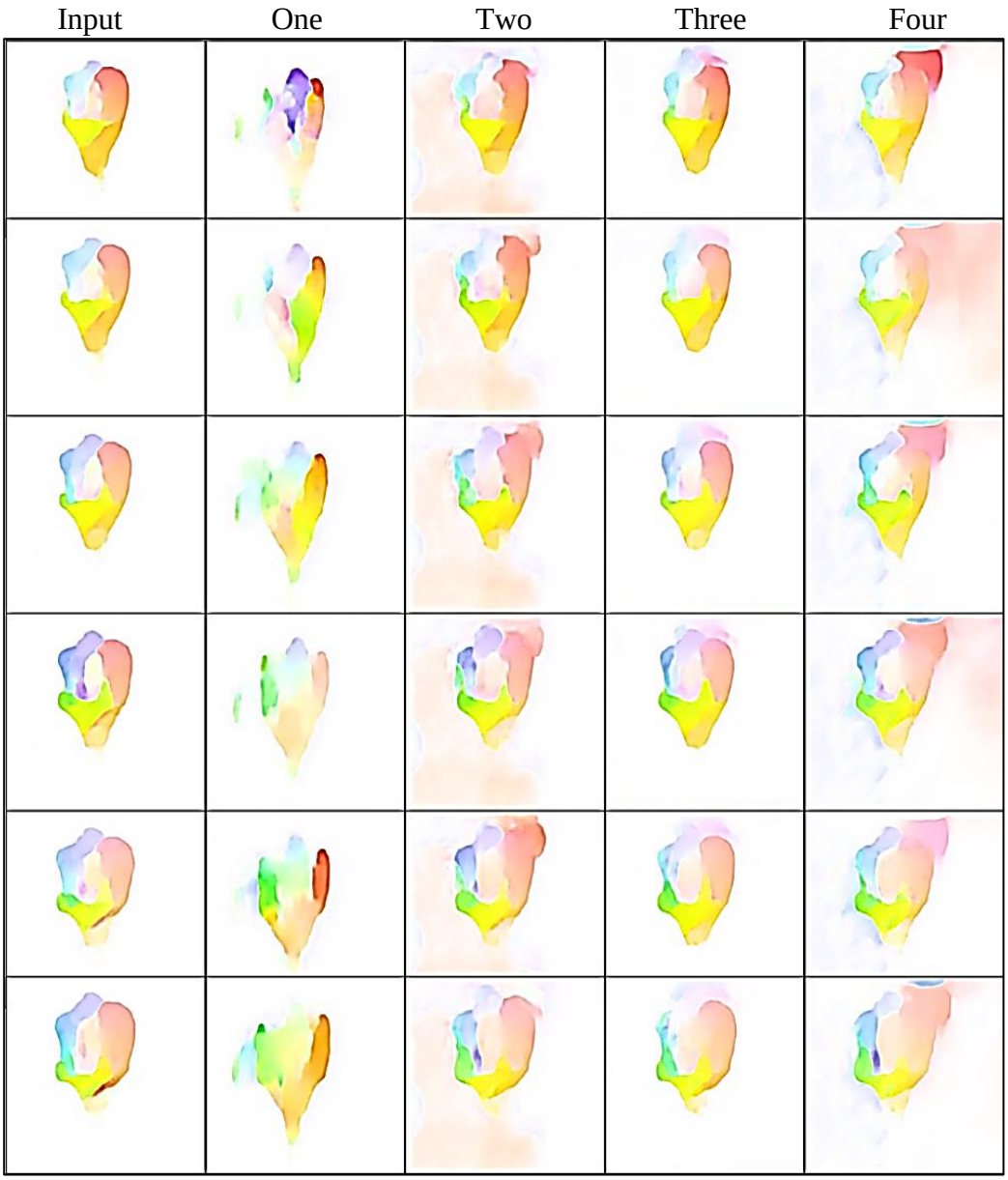
[55] S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic Routing Between Capsules,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-December, pp. 3857–3867, Oct. 2017.

[56] A. Vaswani *et al.*, “Transformer: Attention is all you need,” *Adv. Neural Inf. Process. Syst. 30*, pp. 5998–6008, 2017.

APPENDIXES

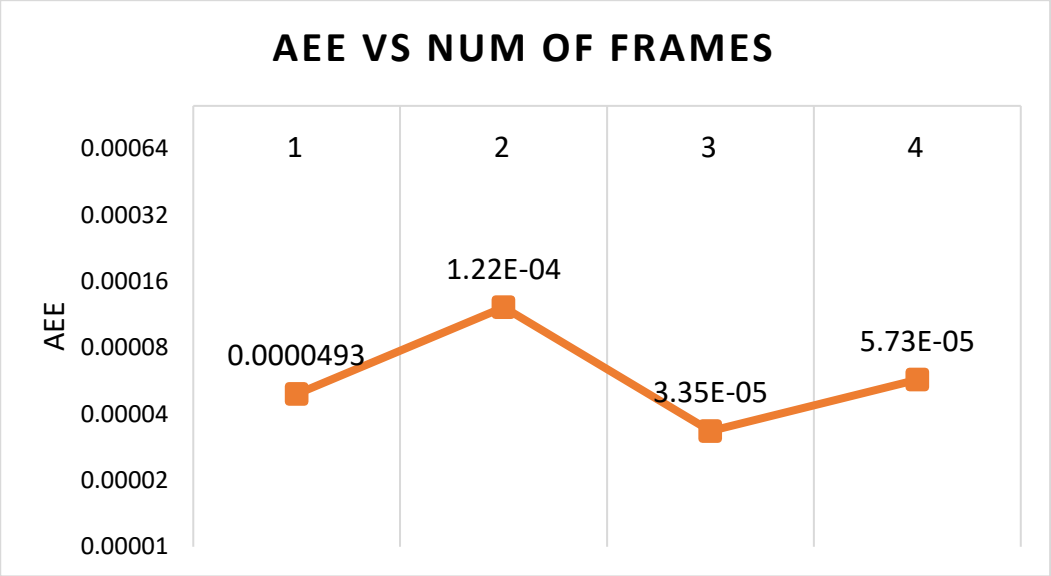
Appendix A: Ablation study

This thesis work proposed to implement a temporal discriminator network, which is a modification on the patchGAN network by concatenated previously generated output images. This Ablation study has been done to come up with the number of images the discriminator network shall consider to justify output is whether real or fake.



The optical flow between consecutive frames of (input) original image, (one) vanilla patchGAN, (two) discriminator consider two frames, (three) discriminator consider three frames, (four) discriminator considers four frames.

The figure above shows a comparison among different discriminator models and it indicates the GAN network performs well when three frames are considered by the discriminator[10]. The average endpoint error is a commonly used metrics to measure optical flow error this study test comparative discriminator networks, the result as shown in the above table indicates a discriminator network with three frames performs well comparatively.



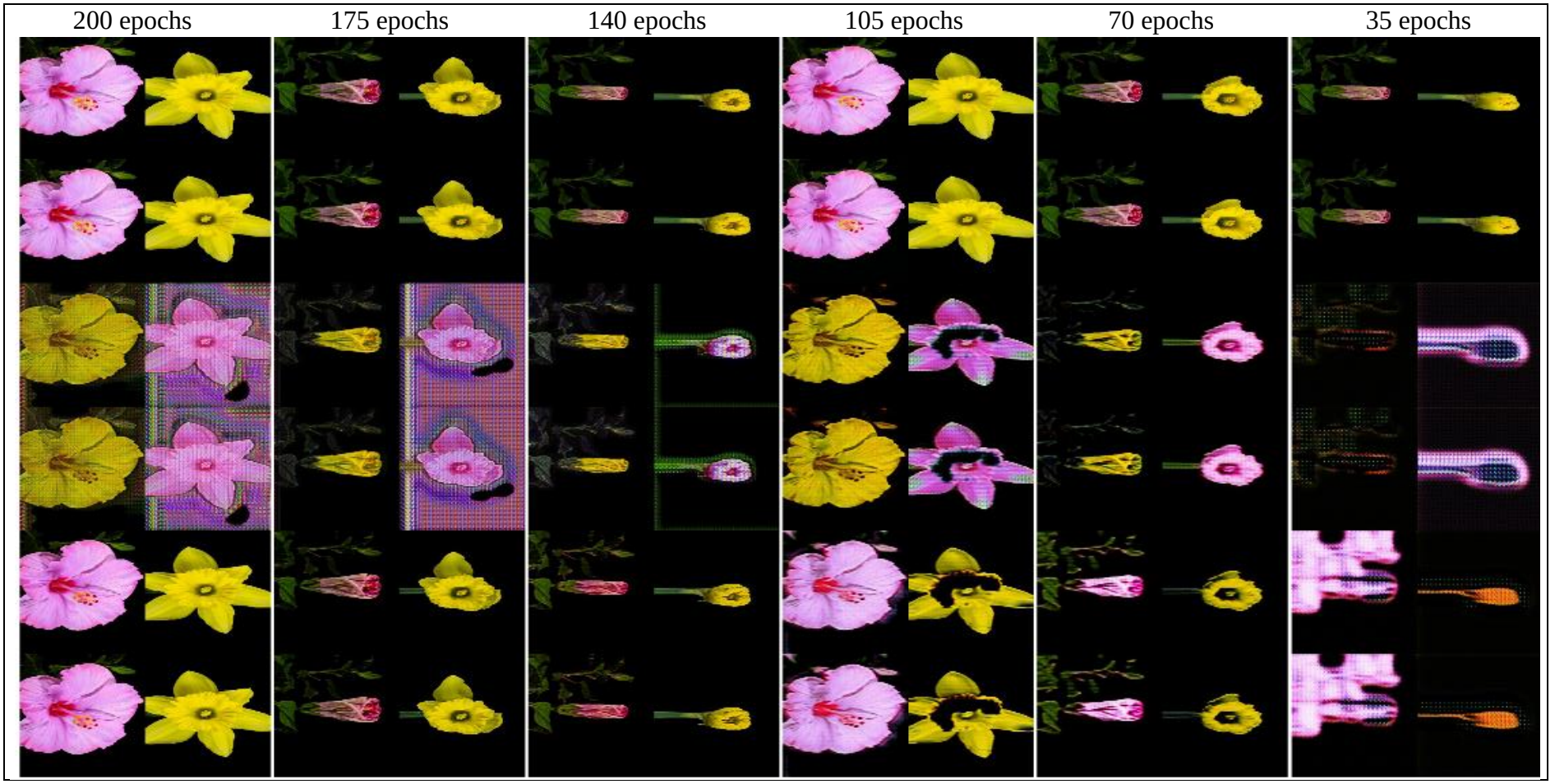
Num of frames considered	1	2	3	4
AEE	4.93E-05	1.22E-04	**3.35E-05**	5.73E-05

AEE = Average endpoint error = L2 norm.

A lower value indicates better performance.

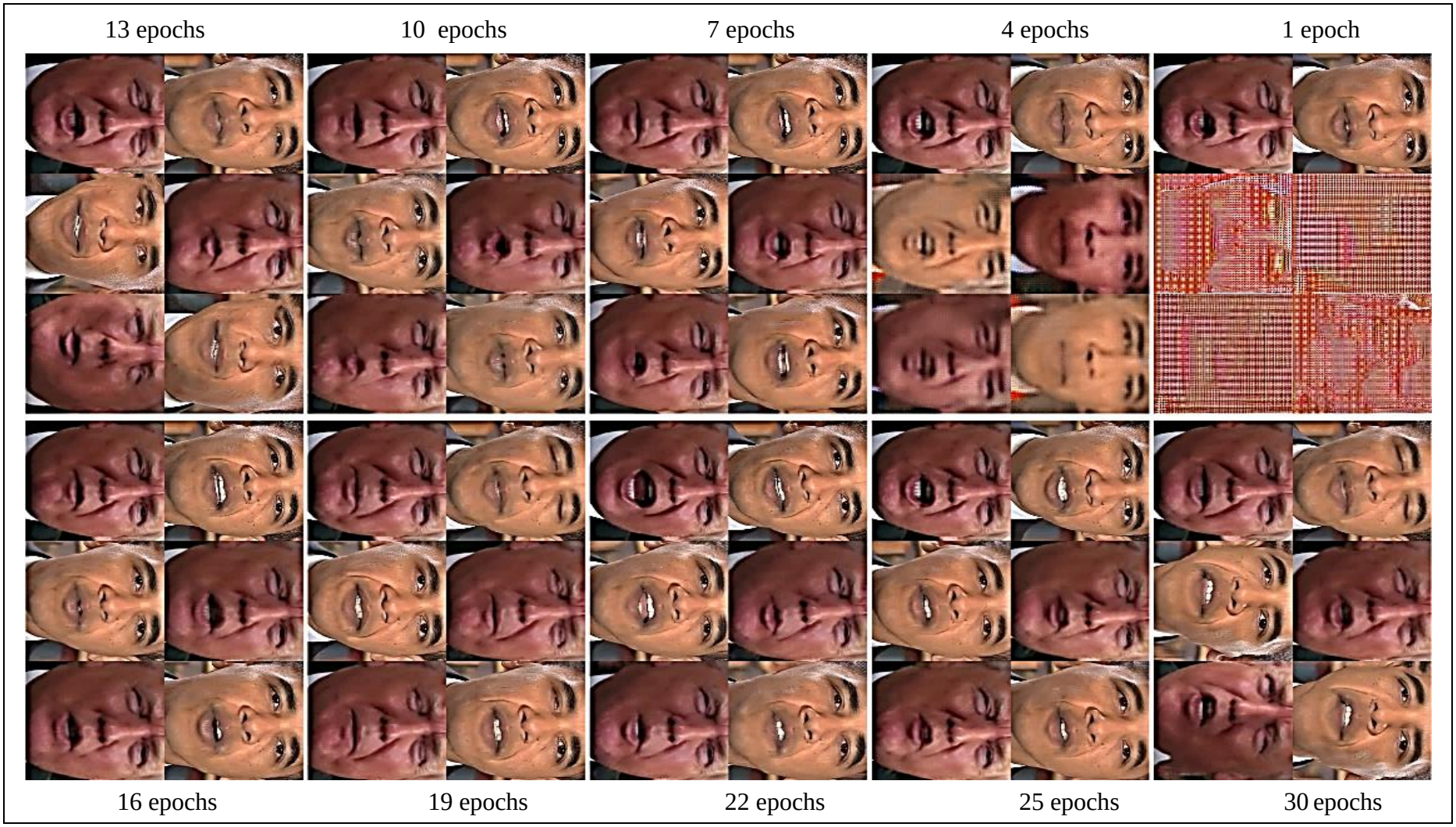
[10] For better judgement please watch: https://youtu.be/OWh-ffKROfQ

Appendix B: Result on Different epochs

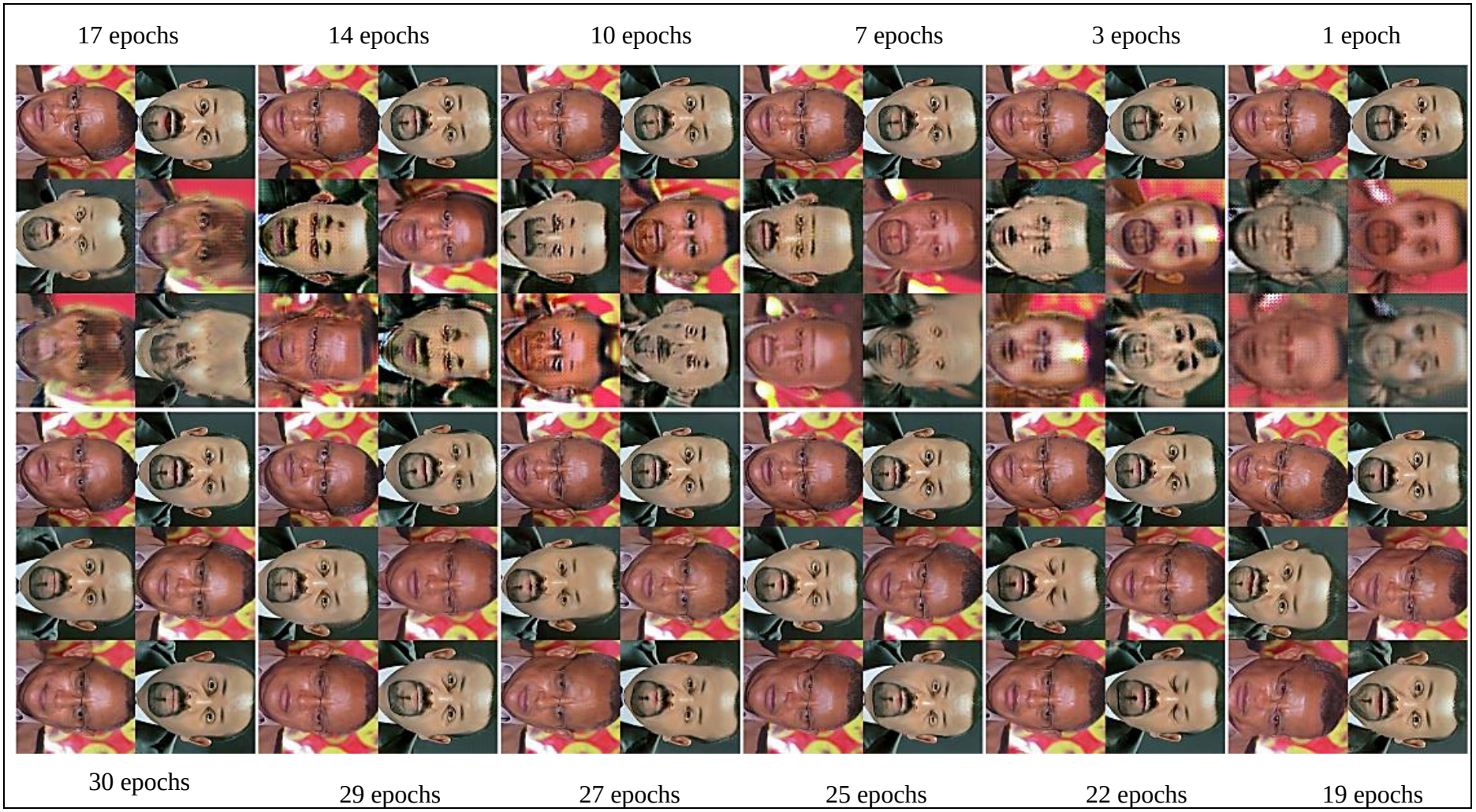


The result on different epoch on the flower dataset, the label shows the corresponding epoch

The result on different epoch on the viper dataset, the label shows the corresponding epoch



The result on different epoch on the Obama-Trump dataset, the label shows the corresponding epoch



The result on different epoch on the Adiss (አዲስ) dataset, the label shows the corresponding epoch

Appendix C: Sample Code

Train generator network

```
@tf.function
def train_G(A, A_1, A_2, B, B_1, B_2, A2B_1, A2B_2, B2A_1,
B2A_2):
     with tf.GradientTape() as t:
          A2B = G_A2B(A, training=True)
          B2A = G_B2A(B, training=True)
          A2B2A = G_B2A(A2B, training=True)
          B2A2B = G_A2B(B2A, training=True)
          A2A = G_B2A(A, training=True)
          B2B = G_A2B(B, training=True)

          M_A, M_A2B = get_content_features(A, A2B)
          M_A_A2B = identity_loss_fn(M_A, M_A2B)
          M_B, M_B2A = get_content_features(B, B2A)
          M_B_B2A = identity_loss_fn(M_B, M_B2A)

          A2B_d_logits = D_B((A2B, A2B_1, A2B_2),
training=True)
          B2A_d_logits = D_A((B2A, B2A_1, B2A_2),
training=True)

          A2B_g_loss = g_loss_fn(A2B_d_logits)
          B2A_g_loss = g_loss_fn(B2A_d_logits)
          A2B2A_cycle_loss = cycle_loss_fn(A, A2B2A)
          B2A2B_cycle_loss = cycle_loss_fn(B, B2A2B)
          A2A_id_loss = identity_loss_fn(A, A2A)
          B2B_id_loss = identity_loss_fn(B, B2B)

          G_loss = (A2B_g_loss + B2A_g_loss) +
```

```
(A2B2A_cycle_loss + B2A2B_cycle_loss) *
args.cycle_loss_weight + (A2A_id_loss + B2B_id_loss) *
args.identity_loss_weight + (M_A_A2B +
M_B_B2A)*args.identity_loss_weight

    G_grad = t.gradient(G_loss, G_A2B.trainable_variables +
G_B2A.trainable_variables)
    G_optimizer.apply_gradients(zip(G_grad,
G_A2B.trainable_variables + G_B2A.trainable_variables))

    return A2B, B2A, {'A2B_g_loss':
A2B_g_loss,'M_A_A2B_loss':M_A_A2B,'M_B_B2A_loss':M_B_B2A,'B2
A_g_loss': B2A_g_loss,'A2B2A_cycle_loss': A2B2A_cycle_loss,
'B2A2B_cycle_loss': B2A2B_cycle_loss, 'A2A_id_loss':
A2A_id_loss, 'B2B_id_loss': B2B_id_loss}
```

Train discriminator network

```
@tf.function
def train_D(A, B, A2B, B2A):
    with tf.GradientTape() as t:
        A_d_logits = D_A((A,A_1,A_2), training=True)
        B2A_d_logits = D_A((B2A,B2A_1,B2A_2), training=True)
        B_d_logits = D_B((B,B_1,B_2), training=True)
        A2B_d_logits = D_B((A2B,A2B_1,A2B_2), training=True)
        #UPDATE PREVIOUS VALUES
        A2B_2 = tf.compat.v1.assign(A2B_2, A2B_1)
        A2B_1 = tf.compat.v1.assign(A2B_1, A2B)
        B2A_2 = tf.compat.v1.assign(B2A_2, B2A_1)
        B2A_1 = tf.compat.v1.assign(B2A_1, B2A)

        A_d_loss, B2A_d_loss = d_loss_fn(A_d_logits,
B2A_d_logits)
        B_d_loss, A2B_d_loss = d_loss_fn(B_d_logits,
```

```
A2B_d_logits)

        D_A_gp = gan.gradient_penalty(functools.partial(D_A,
training=True), [A,A_1,A_2], [B2A,B2A_1,B2A_2],
mode=args.gradient_penalty_mode,numberof_img = 3)
        D_B_gp = gan.gradient_penalty(functools.partial(D_B,
training=True), [B,B_1,B_2], [A2B,A2B_1,A2B_2],
mode=args.gradient_penalty_mode,numberof_img = 3)

        D_loss = (A_d_loss + B2A_d_loss) + (B_d_loss +
A2B_d_loss) + (D_A_gp + D_B_gp) *
args.gradient_penalty_weight

    D_grad = t.gradient(D_loss, D_A.trainable_variables +
D_B.trainable_variables)
    D_optimizer.apply_gradients(zip(D_grad,
D_A.trainable_variables + D_B.trainable_variables))
    return {'A_d_loss': A_d_loss + B2A_d_loss,
          'B_d_loss': B_d_loss + A2B_d_loss,
          'D_A_gp': D_A_gp,
          'D_B_gp': D_B_gp}
```

Train Temporal Predictor

```
def P_loss_fn(real, fake):
    return tf.reduce_mean(tf.abs(real - fake))

#@tf.function
def train_P(A, A_1, A_2, B, B_1, B_2):
    with tf.GradientTape() as pt:
        A_p = Px([A_1, A_2],training = True)
        B_p = Py([B_1, B_2],training = True)
        xl1_loss = P_loss_fn(A, A_p)
        Px_loss = xl1_loss * LAMBDA
```

```
        yl1_loss = P_loss_fn(B, B_p)
        Py_loss = yl1_loss * LAMBDA

        P_loss = (Px_loss + Py_loss)* args.cycle_loss_weight

    P_grad = pt.gradient(P_loss, Px.trainable_variables +
Py.trainable_variables)
    P_optimizer.apply_gradients(zip(P_grad,
Px.trainable_variables + Py.trainable_variables))
    return A_p, B_p, {'Px_loss': Px_loss,
                      'Py_loss': Py_loss}
```

Appendix D: Human Evaluation Study Google Form Snapshot Sample

Thank you for participating Human Evaluation User study.

Thank you for participating Human Evaluation User study. I hope you had a 5 minutes.
ለተጠቃሚ ጥናት ስለተሳተፉ እናመሰግናለን ::

Please fill this quick survey and let us know your thoughts (your answers will be anonymous).
እባክዎን ይህንን ፈጣን የዳሰሳ ጥናት ይሙሉ እና ሀሳብዎን ያሳውቁን
* Required

Debretsion video

Please carefully watch next two videos, and answer the question accordingly
እባክዎን የሚቀጥሉትን ሁለት ቪዲዮዎች በጥንቃቄ ይመልከቱ ፣ እናም ለጥያቄው በዚሁ መሠረት ይመልሱ?

Video A (ቪዲዮ ሀ)

http://youtube.com/watch?v=UTYbPOlR5eE

Video B(ቪዲዮ ለ)

http://youtube.com/watch?v=YF0VMzolX4I

1. From the Above 2 Videos which one looks more real Debretsion Video? (ከላይ ከ 2 ቪዲዮዎች የትኛው ይበልጥ እውነተኛ የደብረፅዮን ቪዲዮን ይመስላል?) *

   Mark only one oval.

   - ◯ Video A (ቪዲዮ ሀ)
   - ◯ Video B (ቪዲዮ ለ)

Abiy Ahmed

Please carefully watch next two videos, and answer the question accordingly

እባክዎን የሚቀጥሉትን ሁለት ቪዲዮዎች በጥንቃቄ ይመልከቱ ፣ እናም ለጥያቄው በዚሁ መሠረት ይመልሱ?

Video A (ቪዲዮ ሀ)

http://youtube.com/watch?v=PYXhCSyts80

Video B (ቪዲዮ ለ)

http://youtube.com/watch?v=fytKHolgejg

2. From the Above 2 Videos which one look more real Abiy Video? (ከላይ 2 ቪዲዮዎች የትኛው ይበልጥ እውነተኛ የአብይ ቪዲዮን ይመስላል?) *

Mark only one oval.

- ◯ Video A (ቪዲዮ ሀ)
- ◯ Video B (ቪዲዮ ለ)

Abiy to Debretsion

Please carefully watch next two videos, and answer the question accordingly
እባክዎን የሚቀጥሉትን ሁለት ቪዲዮዎች በጥንቃቄ ይመልከቱ ፣ እናም ለጥያቄው በዚሁ መሠረት ይመልሱ?

in the next videos you will see Abiy to Debretsion video translation
በቀጣዮቹ ቪዲዮዎች ውስጥ ከ አብይ ወደ ደብረፅዮን የቪዲዮ ትርጉም ያያሉ

you will see two concatenated videos, the one in the left is original and right is fake.
ሁለት ተጣምረው የሚንቀሳቀሱ ቪዲዮዎችን ያያሉ ፣ በስተግራ ያለው የመጀመሪያው ኦሪጅናል ሲሆን በቀኝ በኩል ደግሞ ሃሰተኛ ነው።።

Video A (ቪዲዮ ሀ)

http://youtube.com/watch?v=zTWQf1xB6As

Video B (ቪዲዮ ለ)

http://youtube.com/watch?v=YSJ-vw-1OB4

3. From the Above 2 ask which looks like a more realistic and natural translation?
ይበልጥ ተጨባጭ እና ተፈጥሮአዊ የምስል ትርጉም ያለው የትኛው እንደሆነ ይመስላል? *

Mark only one oval.

- ◯ Video A (ቪዲዮ ሀ)
- ◯ Video B (ቪዲዮ ለ)

Debretsion to Abiy

Please carefully watch next two videos, and answer the question accordingly
እባክዎን የሚቀጥሉትን ሁለት ቪዲዮዎች በጥንቃቄ ይመልከቱ ፣ እናም ለጥያቄው በዚሁ መሠረት ይመልሱ?

in the next videos you will see Debretsion to Abiy video translation
በቀጣዮቹ ቪዲዮዎች ውስጥ ከ ደብረፅዮን ወደ አብይ የቪዲዮ ትርጉም ያያሉ

you will see two concatenated videos, the one in the left is original and right is fake.
ሁለት ተጣምረው የሚንቀሳቀሱ ቪዲዮዎችን ያያሉ ፣ በስተግራ ያለው የመጀመሪያው ኦሪጅናል ሲሆን በቀኝ በኩል ደግሞ ሃሰተኛ ነው::

Video A (ቪዲዮ ሀ)

http://youtube.com/watch?v=VdxmVNQejmU

Video B (ቪዲዮ ለ)

http://youtube.com/watch?v=6tF-KjOJyaE

4. From the Above 2 ask which looks like a more realistic and natural translation?
ይበልጥ ተጨባጭ እና ተፈጥሮአዊ የምስል ትርጉም ያለው የትኛው እንደሆነ ይመስላል? *

Mark only one oval.

- ◯ Video A (ቪዲዮ ሀ)
- ◯ Video B (ቪዲዮ ለ)

Appendix E: Human Evaluation Study Result

Human evaluation study on Flower-to-Flower

Timestamp	Q1	Q2	Q3	Q4
9/10/2020 10:50:22	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)
9/10/2020 11:03:42	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)
9/10/2020 11:20:28	Video A (ቪዲዮ ሀ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video A (ቪዲዮ ሀ)
9/10/2020 12:58:02	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)
9/10/2020 12:58:17	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)
9/10/2020 13:39:00	Video A (ቪዲዮ ሀ)	Video A (ቪዲዮ ሀ)	Video C (ቪዲዮ ሐ)	Video B (ቪዲዮ ለ)
9/10/2020 17:12:30	Video B (ቪዲዮ ለ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)
9/11/2020 11:33:16	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)
9/14/2020 14:41:46	Video A (ቪዲዮ ሀ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video A (ቪዲዮ ሀ)
9/16/2020 10:28:54	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)
9/16/2020 13:48:11	Video A (ቪዲዮ ሀ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video A (ቪዲዮ ሀ)
9/16/2020 17:53:04	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)
9/17/2020 15:43:22	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)
9/19/2020 16:25:49	Video B (ቪዲዮ ለ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video A (ቪዲዮ ሀ)
9/23/2020 18:12:45	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)

Q1. From the Above 3 Videos which one looks more Natural? (ከላይ 3 ቪዲዮዎች የትኛው ተፈጥሮአዊ የሚመስለው ነው?)

Q2. From the Above 3 Videos which one looks more Natural? (ከላይ 3 ቪዲዮዎች የትኛው ተፈጥሮአዊ የሚመስለው ነው?)

Q3. From the Above 3 ask which looks like a more realistic and natural translation? ይበልጥ ተጨባጭ እና ተፈጥሮአዊ የምስል ትርጉም ያለው የትኛው እንደሆነ ይጠይቁ?

Q4. From the Above 3 ask which looks like a more realistic and natural translation? ይበልጥ ተጨባጭ እና ተፈጥሮአዊ የምስል ትርጉም ያለው የትኛው እንደሆነ ይጠይቁ?

Human evaluation study on Day-to-Sunset dataset

Timestamp	Q1	Q2	Q3	Q4
9/18/2020 18:13:42	Video B (ቪዲዮ ለ)	Video C (ቪዲዮ ሐ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)
9/18/2020 18:15:54	Video B (ቪዲዮ ለ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)
9/19/2020 13:01:05	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)
9/20/2020 15:11:32	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video B (ቪዲዮ ለ)
9/20/2020 16:39:48	Video C (ቪዲዮ ሐ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video C (ቪዲዮ ሐ)
9/20/2020 16:40:09	Video C (ቪዲዮ ሐ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video C (ቪዲዮ ሐ)
9/20/2020 16:52:48	Video A (ቪዲዮ ሀ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)
9/22/2020 14:54:19	Video B (ቪዲዮ ለ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)
9/23/2020 18:25:27	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)
9/28/2020 11:43:15	Video C (ቪዲዮ ሐ)	Video B (ቪዲዮ ለ)	Video C (ቪዲዮ ሐ)	Video C (ቪዲዮ ሐ)

Q1. From the Above 3 Videos which one looks more Natural Sunset Video? (ከላይ 3 ቪዲዮዎች የትኛው ተፈጥሮአዊ የፀሐይ መጥለቅ ቪዲዮን ይመለከታል?)

Q2. From the Above 3 Videos which one looks more Natural Day Video? (ከላይ 3 ቪዲዮዎች የትኛው ተየተፈጥሮ ቀን ቪዲዮን ይመለከታል?)

Q3. From the Above 3 ask which looks like a more realistic and natural translation? ይበልጥ ተጨባጭ እና ተፈጥሮአዊ የምስል ትርጉም ያለው የትኛው እንደሆነ ይጠይቁ?

Q4. From the Above 3 ask which looks like a more realistic and natural translation? ይበልጥ ተጨባጭ እና ተፈጥሮአዊ የምስል ትርጉም ያለው የትኛው እንደሆነ ይጠይቁ?

Human evaluation study on Trump-to-Obama

Timestamp	Q1	Q2	Q3	Q4
9/21/2020 15:25:34	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)
9/21/2020 15:37:44	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video A (ቪዲዮ ሀ)
9/22/2020 14:38:53	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video A (ቪዲዮ ሀ)
9/22/2020 14:45:39	Video A (ቪዲዮ ሀ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)
9/22/2020 14:48:41	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)
9/22/2020 16:21:42	Video B (ቪዲዮ ለ)	Video A (ቪዲዮ ሀ)	Video A (ቪዲዮ ሀ)	Video A (ቪዲዮ ሀ)
9/22/2020 22:14:58	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)
9/23/2020 12:01:38	Video A (ቪዲዮ ሀ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video A (ቪዲዮ ሀ)
9/27/2020 11:16:41	Video B (ቪዲዮ ለ)	Video A (ቪዲዮ ሀ)	Video A (ቪዲዮ ሀ)	Video A (ቪዲዮ ሀ)
9/28/2020 10:31:30	Video B (ቪዲዮ ለ)	Video A (ቪዲዮ ሀ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)
9/30/2020 12:38:32	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video A (ቪዲዮ ሀ)

Q1. From the Above 2 Videos which one looks more real Trump Video? (ከላይ ከ 2 ቪዲዮዎች የትኛው ይበልጥ እውነተኛ የትራምፕ ቪዲዮን ይመስላል?)

Q2. From the Above 2 Videos which one looks more real Obama Video? (ከላይ ከ 2 ቪዲዮዎች የትኛው ይበልጥ እውነተኛ የኡባማ ቪዲዮን ይመስላል?)

Q3. From the Above 2 ask which looks like a more realistic and natural translation? ይበልጥ ተጨባጭ እና ተፈጥሮአዊ የምስል ትርጉም ያለው የትኛው እንደሆነ ይመስላል?

Q4. From the Above 2 ask which looks like a more realistic and natural translation? ይበልጥ ተጨባጭ እና ተፈጥሮአዊ የምስል ትርጉም ያለው የትኛው እንደሆነ ይመስላል?

Human evaluation study on Adiss dataset

Timestamp	Q1	Q2	Q3	Q4
9/30/2020 10:24:00	Video B (ቪዲዮ ለ)	Video A (ቪዲዮ ሀ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)
9/30/2020 10:54:37	Video A (ቪዲዮ ሀ)	Video A (ቪዲዮ ሀ)	Video A (ቪዲዮ ሀ)	Video A (ቪዲዮ ሀ)
9/30/2020 16:23:48	Video B (ቪዲዮ ለ)	Video A (ቪዲዮ ሀ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)
9/30/2020 16:52:16	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)
9/30/2020 17:58:24	Video A (ቪዲዮ ሀ)	Video A (ቪዲዮ ሀ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)
9/30/2020 18:15:52	Video A (ቪዲዮ ሀ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video A (ቪዲዮ ሀ)
10/1/2020 9:50:02	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)
10/1/2020 9:52:44	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video A (ቪዲዮ ሀ)	Video A (ቪዲዮ ሀ)
10/1/2020 9:58:28	Video A (ቪዲዮ ሀ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)
10/1/2020 10:04:44	Video A (ቪዲዮ ሀ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)	Video B (ቪዲዮ ለ)

Q1. From the Above 2 Videos which one looks more real Debretsion Video? (ከላይ ከ 2 ቪዲዮዎች የትኛው ይበልጥ እውነተኛ የደብረፅዮን ቪዲዮን ይመስላል?)

Q1. From the Above 2 Videos which one looks more real Abiy Video? (ከላይ 2 ቪዲዮዎች የትኛው ይበልጥ እውነተኛ የአብይ ቪዲዮን ይመስላል?)

Q1. From the Above 2 ask which looks like a more realistic and natural translation? ይበልጥ ተጨባጭ እና ተፈጥሮአዊ የምስል ትርጉም ያለው የትኛው እንደሆነ ይመስላል?

Q4. From the Above 2 ask which looks like a more realistic and natural translation? ይበልጥ ተጨባጭ እና ተፈጥሮአዊ የምስል ትርጉም ያለው የትኛው እንደሆነ ይመስላል?