"***Temporal Cycle consistency:*** for a video to video translation."

Kirubel Abebe Senbeto

A Thesis Submitted to the Department of Computing School of Electrical Engineering and Computing

Presented in Partial Fulfilment of the Requirement for the Degree of Masters in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

Adama, Ethiopia

December 2019

"***Temporal Cycle consistency:*** for a video to video translation."

Kirubel Abebe Senbeto

Advisor Prof Yun Koo Chung

A Thesis Submitted to the department of Computing School of Electrical Engineering and Computing.

Presented in Partial Fulfilment of the Requirement for the Degree of Masters in Computer Science and Engineering

School of Electrical Engineering and Computing

Office of Graduate Studies

Adama Science and Technology University

Adama, Ethiopia

December 2019

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 (PHD)	_____

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).

Yun Koo Chung (PHD)	_____	_____
(Advisor)	Signature	Date
_____	_____	_____
(Chairperson)	Signature	Date
_____	_____	_____
(Internal Examiner)	Signature	Date
_____	_____	_____
(External Examiner)	Signature	Date

ACKNOWLEDGMENT

First and foremost, I would like to thank the HOLY TRINITY FATHER, SON, and HOLY SPRITE GHOST the Almighty GOD who created everything seen and unseen. Also, I love to Praise the Virgin Mary the Holy Mother of 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 process of study.

I want to thank my advisor and computer vision Prof. Yun Koo Chung (Ph.D.) Special Interest Group chief for his patient advice and helpful encouragement before the completion of this thesis, as well as for his suggestion for a starting method.

I also thank the staff of the computer vision program and postgraduate students, friends, and parents for all the encouragement and guidance they have given me throughout my journey.

Table of Contents

List of Table

List of Figure

Table of Equations

Abbreviations

GAN	Generative Adversarial Network
Cycle-GAN	Cycle consistent Generative Adversarial Network
ReCycle-GAN	Recurrent Cycle Consistency Generative Adversarial Network
RNN	Recurrent Neural Network
G_x	Generator network transfer to target domain Y
G_y	Generator network transfer to target domain X
Dx	Discriminator network for Generator G_x
Dx	Discriminator network for Generator G_y
IoU	Intersection-Over-Union
MP	Mean Pixel Accuracy
AC	Average Class Accuracy
GTA v	Grand Thief Auto five(v)
3D	three-dimensional.
Pix2pix	pixel to pixel
FCN	Fully connected network

Abstract

CHAPTER ONE

1. Introduction

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.

In order to solve computer vision problems, Artificial Intelligence (AI) is an active field that concerns this topic. It started when the nascent field of computer science started to ask if a computer could become intelligent or mimic cognitive abilities that lead to knowledge such as learning, problem-solving, and reasoning. At the beginning of the development of AI, the software was hard-coded with knowledge about the world with a list of formal, mathematical rules. This approach never led to a major victory due to the struggle of describing the complexity of the world with sophisticated mathematical rules and formals. Instead of relying on hard-coded knowledge, AI systems needed a capability to extract their own knowledge. Systems started to extract patterns from raw data, this capability comes to be known as Machine Learning (ML) [1].

ML is a field with many different learning capabilities and it is still expanding. There are different types of learning problems, (they are may not be the only types) the first type is called supervised learning, that is when for every input variable (x), the output variable (y) is known so an algorithm learns to map the input to the output and since the output (correct answer) is known for every input, the algorithm is said to be supervised. Another ML problem type is when only the input data (x) is known, this is referred to unsupervised learning. The task here is to organize the data or to discover the structure or distribution of the data in order to learn more about it. Since there are no correct answers (y).

The last type is called semi-supervised machine learning and refers to problems where one part of the dataset is labeled and one part is unlabeled. This is very common because it is very expensive and time-consuming to label big datasets. Suppose a classification problem where the data set is

not fully labeled. Then unsupervised learning techniques can be used to discover the structure in the input variables. Or supervised learning techniques can be used to predict labels to every unlabeled x. Even ML plays very tremendous work but it still fails to process complex data like image and video. So as to work with complex data problems Deep Learning (DL) an option.

DL is a subfield of ML and has a special style for learning representations from data. Instead of learning one representation, DL algorithms learn successive layers of increasingly meaningful representations of the data. In other words, representations are expressed in terms of other, simpler representations. With this approach, a hierarchy of features is built and it is, therefore, possible to extract high-level features from raw data. This hierarchy of layers creates a deep Graph named Deep Learning. The quintessential example of a DL model is an artificial neural network (ANN)[2]. The research around DL exploded in 2012 when Alex Krizhevsky achieved remarkable results in the ImageNet competition (ILSVRC2012) using a convolutional neural network (CNN) [3]. But the pioneer of CNNs goes to Yann LeCun [4] when he in 1989 used a CNN to recognize handwritten digits. At that time, DL algorithms were outperformed by other ML algorithms due to two factors: the first was because of the lack of available data and the second due to bad performance in hardware. So, researchers did not see the potential of DL until a few years ago when the amount of data and the hardware performance increased. Today, DL is used in facial recognition, robotics, object detection, speech recognition, and translation.

One interesting outlet of unsupervised learning techniques is Generative Models. Usually, it is tough to analyze and understand data but generative models can do so. They are trained to discover the essence of data in some domain in command to generate similar data. This technique can be used in many tasks, for instance for image denoising, inpainting, super-resolution, structured prediction, exploration in reinforcement learning, etc. In the long run, the idea is to let computers automatically learn the natural features of data and to get a better understanding of the world.

A generative model that has recently achieved major success is called **Generative Adversarial Network (GAN)** [5] and it was introduced in 2014. GAN has been a hot research topic among computer vision researchers nowadays it learns a given data distribution in order to generate realistic-looking fake distribution. Basically, GAN contains two networks Neural networks that play zero-sum game namely called generator and discriminator- where the generator generates

fake data while the discriminator tries to classify if the data generated is tangible or forged. This work tackle domain transfer for video.

Style 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 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 *style transfer is a process of repainting a given image by style image while preserving it contain.*

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

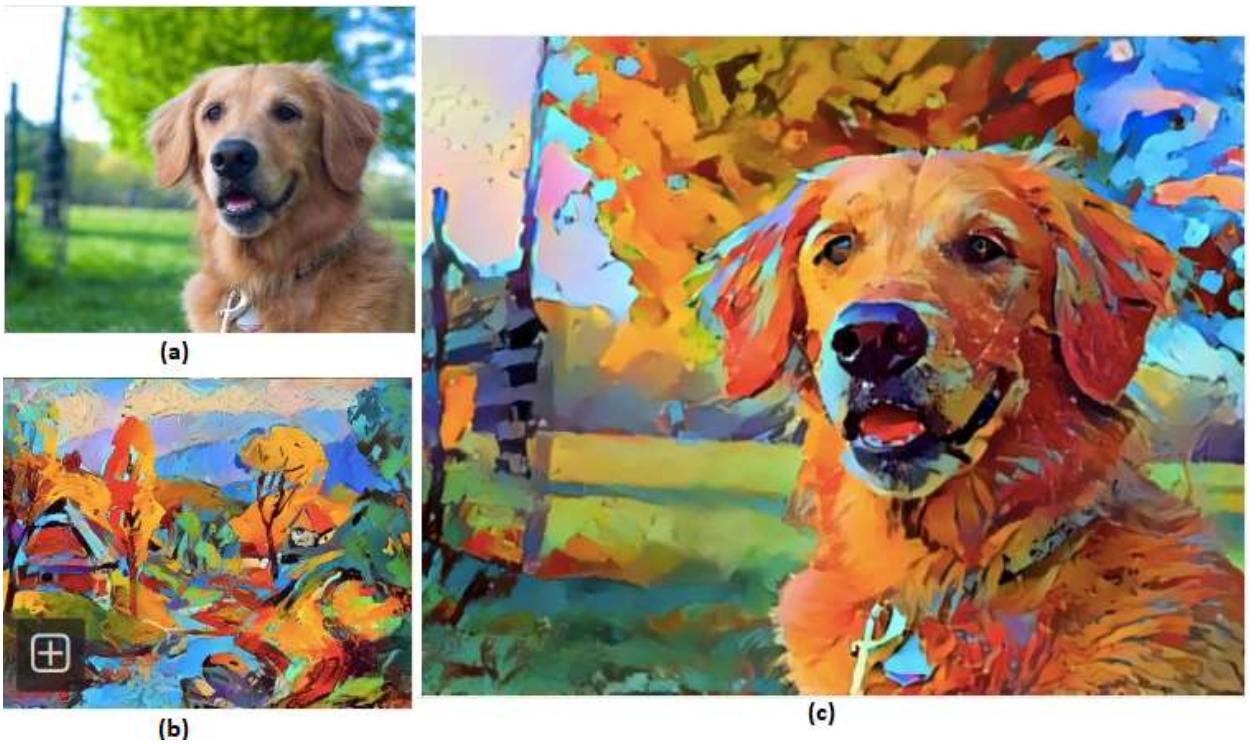


Figure 1-1(a) input image. (b) style image. (c) output image

Last year (2018) all Artificial intelligence news [6] headlines were screaming about a paint drawn 100% by AI sold $432,500 (fascinating, isn't it?). (Because of style transfer data scientist doesn't need to buy a hundred-thousand-dollar painting for decorating his living room while he can have one when he is home siting in Infront of his laptop, marvelous)

Perhaps the first successful neural style transfer paper was published in 2015 by [7]**.** After this work, many researchers came with a more realistic synthetic image. pix2pix [8] 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. [9] present unsupervised style transfer to overcome pix2pix problem due ***cycle consistency*** *–If I take an input image of horse feed it to the network it generates 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 works use the generative adversarial network for retargeting and style transfer 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.

1.1. Backgrounds

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

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.**,** in another word 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, the first network is called a Generator and 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 the Discriminator. This model's goal is 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

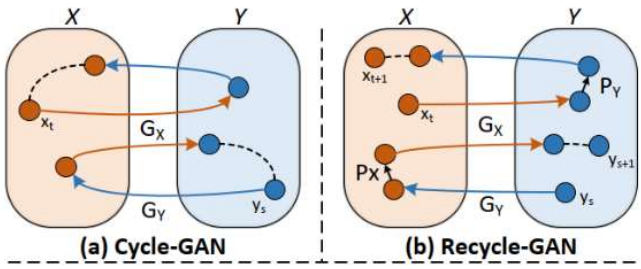


Figure 1-2 Cycle Gan vs Recycle GAN

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

Conditional GAN

The conditional GAN[10] is an extension of the [5] 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 will be 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 as shown Figure 1 3.

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 [10] they lack generalization and doesn't 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 to another pixel. [8] use conditional GAN [11], learn a mapping between paired input to the output

image. The third category is unsupervised and unpaired data domain transfer like Cycle-GAN [12] which works enforcing cycle consistency for the unpaired image.

The recent state of artwork work ReCycle-GAN by Bansal [12] motivated by [9] propose video retargeting via spatiotemporal constraint though directly synthesize future frame via temporal predictor to preserve temporal continuity But Bansal et al., clams video to video translation are **still under constraint** since their work result shows of video to video transfer has very flickering output. This proposal proposes to extend Bansal et al., work to improve temporal continuity between adjacent consecutive frames by introducing additional **cycle motion transfer constraints also proposes to introduce Spatiotemporal video to video** translation for better realistic results.

1.2. Motivation

Recent deep learning achievement has been done because of the huge amount of data available nowadays, but still there is a big problem to collect data especially when we need paired data set (such as day and night) since capturing datasets in two (or more) completely different environments is dreadful. This thesis work plan to improve video to video transfer which is one of the mechanisms to overcome such problems and it could have a great impact on computer vision and deep learning society.

Even though this work focused on unpair dataset there is one major addition this thesis work could back specifically in data augmentation to improve data insufficiency in deep learning so as to improve convergence experience. since there is still no enough data in many computer vision problems. this issue remains one of the extremely challenging problems in computer vision when the real-life scenario is considered. This study tries to solve the video temporal discontinuity problem by extending solutions presented in previous works [12], [13] explicitly for a video to video translation.

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. Let we have two videos archives in source and target domain $X = \{x\}_{t=1}^{T}$ and $Y = \{y\}_{s=1}^{S}$ respectively, cycle constraint enables an image to image translation in mutually frontward and backward mapping. There are two mapping functions $G_x \; and \; G_y$ mapping from domain $X \rightarrow Y$ and $Y \rightarrow X$ correspondingly form target domain to

source and vice versa. $G_x(x_t) = \dot{x}_t$ where x_t is input video frame at time t and $\dot{x}_t$ is a synthetic frame in X domain same is true in Y domain. Cycle consistency constraint $G_y(\dot{x}_t) = x_{t\,rec}$ s.t $x_{t\,rec} \approx x_t$ as well as $G_y(\dot{x}_t) = x_{t\,rec}$ s.t $y_{s\,rec} \approx y_s$.

Besides the preservation of cycle consistency in each frame this work-study mapping cycle motion consistency between consecutive frames in both domains. Meaning let optical flow between x_t and x_{t+1} is f_t and optical flow between $x_{t\,rec}\ and\ x_{t\,rec+1}$ is $f_{t\,rec}$, then, temporal cycle consistency need to enforce motion consistency via minimizing the difference between $f_t\ and\ f_{t\,rec}$. Recycle-GAN [12] claims **"*video to video translation is under constraint*"** this work proposes toward add temporal cycle consistency to the extended video to video translation to see more constraints in its result.

To do so an extensive experimental attempt was done with the purpose of answering the following research question.

» ***Adding additional constraints can improve temporal coherency for video translation?***
» *What is the effect of temporal cycle consistency on the unsupervised video to video transfer?*

1.4. The Objective of the Thesis

1.4.1. General Objective

The general objective of the study is to design and implement **Temporal Cycle consistency** for the **video to video translation**. This work is motivated by [12] ReCycle-GAN.

1.4.2. Specific Objective

The following specific objectives will be addressed to achieve the general objective.

» Reviewing related works to understand the area and the works that are done by others.
» Gather dataset for training and testing.
» To preprocess the dataset in order to enhance its quality.
» To extract temporal information from the video.
» To add learning constraints to the network.
» To design a deep learning video translation model using Keras and TensorFlow framework.

» To blend spatial Information and Temporal Information.
» To train the model using a proper dataset.
» To test the trained model with a test set.
» To assess the performance of the model.

1.5. Research Methodology

The following methods and techniques are applied in order to meet the objectives of this study.

1.5.1. Data Collection

This study uses a machine learning approach to solve the problem, so data is an essential part of the study. Videos (sequence of Images) are collected for both training and testing. Those data (Datasets) are collected directly from the internet (available popular unpaired dataset) for the purpose of the study. Besides the popular datasets available on the internet this work plan to collect local video dataset to inference the study. Most of these videos were long and made up of several frames, each shot being a different scene.

1.5.2. Literature Review

This study uses a literature review to enhance the research. Recent related literature is reviewed to get an insight into current trends and methods to solve the problem at hand. Necessary documents and tools are also reviewed for the development of the prototype.

1.5.3. Evaluation

The result will be analyzed to describe the performance of the proposed architecture on a test data set. The performance of this work will be analyzed in real-world scenarios videos from the dataset.

1.5.4. Implementation Tools

For the development of the deep learning network architecture in addition to reporting this thesis work finding the following tools and software will be used.

» OpenCV, TensorFlow, and Keras API, MATLAB will be used for modeling networks, coding the as well as training and testing.

» Microsoft Word, PowerPoint, and Grammarly are software plain to use for editing, Presentation, and check Plagiarism checking.
» GPU to train the network more efficiently.

1.6. Scope and Limitation

1.6.1. Scope

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

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

1.6.2. Limitations

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

- ***One to many video to video translation*** is **not** a part of this work. The network will be trained to translate from one domain image to another domain, which is one to one correspondence (Doesn't consider multi-domain translation).
- The video does **not zoom in** or **out** throughout the whole process.

1.7. Organization of the thesis

The remainder of this thesis is organized as follows:

Chapter Two: discusses the background literature and related works regarding the 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 solution in detail and the working environment setup. Discuss the specification of an image to image translation networks and temporal

information blending with the spatial model. Flow chart and pseudocode for implementation, training, and testing with mathematical correspondent descriptions have been discussed.

Chapter Five: Explains how the desired proposed solution is implemented. The working environment, cycle-GAN implementation, with training pseudocode 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

2.1. Introduction

The most impressive success in Deep Learning has, so far, involved discriminative models, i.e. models that map the dependence of unobserved target variables (y) on observed variables (x) – Classification problem. In simple terms, discriminative models suppose outputs based on inputs without considerate about how the input was generated. In another sense, Generative models are opposed to discriminative models, which maps how the input data was generated.

GANs (Generative Adversarial Networks) [5] is a fit into the generative type of network. GANs are taught to generate synthetic data alike to known input data. A GAN model consists of two types of neural networks inside, a generator model and a discriminator model. The two Deep Neural networks have an adversarial relationship where they fight against each other[1]. The

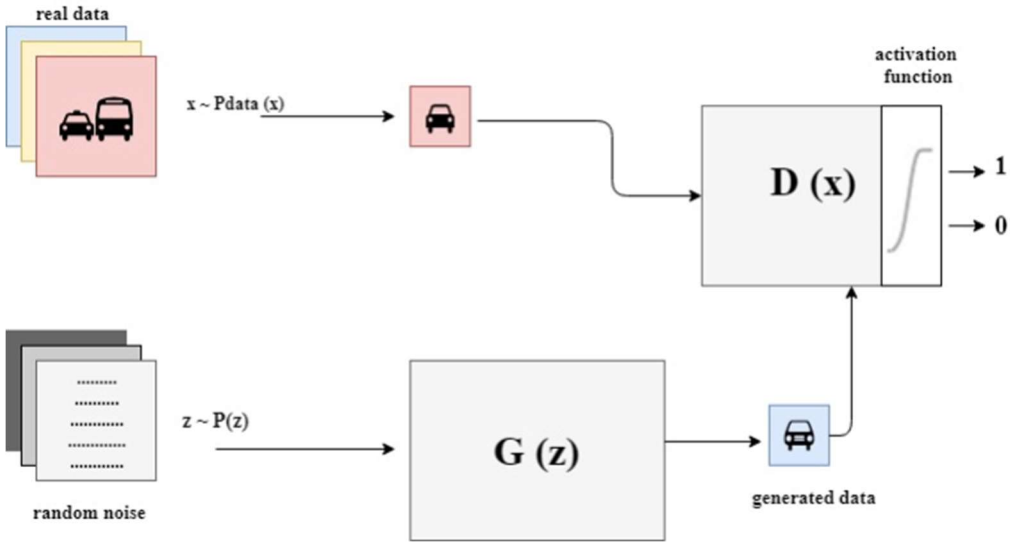


Figure 2-1 GAN framework structure GAN framework consists of two networks: Discriminator (D) and Generator (G)

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

generative model learns to mimic data while the discriminative model learns to determine whether a sample is from the model distribution $p(z)$ or the data distribution $pdata(x)$.

During training, both models improve their methods until the artificially generated data are indistinguishable from real data. In this chapter, the paper briefly describes the technologies, methods, and frameworks mentioned throughout the thesis.

2.2. Inside GAN

In this section let 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 matrix 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 [5]:

$$L_{adv} = \min_G \max_D E_{x \in X}(\log D(x)) + E_{z \in Z}\left(\log\left(1 - D\left(G(z)\right)\right)\right)$$

Equation 1 Adversarial loss function

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 can be seen as 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 training GAN.

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 so let's dive into it. But 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 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. 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 of 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 minimax function (equation 1). Regarding GAN at Nash equilibrium discriminator no longer able to distinguish between real and fake samples so it randomly classifies $(accuracy = 50\%)$.

2.2.2. Conditional GAN

Even though GAN models are able to produce new random possible examples for sample data, there is no means of monitoring the types of images produced. But the network tries to figure out

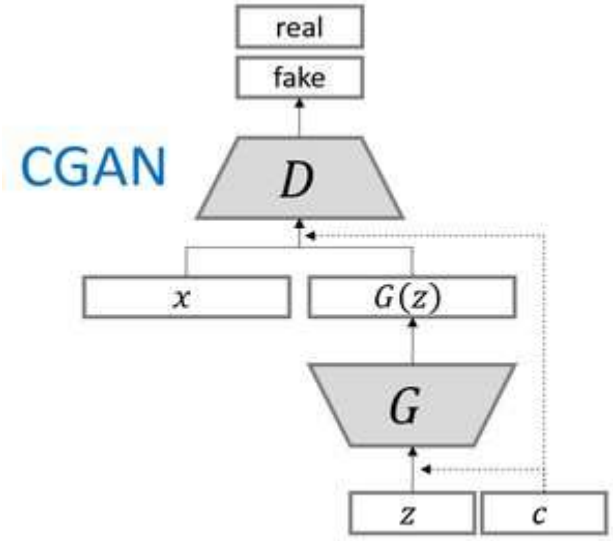


Figure 2-2 cGAN Architecture

the composite association between the latent space input to the generator in order to mimic the real dataset and the generated images[5], [14].

Mirza et al propose The conditional generative adversarial network, or cGAN [11] 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 [8], [12]. The next section will discourse about Image to image translation.

2.3. Variational Autoencoders

Variational Autoencoders (VAEs) is a method that uses convolutional neural networks to generate data. An autoencoder can be described as a network that learns how to compress data in a way that enables it to be reconstructed again. The purpose of the autoencoder is to minimize the dimensionality of the data, while still being able to reconstruct it with as little loss as possible. Similar to a typical autoencoder the VAE also consists of an encoder and a decoder. The purpose of the VAE is, therefore, to learn the probability distribution of the data. A data sample can then be generated by drawing a sample from the probability distribution and feeding it to the decoder.

2.4. GAN based Image-to-Image Translation using unpaired training data

Let start by Abto software AI software company from Europe say about style 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****"* [15] I was wondering too winter to summer Absolutely fascinating.

Recent advancements in GANs [5] empowers style transfer models to create realistically looking [7]–[9], [16] adapted image (2-4 B show image to image transfer from sunny to rainy).

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

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 2-4. The major 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[8] is one of them which is a conditional Generative model by Isola et al train in a supervised manner using a paired dataset that fits into a supervised image to image translation. Pix2pix as the name indicates learn to map pixel from the first image to the second one.

Because in reality pair datasets are very rear and expensive Zhu et al. came up with Cycle-GAN [9] which was invented to learn bidirectional mapping in 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 very old news in natural language processing, the following example gives a simple illustration. Assume using language translation from English to Amharic in both directions. When the user input "My name is Abebe" the model should generate "ስም አበበ ይባላል።" perhaps if the user translates "ስም አበበ ይባላል።" to English back again it should generate the original text "My name is Abebe". Meaning the difference between the original text and regenerated text should be minimum. I use google translator to demonstrate this example as shown in figure 2-3.

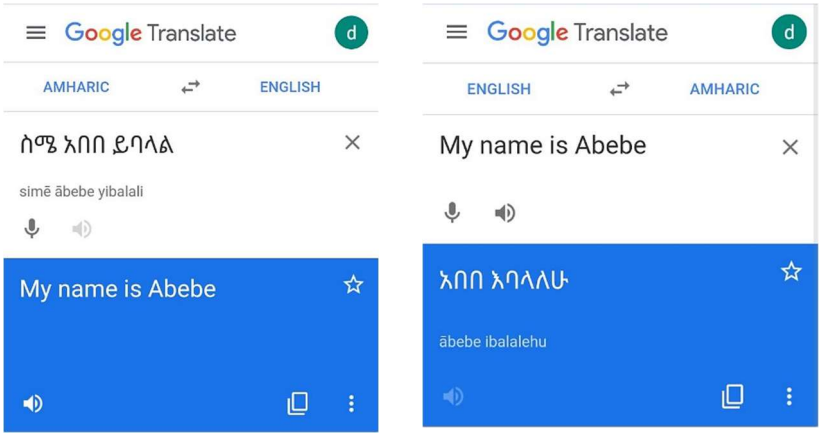


Figure 2-3 Amharic to English language translation using google translator (Example).

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's see it in bit detail using a table 2.

Table 2 Cycle-GAN generator and discriminator operation.

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

$A\ and\ B\ are\ real\ image\ from\ domain\ A\ and\ B\ respectively.$

$While\ \dot{A}\ and\ \dot{B} are\ generated\ images\ from\ G_{A_B} and\ G_{B_A} respectively.$

$R.B\ \dot{A}\ is\ in\ domain\ B\ whwere\ as\ \dot{B}\ in\ domain\ A$

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 x_{rec} the difference should be the minimum ($x \approx x_{rec}$). x_{rec} 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 from input and output image

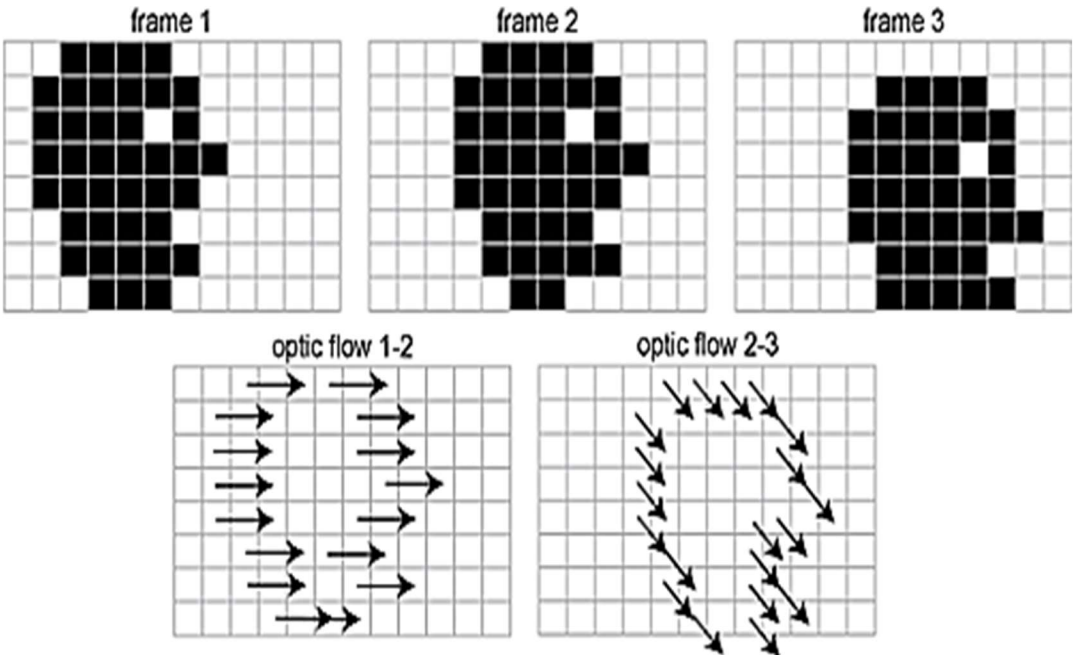


Figure 2-5 General Optical flow Computation

2.5. 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 has a relation between them. This approach is non-trivial since this is key issues that underlie the flickering [17], [18] effect in the output video. To overcome the flickering effect Chen et al [18] consider temporal information along with spatial information. Specifically, they exploit previous frame optical flow to warp the current frame towards impose temporal constraints. Let see what is temporal information and different mechanism to extract.

2.6. Problems in Translation Networks

Highly expressive parametric models have achieved great success in machine learning [19], where learning goals and measurement measures are typically well defined and easily calculated. 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 are the most common problems for video translation.

Let say we have 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. $X = \{x_1, x_2, x_3 \dots, x_n\}$ where x_i are the i[th] frame of video X. Each frame may contain various objects. $\{O^1_{x_1}, O^2_{x_1}, O^3_{x_1} \dots O^n_{x_1}\} \epsilon\, x_1$

» **Object disappearance**: is a problem object O_i in a given video frame x_t in domain A shall also appear in translated appear $\dot{x_t}$ in another domain image. meaning if a car appears in x_t is should also appear in $\dot{x_t}$. Mathematically,

$$if\ \{O^i_{x_1}\} \epsilon\, x_1, then\ \{O^i_{\dot{x_1}}\} \epsilon\, \dot{x_1} where: x_t = G_{AB}(x_t)$$

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

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

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

» **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.

The problems described above are appropriate for the problem of translation, where only spatial transformation is considered. For an approach to mixing spatial and temporal learning functions, this thesis work gets a better result.

2.7. Temporal information

Temporal refers to time-domain where in our case it can be seen as a relation between sequence of frames in the video while Spatial refers to RGB space frames. Spatiotemporal or Spatio-temporal is used in the study of information as data is gathered over time and space. Straight

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

forward approaches basically fail because they can't consider both domains. Temporal information for video can describe a phenomenon in a certain pixel location with position change in time.

For a video to video translation, we have various[4] options to represent motion information. The next section would discuss those topics. 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 as optical flow and pose estimation, and then the model imposes temporal information on the generated frame. The other tacit one does not specifically collect temporal knowledge. Examples could be 3D Conv-nets, RNNs, and temporal constraint models. Indeed, some of the works have been done blend the above techniques, such as Park et.al. [13].

2.7.1. Optical flow

Optic flow is the change of structured pixels with specific intensity in successive images, or in another word, Optical flow is the motion of objects among successive frames, caused by the comparative movement among the object and camera. This make optical flow an ideal for encoding temporal information[12], [13], [20].

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 optic flow, which is a computationally challenging task.

Nowadays, better computational resources and Recent advancements in Deep learning enable researchers to estimate optical flow. Generally, such approaches take two video frames as input to output the optical flow (color-coded image), which may be expressed as $(u, v) = f(I_{t-1}, I_t)$ 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 I_{t-1} (frame at time $= t - 1$) and I_t (frame at time $= t$) as input.

Computing optical flow with deep neural networks [21], [22] requires huge amounts of training data which is principally hard to obtain. This is because marking optical flow video footage

[4] only significant related approach for this kind of problem discussed (regarding to temporal information extraction for video).

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, [21], [23], [24] 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 [25], an open-source CGI movie with optical flow labeling rendered for numerous sequences, and Flying Chairs [23], a dataset of numerous chairs hovering around random backgrounds both generated from the virtual world using Computer Graphics.

2.7.2. 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. Particularly human pose estimation is used in transferring motion from one person to another as used [26], to transfer motion between different domain videos specifically for animating static image by driving motion as shown in figure 2-6 [27] and facial expression transfer [28] between source and the target person.

We have 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.

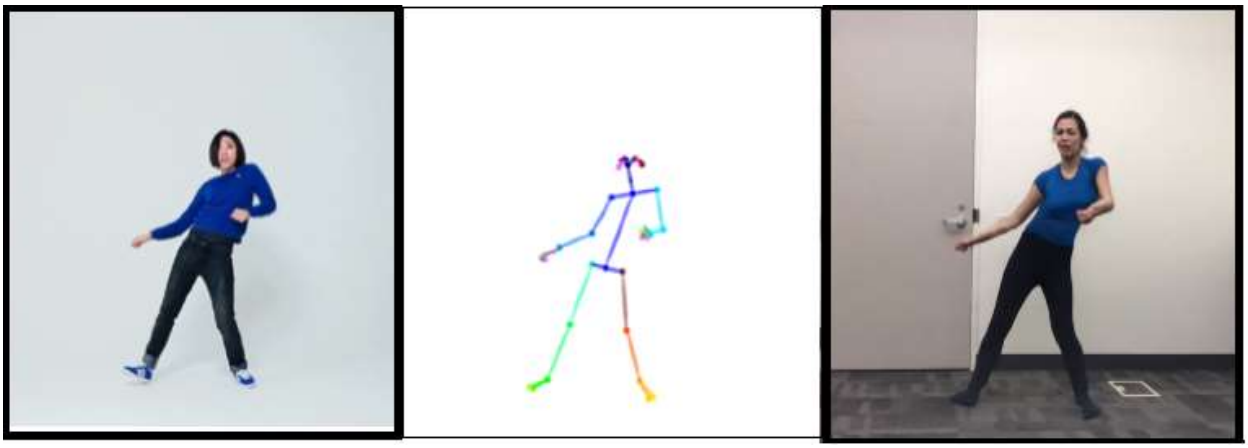


Figure 2-6 pose extraction

2.7.3. 3D convolutional tensor

The 3D convolutional tensor mechanism is one of the orthodox methods which basically doesn't consider temporal information explicitly. Since it considers presenting video scene [29] 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 basic reasons requires a high-efficiency machine, and the network becomes a fully black box. Meaning hard to tune parameters basically done in training Deep Learning models.

2.7.4. 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 take into account all previous frames as an input to minimize temporal residual error [20].

2.8. Recent works on Spatio-temporal information

In the previous section, we discussed the temporal information (motion information) extraction mechanism. But since video consists of both temporal and spatial information, we need to discuss mechanisms to get the advantage over an early approach (spatial only). So instead of applying Spatio 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. This approach is non-trivial since the key issues which motivate the flickering effect in the results output video[17], [18].

To overcome the flickering effect Chen et al [18] 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 doesn't consider newly introduced pixels in given frame scene). another fine work by Chen et al [30] MoCycle-GAN introduces temporal motion translation to transfer estimated motion from source to target video while preserving temporal consistency. this work also relives the temporal cycle constraint for motion reconstruction.

The current state of artwork [12] ReCycle-GAN further extend cycle consistency constraint by intercorporate it with temporal predictor network to predict over spatiotemporal predictor though directly synthesize future frame via temporal predictor to preserve temporal continuity. Another recent quality work by [20] proposes an optical flow residual error between ground truth and warped frame mechanism to guarantee the local and global consistency to overcome the temporal flickering and motion inconsistency between frames.

2.9. Recent work summary

the following table illustrates a summary of previous works on the video to video translation[5].

Table 3 previous works summary on the video to video translation.

	Dataset (Data collection)	Architecture	Temporal information modeling	Network constraint applied	Evaluation matrix used	Limitation
Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks	Cityscapes, Horse to Zebra and Apple to Orange, Summer to Winter Yosemite, etc..	Cycle Consistency Constraint.	No temporal Information is considered.	-	FCN, AMT perceptual studies	Framewise image to image translation.

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

Recycle-GAN: Unsupervised Video Retargeting ReCycle-GAN [12]	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	Temporal predictor basically fail to correctly predict, and no cycle consistency temporal cycle considered
Mocycle-GAN: Unpaired Video-to-Video Translation MoCycle-GAN [30]	Flower video and viper dataset	Cycle-GAN with motion translator-based motion cycle consistency	Flownet2.0 with motion translator network	Generator Network	Human evaluation, IoU, pixel accuracy, Average class accuracy	Explicit motion translator

Animating Arbitrary Objects via Deep Motion Transfer Monkey-Net[28]	UvA-Nemo, Tai-Chi, and BAIR robot pushing datasets	RNN based Dense motion predictor and motion translation network	Keypoint detector with motion transfer network based on motion heatmap	Generator Network	L1, AKD, MKR, AED, and FID	No random input size.
Unsupervised Video-to-Video Translation Dina [29]	Volumetric MNIST, GTA segment to video and MRI-to-CT	3D Cycle-GAN	The network implicitly learns to form input video (3D-Conv- net)	Generator Network	Human evaluation, pixel accuracy, and L2 error between original and retranslated image	3D tensor fails for temporal learning consistency between frames.

Preserving Semantic and Temporal Consistency for Unpaired Video-to-Video Translation [13]	Viper dataset	RNN based Cycle-GAN with flow estimator network and consistency warping network	Flownet2.0 base temporal fuse with spatial for improving occlusions problem	Generator Network, Use [31] to further reduce the Temporal warping error.	mIoU, fwIoU, and pixel accuracy	Only consider local temporal consistency
Video-to-Video Translation with Global Temporal Consistency[20]	DAVIS 2017	RNN based Cycle-GAN, and RNN based Discriminator for global temporal consistency	Flownet2.0, temporal residual error minimizer	Generator + Discriminator Network	Peak Signal to Noise Ratio, Region Similarity, and Contour Accuracy	Complex architecture hard to train doesn't consider temporal cycle continuity

As shown in the above table, researchers design complex architectures used in previous works so as to learn a mapping from a domain to domain in an unsupervised manner.

CHAPTER THREE

3. Materials and Methods

3.1. 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 machine 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 [12]. In addition to inference the result of this work I collect a local dataset called **አዲስ** (Adiss).

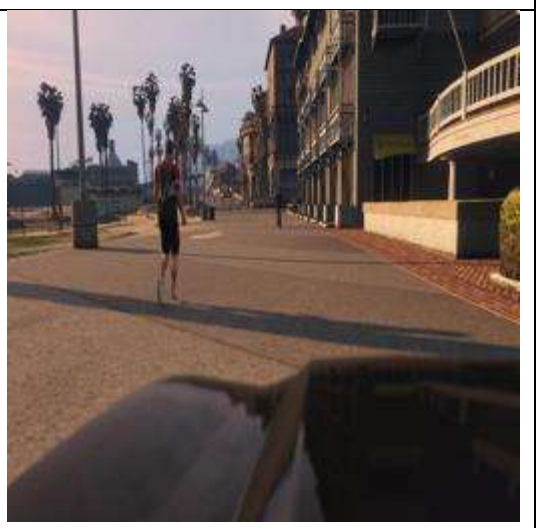
- **Obama-trump:** is a recently released dataset for style transfer and video retargeting. This dataset contains a sequence of images of Obama and Trump making an interview (though at a different time and completely talk about different things). Each frame is 256 x 256 and about 8617 images are included.
- **Flower Video 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 sync. The resolution of the respective videos is 256 × 256. This work use flower-to-flower for domain transfer between dissimilar types of flowers.
- **Viper:** 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).

Table 4 Training Dataset Sample (from Obama - Trump and Flower Datasets)

Flower1					
Flower2					
Obama					
Trump					

Table 5 Viper Dataset Sample Examples

Environment	Sample1	Sample2
Rain		
Snow		
Sunset		

Night		
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 tools that are relevant to the research. The following sections give a brief detail about these development tools.

3.4. Design tools

Design tools are mediums that are used for the creation, presentation, and interpretation of design concepts. Edraw Max [32] 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 [32].

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

3.5. Prototype development framework

3.5.1. 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. Supports artificial learning, deep learning, and versatile numerical computing [33]. The following diagram displays the power score of the deep learning system based on application, popularity, and interest [34].

The following diagram demonstrates the power score of the deep learning system based on application, and popularity.

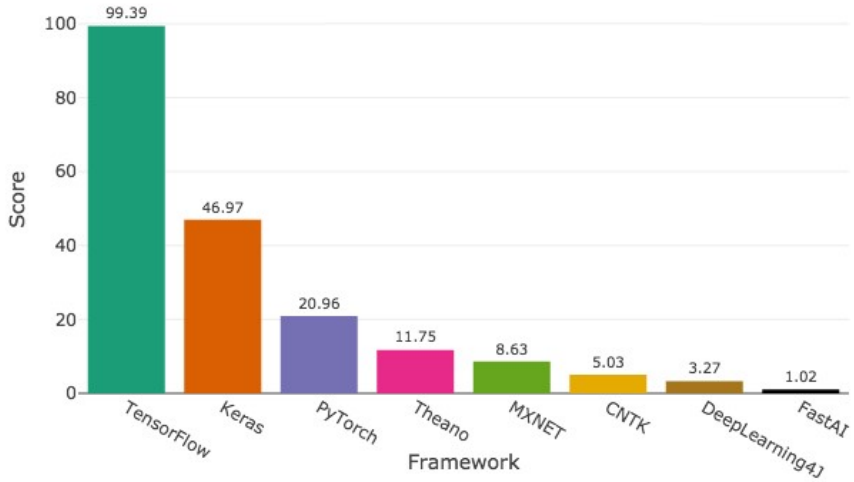


Figure 3-1 Deep Learning Framework comparison.

As shown in the above diagram, 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.

3.5.2. OpenCV

OpenCV is an open-source computer vision software library intended to provide a shared infrastructure for image processing and computer vision applications [35]. 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.

3.5.3. MATLAB Deep Network Designer

MATLAB deep network designer [36] 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 specially for complex network deep architectures and GAN networks. This even further helps Developers to track and debug errors on the premature design stage.

3.6. Baselines

To validate our model's effectiveness, we equate it with models that dwell on translating video with GANs. Since our model architecture is based on Recycle-GAN and takes as input unpaired video data, we chose Cycle-GAN [8] and Recycle-GAN [11] as the baselines for our experiments.

- » Cycle-GAN [8] 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 better understand the spatio-temporal constraint effect.
- » ReCycle-GAN [11] uses two generators and two predictors for video translation. It puts forward a recycle loss to work with cycle loss and recurrent loss for content conversion and style preservation, taking into account 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.

3.7. Evaluation methods

The result will be analyzed to describe the performance of video to video translation model 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 succeeding is a qualitative analysis metric to evaluate this work.

3.7.1. Human evaluation study

This evaluation method uses volunteers to assess whether the given video is real or fake after he/she sees random real and fake videos to determine whether or not the generated data is any good. The normal 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 and 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**.

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

$$\mathrm{IS}(G) = \exp\left(\mathbb{E}_{\mathbf{x}\sim p_g} D_{KL}(p(y|\mathbf{x}) \parallel p(y)) \right),$$

Equation 2 Inception Score

images. The IS (inception score) for short, **measure the variety and the quality of the created images**. The superiority of the model is good if it has a high inception score.

CHAPTER FOUR

4. Proposed work.

4.1. 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, so with the purpose of achieving this objective image to image translation, temporal information extraction and Spatio-temporal information fusion are main building stones. This chapter can be ideally portioned to 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 our model.

4.2. Model Architecture.

The model architecture has directly influenced by the architecture defined in *"Unpaired Image-to Image Translation using Cycle-Consistent Adversarial Networks"* [9] and *"Recycle-GAN: Unsupervised Video Retargeting"* [12] for Learning Domain Translation. Adjustments have been

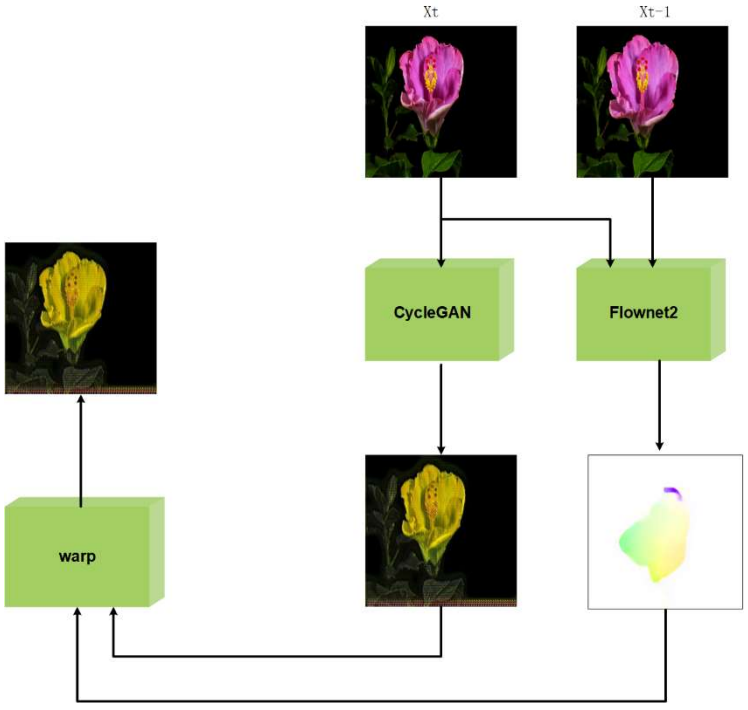


Figure 4-1 Generator Network Architecture

made to the discriminator network and additional losses have been applied to the Generative

Network including Temporal warping. Section 5.4 addressed the depth Implementation detail of the proposed work.

4.3. Model learning functions

The key objective of this thesis work is to optimize the use of space-time knowledge, and in order to address our research query, we add loss functions and change the discriminator network so that it can address temporal coherency to the Cycle-GAN and ReCycle-GAN. As our architectural model, Cycle-GAN and Recycle-GAN are based.

We seek 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} : X $\rightarrow$ Y. Note that our model uses sequential unpaired image data as input during training.

4.3.1. Proposed Network Learning Function.

Because we follow the GAN architecture, the vanilla adversarial loss is also used in our work, called $\boldsymbol{\ell}_{\text{GAN}}$. And the cycle consistency loss $\boldsymbol{\ell}_{cycle}$ in Cycle-GAN [9] is adopted. Besides, the recurrent loss $\boldsymbol{\ell}_{recurrent}$ and the recycle loss $\ell_{recycle}$ in Recycle-GAN [12] are also leveraged. Meanwhile, this work introduces constrain $\ell_{featurePreserving}$ to impel the model and improve the whole translation. The full loss function of our work is as follows:

$$\min_{G,P} \max_{D} \boldsymbol{\ell} all(G,P,D)$$
$$= \boldsymbol{\ell}_{\text{GAN}}(G_x, D_x) + \boldsymbol{\ell}_{\text{GAN}}(G_y, D_y) + \alpha\boldsymbol{\ell}_{cycle}(G_x, G_y) + \alpha\boldsymbol{\ell}_{cycle}(G_y, G_x)$$
$$+ \beta\boldsymbol{\ell}_{recurrent}(P_x) + \beta\boldsymbol{\ell}_{recurrent}(P_y) + \gamma\boldsymbol{\ell}_{recycle}(G_x, G_y, P_x)$$
$$+ \gamma\boldsymbol{\ell}_{recycle}(G_x, G_y, P_y) + \theta\boldsymbol{\ell}_{featurePreserving}(G_x, G_y)$$
$$+ \theta\boldsymbol{\ell}_{featurePreserving}(G_x, G_y)$$

Where $\alpha, \beta, \gamma, \delta$ are used parameter of learning. Indeed, the network needs more learning constraints, the aim of which is to demonstrate a significant consistency. Let us look in detail at all loss constraints.

One thing to keep in mind is that the translated image 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 training.

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 our process, which can be written as:

$$L_{cyc}(G_{AB}, G_{BA}) = E_{x\sim pdata}(x)\ [\|G_{BA}(G_{AB}(x) - x)\|_1] + \ E_{y\sim pdata}(y)\ \left[\left\|G_{AB}\left(G_{BA}(y)\right) - y\right\|_1\right]$$

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 simplest loss, Identity loss ensures that the network retains the overall color structure of the image. So, adding a concept of regularization that lets us keep the tint of the photo in line with the original shot. Imagine that as a way to guarantee that you can still recreate the original image even after adding several filters.

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. This means translating Amharic text to Amharic using English to Amharic translator since the input is Amharic the network should make no change.

$$L_{identity}(G_{AB}, G_{BA}) = E_{x\sim pdata}(x)\ [\|G_{AB}(x) - x\|_1] + \ E_{y\sim pdata}(y)\ [\|G_{BA}(y) - y\|_1]$$

So, the hole loss would be:

$$\text{Cycle} - \text{GAN}\ = GAN\ loss\ + cycle\ \ loss + Identity\ Loss$$

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

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 also extended by for 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.

$$L_{Fpreserving}(G_{AB}, G_{BA}) = E_{x\sim pdata}(x)\left[\left\|mNET(G_{AB}(x)) - mNET(x)\right\|_1\right] + E_{y\sim pdata}(y)\left[\left\|mNET(G_{BA}(y)) - mNET(y)\right\|_1\right]$$

Where mNET stand for pretrained EfficientNet-B7

By adding the above loss, we inspire the network to minimize the **Object Dislocation** and **Object Disappearance** problem list in section 2.5 to have consistent semantics earlier and afterward the translations. This thesis work uses EfficientNet-B7[37] as a feature extractor that enforces the content information that appears in the original image also should appear on translate. as an example, if a person and a dog appear in image A so does in translated image A2B albeit the style modified. (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 [12], 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:

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

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 [9] and temporal prediction network. The recycle loss[12] across domains and time can be described as:

$$\min_{P_x G_y P_y,} l_{recycle}(G_x, G_y, P_x) = \sum_t \left\| x_{t+1} - G_x(P_y(G_y(x_{t-1}^{t}))) \right\|_1$$

4.4. Temporal warping

To eliminate temporal flickering errors and false discontinuities in the video effects temporal information extracted from temporally consistent frames between recurrent frames as shown in figure 4-1. Notice that the time dynamics of the translated videos should be close to those of the source videos. This thesis study uses flonet2 to measure optical flow [38]. The temporal warping is defined as

$$fx_t = flownet(x, x_{t-1}), fy_t = flownet(y, y_{t-1})$$

$$\tilde{x} = warp(\dot{x}, fx_t), \qquad \tilde{y} = warp(\dot{y}, y_{t-1})$$

4.5. Temporal aware Discriminator

improve visual quality further, a discriminator that takes consecutive images to decide its real or fake. The architecture and the output stay the same with Patch GAN [8] instead the differences are just the input and the number of channels meaning rather that of differentiating between single frames, discriminator design it in a way that it observes two constitutive of synthesized frames and relates them with two constitutive of the real frames which make it ideal since the discriminator takes into account a temporal aspect of the video generation problem.

4.6. Training Pseudocode

Training algorithms for this thesis work have been introduced in this section. As this study compares earlier research, Cycle-GAN. Their training algorithms have been presented in Appendix D and E in order to compare them.

Table 6 Training pseudocode

This thesis work Training pseudocode:
$Take\ a\ sample\ mini-batch: x, x_{t-1}, x_{t-2}, y, y_{t-1}, y_{t-2}$
Train P:
$A, B:\ x\_pred\ =\ P_x([x_{t-1}, x_{t-2}])$
$A, B:\ y\_pred\ =\ P_y([y_{t-1}, y_{t-2}])$
$Compute: p_loss = \frac{1}{2} mea\left((x, x_pred), (y, y_pred)\right)$
$update\ \Theta^{(p_loss)}\ to\ minimize\ prediction\ loss$
Train D:
$Translate\ A, B:\ \tilde{x}\ =\ G_{AB}(x),\ \tilde{y}\ =\ G_{BA}(y)$
$Compute: \boldsymbol{D_A}(\boldsymbol{x}, \boldsymbol{x_{t-1}}, \boldsymbol{x_{t-2}}), \boldsymbol{D_B}(\boldsymbol{y}, \boldsymbol{y_{t-1}}, \boldsymbol{y_{t-2}}), \boldsymbol{D_A}(\boldsymbol{y}, \widetilde{\boldsymbol{y_{t-1}}}, \boldsymbol{y_{t-2}}), \boldsymbol{D_B}(\boldsymbol{x}, \widetilde{\boldsymbol{x_{t-1}}}, \boldsymbol{x_{t-2}})\ then,$
$dloss\ =\ \frac{1}{4} * \sum((D_A(x), D_B(y)), (D_A(\tilde{x}), D_B(\tilde{y})))$
$update\ \Theta^{(dloss)}\ to\ minimize\ classification\ loss.$
Train G:
$Compute: \boldsymbol{fx_t}\ =\ \boldsymbol{flownet}(\boldsymbol{x}, \boldsymbol{x_{t-1}}), \boldsymbol{fy_t}\ =\ \boldsymbol{flownet}(\boldsymbol{y}, \boldsymbol{y_{t-1}})$
$\tilde{\boldsymbol{x}} = \boldsymbol{warp}(\boldsymbol{x}, \boldsymbol{fx_t}),\qquad \tilde{\boldsymbol{y}} = \boldsymbol{flownet}(\boldsymbol{y}, \boldsymbol{y_{t-1}})$
$\tilde{\tilde{x}}\ =\ G_{AB}(\tilde{x}),\ \tilde{\tilde{y}}\ =\ G_{AB}(\tilde{x}), xI\ =\ G_{BA}(x), yI\ =\ G_{AB}(y),$
$\widetilde{\widetilde{x\_pred}}\ =\ G_{BA}(Py([G_{AB}(x_{t-1}), G_{AB}(x_{t-2}))),$
$\widetilde{\widetilde{y\_pred}}\ =\ G_{AB}(Px([G_{BA}(y_{t-1}), G_{BA}(y_{t-2}))))$
$cycle\_loss\ =\ \frac{1}{2}(mae((x\ -\ \tilde{\tilde{x}}), (y\ -\ \tilde{\tilde{y}})))$
$Recycle\_loss\ =\ \frac{1}{2}(mae((x\ -\ \widetilde{\widetilde{x\_pred}}), (y\ -\ \widetilde{\widetilde{y\_pred}})))$
$identity\_loss\ =\ \frac{1}{2}(mae((x\ -\ xI), (y\ -\ yI)))$
$\boldsymbol{feature\_preserving\_loss} = \frac{1}{2}(\boldsymbol{mae}((\boldsymbol{mNet}(\boldsymbol{x})\ -\ \boldsymbol{mNet}(\tilde{\boldsymbol{x}})), (\boldsymbol{mNet}(\boldsymbol{y})\ -\ \boldsymbol{mNet}(\tilde{\boldsymbol{y}}))))$
$\boldsymbol{D_A}(\tilde{\boldsymbol{y}}), \boldsymbol{D_B}(\tilde{\boldsymbol{x}}): \boldsymbol{d\_loss}\ =\ \frac{1}{2}\sum(\boldsymbol{DB}(\boldsymbol{x}, \widetilde{\boldsymbol{x_{t-1}}}, \boldsymbol{x_{t-2}}), \boldsymbol{DA}(\boldsymbol{y}, \widetilde{\boldsymbol{y_{t-1}}}, \boldsymbol{y_{t-2}}))$
$update\ \Theta^{(\text{d\_loss,cycle\_loss,identity\_loss,}p\_loss,Recycle\_loss,feature\_preserving\_loss)}\ min\ loss.$

CHAPTER FIVE

5. Implementation of the Proposed work

5.1. 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.

In this section explain the hardware stack that we used to implement our experiments. In addition to describing the hardware stack.

- » Laptop: The Laptop computer is used for developing a Network Architecture.
  - o Operating system: Windows 10
  - o Processor: Intel ® Core™ i7-2300QM CPU @ 2.00GHz
  - o Graphics: Intel ® Graphics 3000
  - o Primary Memory (RAM): 8.00 GB
  - o System Type: 64-bit Operating System, x64-based Processor
- » Desktop: The desktop computer is used for developing a video for video translation.
  - o Operating system: windows 10
  - o Processor: Intel ® Core™ i5-4580 CPU @ 3.29GHz x 4
  - o Graphics: Intel ® HD Graphics 4600
  - o GPU: GeForce RTX 2070 Super 6 GB RAM
  - o Primary Memory (RAM): 14.00 GB
  - o System Type: 64-bit Operating System, x64-based Processor

Visual Studio Code and Jupiter 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. and TensorFlow-GPU 2.2 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 has 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 specification

During the experiments, Cycle-GAN implemented by TensorFlow Keras has been used in this study, a Cycle-GAN with on 9 residual blocks **used in this study.**

Cycle-GAN comprises of two identical Generator and two identical Discriminator networks since we have two domains. 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 beside the last one.

The discriminator had 70 x 70 Patch GAN. this network composed of 5 convolutional layers denotes a 4 × 4 Convolution-Instance Normalization with LeakyReLU layer and stride 2. After the last layer, apply a convolution to produce a 1-dimensional output. We do not use Batch Normalization for the first layer. The slope of leaky in leakyReLU was 0.2.

Weights in convolutional layers were initialized with a truncated normal distribution initializer with a standard deviation of 0.02, all other layers used a random normal initializer with a standard deviation of 0.02. All biases were initialized to 0. The decay for the moving average for the batch instance normalization was set to 0.9, the epsilon was set to 10−5. Every network used ADAM optimizer with the momentum term $\beta 1$ set to 0.5 and the learning set to 0.0002.

In the LeakyReLU activation, the gradient of the leak was set to 0.2. Lastly, the training process was balanced by making two training steps for the generator for each training step made for the discriminator. Most of the configurations were adopted from the Cycle-GAN paper found and from implementations of Cycle-GAN by its authors in GitHub.

5.5. Implement Cycle-GAN

A Cycle-GAN is made up overall of two architectures: a generator and a discriminator. The generator architecture is used to create two models, Generator AB and Generator BA. The discriminator architecture is used to create an additional two architectural models, Discriminator A, and Discriminator B.

The generator network is an encoder-decoder category network. It takes an image as an input and outputs another image. Based on our base work I define two generator networks

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

The discriminator network is equivalent to the architecture of the discriminator in a Patch GAN network[8]. Basically, it takes an image of the shape of (256, 256, 3) and predicts whether the image is real or fake.

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

The general architecture composed of the above four independent networks. The objective of Generator is to diminish the adversarial loss function against an adversary Discriminator, which constantly tries to maximize it.

```
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])
```

similar to other network types GAN is no different. Learning function has to explicitly defined in order to the network learns to translate image.

```
#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
```

5.6. Temporal Predictor Network Implementation

This thesis work uses Recycle-GAN temporal predictor network Px and Py, which identical to the pix2pix generator network but 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.

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

Similar to every neural network temporal predictor network is similar and has been defined explicitly.

```
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}
```

5.7. Feature Preserving Loss Implementation

As discussed in the previous sections feature preserving loss aims to minimize contain information deference between fake and translated images.

```
import efficientnet.tfkeras as eff #import pretrained EfficientNet-B7
#remove the last four layers
base_model = eff.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)
```

Using pretrained EfficientNetB7 model the new framework has been created. Another method was then established to measure the content of two pictures.

```
def get_content_features(a,b):
      return mNet(a), mNet(b)
```

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.

```
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)
```

5.8. Temporal aware Discriminator Network Implementation

This work also uses additional temporal aware discriminator network.

```
A2B_d_logits = D_B(A2B, A2Bprev, training=True)
```

And cycle loss of temporal aware network would be

```
A2B2A_cycle_loss = cycle_loss_fn(A, A2B2A)#A2B2A is retranslate image.
```

5.9. Temporal information Implementation

Section 4.4 Discuss the extraction mechanism of temporal information used in this paper briefly. Eventually, this section presents a python implementation. As discussed this work use flownet2 as a temporal extractor [21].

```
A2B = G_A2B(A, training=True) #translated image of current frame.
f = flownet2(A, prev)# compute optical flow between successive frames.
prev = A #set A as previous frame for next iteration.
```

```
A2B = insertTemporalInformation(A2B,f) #warp next frame using f
A2Bprev = A2B # for temporal aware discriminator network
```

5.10. Experiment Class

To evaluate the essence of temporal information for video translation testing an initial hypothesis is mandatory, to do so three different classes of experiments are conducted as shown below on the table for each dataset group.

The first class is all about vanilla Cycle-GAN image translation on a given sequence of images Which taking into consideration spatial domain only. The second regarding consider using feature preserving loss. The last one includes temporal cycle loss and temporal Discriminator build up on the second experiment.

Table 7 lists of experimental classes.

class	Experiment	Model used
1	Frame wise video translation (spatial domain)	Cycle-GAN .
2	Content preserving loss.	Cycle-GAN and EfficientNET.
3	Cycle loss and Temporal Discriminator	Cycle-GAN , flownet2, & temporal aware Discriminator.

.

CHAPTER SEVEN

6. Evaluation, Results, and Discussion

6.1. Overview

This chapter presents the evaluation of the video to video translation and the integration of temporal information to improve flickering output by the previous approach. It also discusses the result by comparing it with other related works.

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 question posed in this thesis: *How to preserve temporal consistency for a video to video translation? And 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 an image from the scene as an input a generate an equivalent image in other domain with the consideration preserving temporal information. This work conducts different training experiment to explain the qualitative and quantitative outcomes of the comparison with the baselines on which the study is based on different datasets. This research work uses the inception score (IS) and a human 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 6 below

6.2.1. Flower to Flower

Figure 6-1 demonstrates our method's synthesized frames on the Flower dataset.[6] The videos in this dataset show the blooming of different flowers, which is a relatively slow process, meaning the shifts between adjacent frames are quite small. Generally, our algorithm is capable of preserving the consistency of a sense and content information based on the given input image.

[6] It take around 1.6 sec per image and totally training takes about 57.8 hours.

Table 8 IS score and Human evaluation study Result on flower Dataset

Methods	Flower			
	IS		Human evaluation study[7]	
Real data	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	3.35%	26.7%
CC + CP	1.023±0.009	1.122±0.184	0%	10%
CC + CP + TD	**1.138±0.041**	**1.162±0.025**	96.65%	63.3%

Bold values indicate the best results in the experiments.

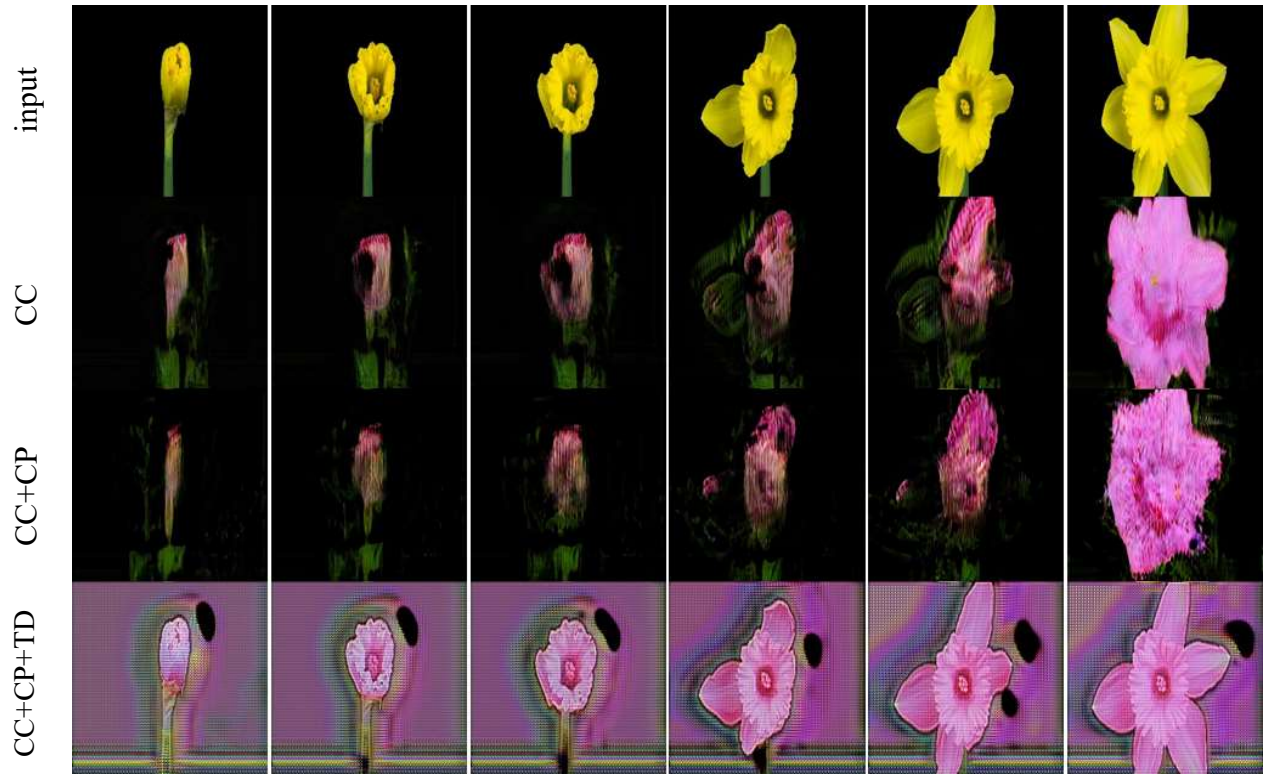


Figure 6-1 flower to flower translation result (from A to B)

The translated flower in each target field retains a continuity for much of the time, with input flower at a different domain. Table 7 shows the inception score of experimental runs of the network. The IS result clearly shows this thesis work advantages over Cycle-GAN(CC) and Cycle-GAN with Feature preserving loss (CC+CP), due to the improvements of video continuity and

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

stability brought by the spatial-temporal constraint. As shown on Figure 6-1 Even if Temporal Continuity can be preserved by this work (CC+CP+TD), it contains a lot of artifacts because it does not really have much weight changes after some epochs due to the vanishing gradient problem (meaning the gradient becomes very low-approximate to zero) so any gradient update does almost nothing in backpropagation.

Figure 6-2 flower to flower translation result (from B to A)

Synthetic images from all evaluated models. The real images shown at the top are inputs 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, the last include Temporal Aware discriminator and Temporal warping.

Observation

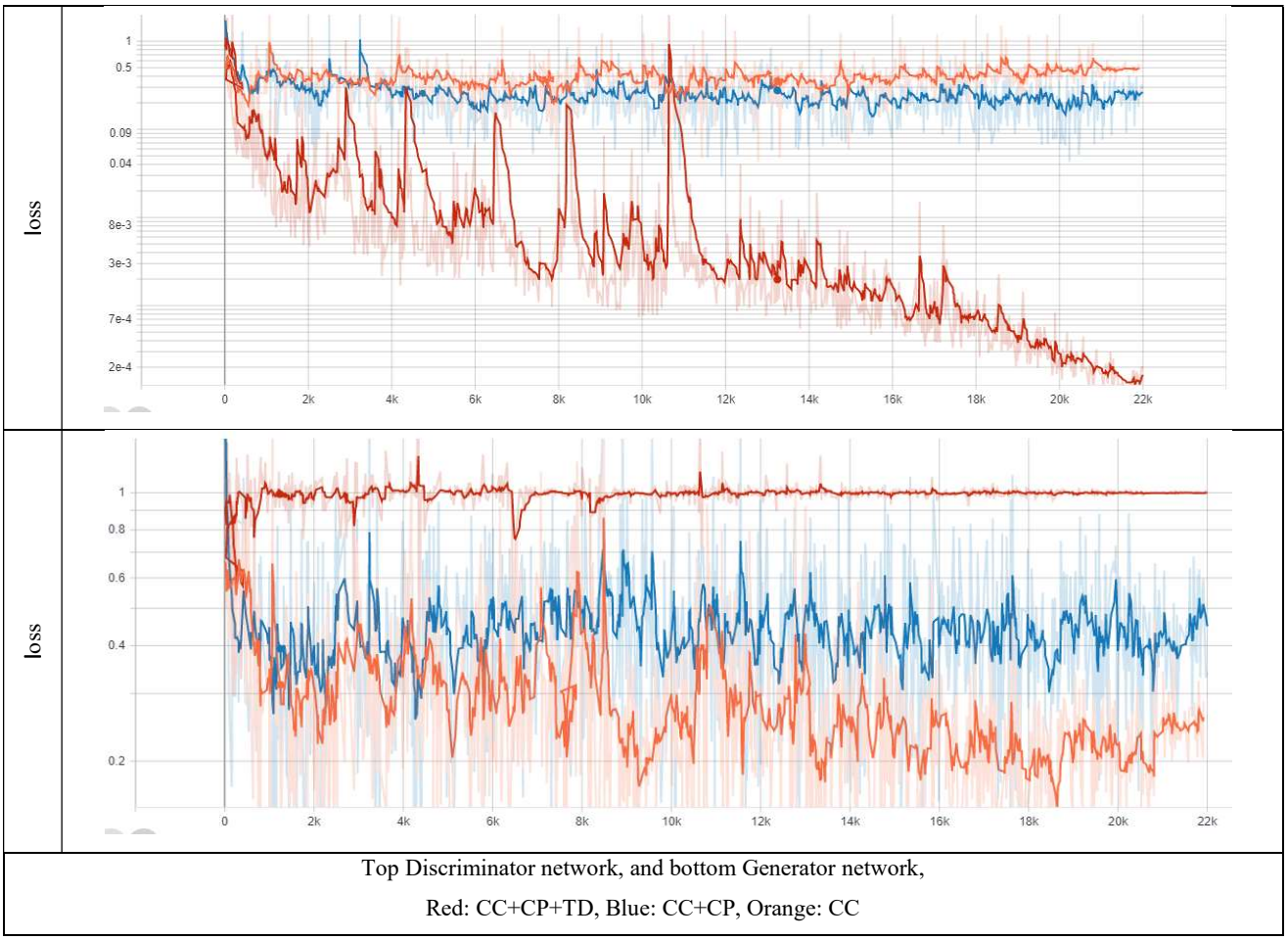
- » Using this proposed work helps the network not only preserve temporal continuity but also help the network to fit in small epochs.
- » The discriminator network becomes too complicated to be tricked by the generator network as shown in figure 6-3, As seen in the picture below, the discriminator loss will be slightly

similar to zero and the generator loss will escalate to one. All the generator generated images are known as false or, in other words, the discriminator network quickly bits the generator, which is not what we want, but we are searching for the Nash equilibrium of the two networks (Generator and Discriminator) to balance each other.

» Even though the Inception score of CC+CP outperform vanilla Cycle-GAN CC Human evaluation study shows CC excel CC+CP.

» Another observation from the output is pixelated and Artifacts because of vanishing gradient and gradient explosion.

Vanishing gradient and gradient explosion problem minimized by applying gradient penalty to the discriminator network. Using the gradient penalty to the network improves the quality of the generated output as shown in figure 6-4 below. (in a result Human evaluation study and Inception score experiment the gradient penalty not included for a fair comparison)



Top Discriminator network, and bottom Generator network,
Red: CC+CP+TD, Blue: CC+CP, Orange: CC

Figure 6-3 weight Vanishing problem on CC+CP+TD

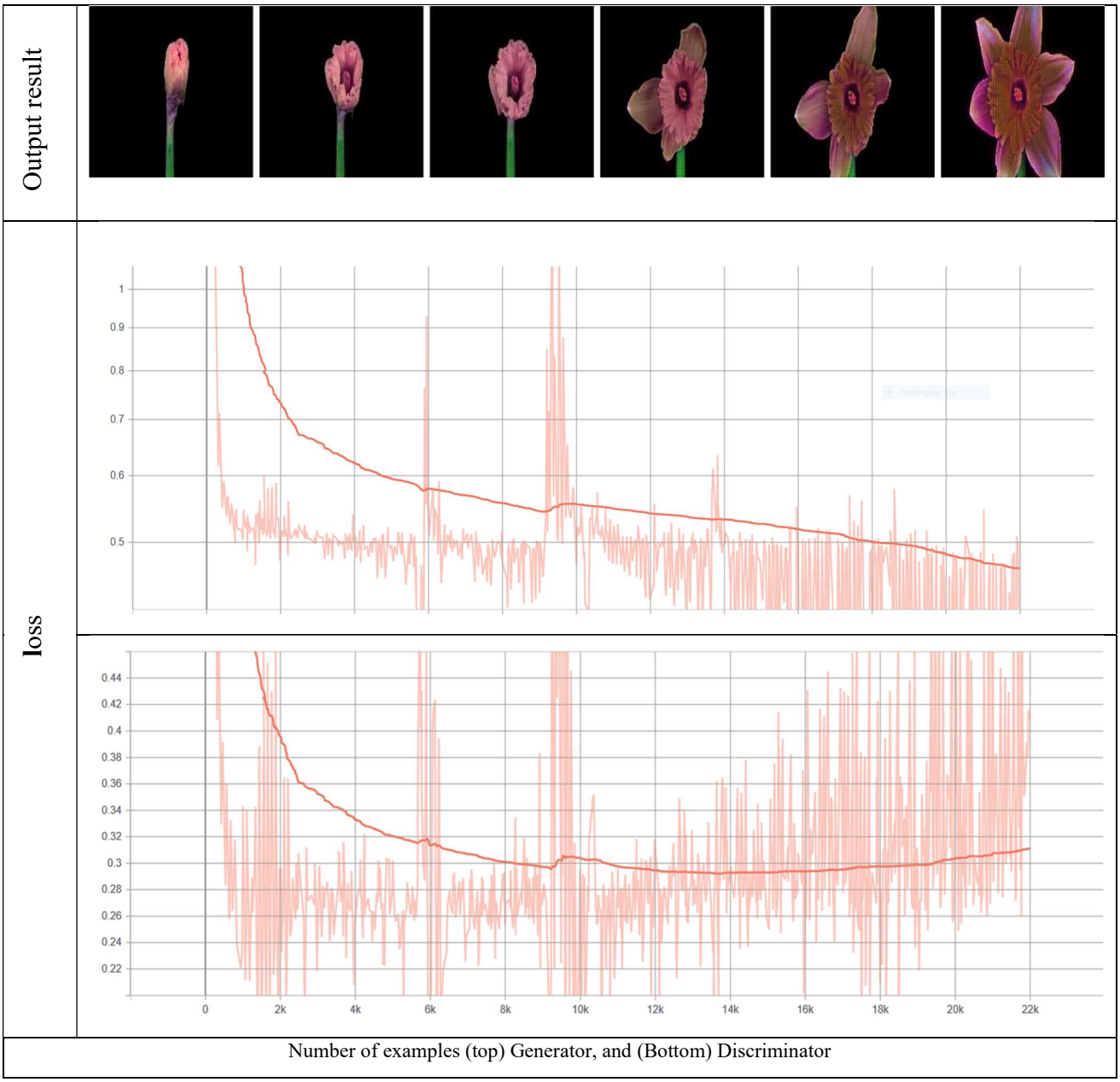


Figure 6-4 CC+CP+TD with gradient penalty

6.2.1. Sunset to Day

Similar to flower translation this experiment uses the same training setup, except uses a subset of viper dataset target to translate Day time image to Sunset and vice versa. It takes around 1.6 sec per image and totally about 136.8 hours to train. The task of Sunset to Day is shown to explain the impact 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 6-6,

Figure 6-5 Day to Sunset translation output Result

Eventually, this work positively improves visual quality as confirmed in IS and Human evaluation study results (Table 8). 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.

Figure 6-6 Sunset to Day translation Output Result

Table 9 IS score and Human evaluation study Result on Viper Dataset

Methods	Day to Sunset			
	IS		Human evaluation study[8]	
Real data	3.56s±0.21	3.81±0.44		
	Domain A	Domain B	Domain A	Domain B
CC	2.50±0.17	2.71±0.19	3.34%	0%
CC + CP	3.09±0.07	**3.64±0.26**	**48.3%**	38%
CC + TD	**3.23±0.13**	3.61±0.11	**48.3%**	**62%**

Observation:

» CC+CP performs a much greater in Sunset to Day datasets Compared to the flower translation as shown in table 8 above, I suppose it's because the Efficientnet-B7 feature preserving network trains on image-net. I didn't think there were substantial flower blooming(in fact it only contains 1197 images of flower and 0 instance of bloomer) instances in training so the network might not be able to extract adequate features in flower video so the network performed badly due to this cause.

» CC+CP+TD has been impacted by the vanishing of the gradient problem in the tiny dataset as seen in the flower dataset, maybe the viper dataset is very big 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. 12%. over Cycle-GAN with feature preserving loss and 53.48% on Cycle-GAN. The quantitative measure inception score shows CC+CP slight edge over our model in Domain B. Table 10 shows that CC does not retain details information, but our model generates a decent result.

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

Table 10 comparison between Cycle-GAN with this thesis work

6.2.2. Face to Face

In this experiment, we evaluate Barack Obama ↔ Donald Trump using the Recycle-GAN configuration. The IS result clearly shows this thesis work (RC+TD) advantages over ReCycle-GAN(RC). ReCycle-GAN with Temporal warping and temporal Discriminator (RC+TD) (**R.B**

Content preserving network can't be implemented in this work since this work focus on video Retargeting).

Table 11 Obama to Trump Translation Result

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

Both approaches are capable of accessing the stylistic facial gestures of Donald Trump. But mouth motion slightly Differ as shown in the above table 11. For example in Trump to Obama translation

[9]our model fetches trump mouth movement more reasonably than the comparative model Recycle GAN.

Again, in comparison with that of the ReCycle-GAN, our network increases the IS scoring favorably and this thesis work outperforms human study by 46% of basic work.

Table 12 Obama to Trump Inception Score and Human evaluation Study.

Methods	Obama to Trump			
	IS		Human evaluation study[10]	
Real data	1.283 ± 0.102	1.069 ± 0.274		
	Obama	Trump	Obama	Trump
RC	1.035 ± 0.120	1.048 ± 0.010	38%	16%
RC + TD	**1.041 ± 0.013**	**1.068 ± 0.011**	**62%**	84%

[9] 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

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

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 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 has 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 start on the ReCycle-GAN claim Among the investigation, the goal was to generate as visually realistic video as possible. To do this, this thesis adds Feature preserving loss, temporal warping, and Temporal aware discriminator to the baseline works. Indeed, these changes make our model very aware of the perpetual spatial-temporal information changes in the video.

In fact, the Temporal aware discriminator improves the classical Cycle-GAN discriminator which totally gives focus only to the Generated image look real or fake based on Spatial information only. But our 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. 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. our model Combine the above two losses to preserve temporal information.

Compared with baseline works Cycle-GAN[9] and ReCycle-GAN[12], qualitative and quantitative experimental findings indicate the achievement of this thesis work. Its thesis excels in the human assessment analysis by xxx percent and xxx percent in the IS score of Cycle-GAN. this Research work concludes that Adding constraints to video to video translation does improve temporal coherency.

7.2. Limitation and Future work

The proposed method doesn't come without limitations. We have observed that the model is strongly dependent on the EfficientNET-B7 outputs and since the feature preserving loss is not designed to be consistent across frames and generated output depends essentially on the performance of the feature extraction network on a specific dataset as discussed on 6.2.1. This naturally leads to inconsistency in the results produced. One approach to resolve this issue is a retune feature extraction network on our dataset or train in flight together.

Finally, the work on this thesis might take us in a very different direction: letting it learn how to construct synthetic intermediate frames between successive frames, which 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 consideration more than that of two frames as used in this thesis work.

References

[1] Jake VanderPlas, *Python Data Science Handbook*. O'Reilly Media, Inc.

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

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

[4] 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.

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

[6] "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].

[7] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, "Temporal cycle-consistency learning," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, no. 12, pp. 1801–1810, Apr. 2019.

[8] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-Janua, pp. 5967–5976.

[9] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, vol. 2017-Octob, pp. 2242–2251.

[10] 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.

[11] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," Nov. 2014.

[12] A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh, “Recycle-GAN: Unsupervised Video Retargeting,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11209 LNCS, pp. 122–138.

[13] 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.

[14] 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.

[15] “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].

[16] Y. Choi, M. Choi, M. Kim, J. W. Ha, S. Kim, and J. Choo, “StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 8789–8797, Nov. 2018.

[17] 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.

[18] D. Chen, J. Liao, L. Yuan, N. Yu, and G. Hua, “Coherent Online Video Style Transfer,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, vol. 2017-Octob, pp. 1114–1123.

[19] T. R. Shaham, T. Dekel, and T. Michaeli, “SinGAN: Learning a Generative Model from a Single Natural Image,” May 2019.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.

[25] 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.

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

[27] 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.

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

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

[30] 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.

[31] 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.

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

[33] “TensorFlow.” [Online]. Available: https://www.tensorflow.org/. [Accessed: 02-Jun-2020].

[34] J. Hale, “Deep Learning Framework Power Scores,” 2018. .

[35] “OpenCV.” [Online]. Available: https://opencv.org/. [Accessed: 02-Jun-2020].

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

[37] 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.

[38] T.-W. Hui, X. Tang, and C. C. Loy, “A Lightweight Optical Flow CNN - Revisiting Data Fidelity and Regularization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, Feb. 2020.

Appendix

Appendix A: Loss function

The loss function is used to calculate the error of an event. Which is used to determine the error between the output of our algorithms and the given target value An example of an event is a neural network that produces an image. The loss function could then be a resemblance measurement between the produced image and a corresponding ground truth image. There currently exists a variety of loss functions, the ones most relevant for this thesis are described below.

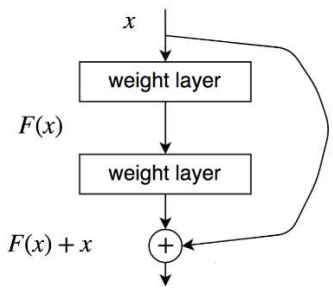
Mean Squared Error (MSE): Used to compare the differences in two images. The difference of the corresponding Pixels of each image is calculated, squared and the mean overall pixels are calculated.

Mean Absolute Error (MAE): is the sum of absolute differences between our target and predicted variables. A difference between the MSE loss and the MAE loss is that outliers in the MSE have a larger impact on the loss since the error is squared.

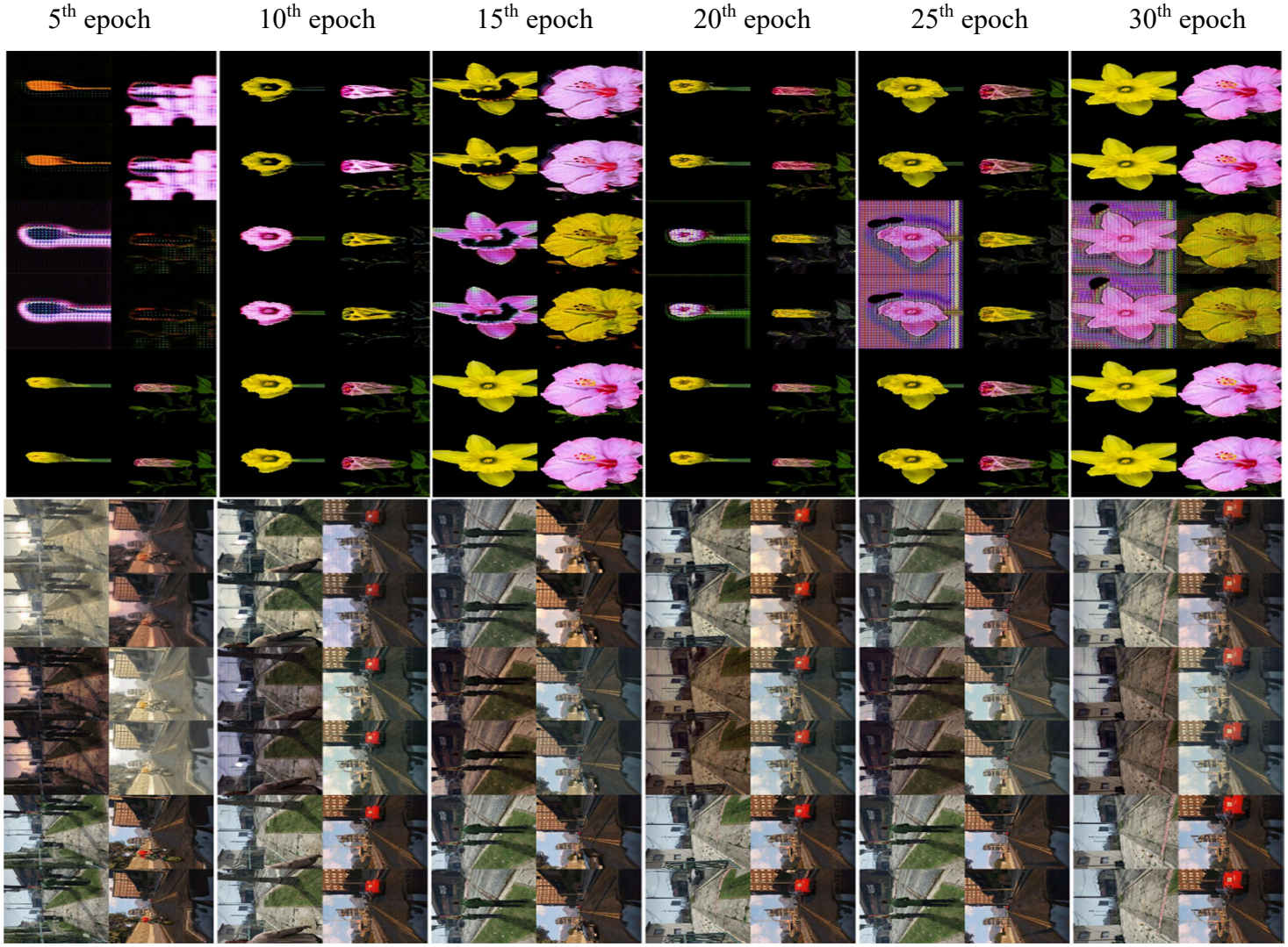
$$MSE = \frac{1}{n}\sum_{i=o}^{n}(y-\hat{y})^2, MAE = \frac{1}{n}\sum_{i=o}^{n}\|y-\hat{y}\|$$

Appendix *B:* Residual Blocks:

Increasing the number of layers (deeper net) in a network provides additional nonlinearities that can benefit the classification task since more complex solutions can be learned, but training becomes more complex. When adding more layers and giving the network more parameters, the performance of the network is not necessarily improved. then the solution would be residual block. in the residual block, the output of the lower layer of the network feeds into the input of the layer.



Appendix C: Result on Different epochs



The result on different epoch left side is on the Viper dataset, and the right side is the flower dataset, row label show the corresponding epoch

25th epoch	30th epoch	35th epoch	40th epoch	45th epoch	50th epoch
					
1st epoch	5th epoch	10th epoch	15th epoch	20th epoch	

Result on different epoch on Obama Trump dataset, row label shows the corresponding epoch

Appendix D: Cycle-GAN Training pseudocode:

Cycle-GAN Training pseudocode:

$take\ a\ sample\ mini-batch: x, y$

Train D:

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

$Compute: D_A(x), D_B(y), D_A(\tilde{y}), D_B(\tilde{x})\ then,$

$$dloss = \frac{1}{4} * \sum((D_A(x), D_B(y)), (D_A(\tilde{x}), D_B(\tilde{y})))$$

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

Train G:

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

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

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

$D_A(\tilde{y}), D_B(\tilde{x}) : d\_loss = \frac{1}{2}\sum(DA(\tilde{y}), DB(\tilde{x}))$

$update\ \Theta^{(\text{d\_loss,cycle\_loss,identity\_loss})}\ to\ maximizeclassification\ loss.$

Appendix E: Cycle-GAN with Feature Preserving Training pseudocode:

Cycle-GAN with Feature Preserving Training pseudocode:

$Take\ a\ sample\ mini-batch: x, y$

Train D:

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

$Compute: D_A(x), D_B(y), D_A(\tilde{y}), D_B(\tilde{x})\ then,$

$$dloss = \frac{1}{4} * \sum((D_A(x), D_B(y)), (D_A(\tilde{x}), D_B(\tilde{y})))$$

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

Train G:

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

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

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

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

$D_A(\tilde{y}), D_B(\tilde{x}) : d\_loss = \frac{1}{2}\sum(DA(\tilde{y}), DB(\tilde{x}))$

$update\ \Theta^{(\text{d\_loss,cycle\_loss,identity\_loss,}feature\_preserving\_loss)}\ to\ minimize\ classification\ loss.$