“***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

September 2020

“***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

September 2020

Declaration

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

|  |  |
| --- | --- |
| Name | Signature |
| Kirubel Abebe Senbeto | \_ |

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

|  |  |
| --- | --- |
| Advisor. Name | Signature |
| Yun Koo Chung (Ph.D.) | \_ |

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

[ACKNOWLEDGMENT i](#_Toc52051158)

[LIST OF TABLES vii](#_Toc52051159)

[LIST OF FIGURES viii](#_Toc52051160)

[LIST OF EQUATIONS ix](#_Toc52051161)

[List of Abbreviations x](#_Toc52051162)

[Abstract xiii](#_Toc52051163)

[1. Introduction 1](#_Toc52051164)

[1.1. Background 1](#_Toc52051165)

[1.1. Backgrounds of the Study 4](#_Toc52051166)

[1.1.1. Generative Adversarial Networks 4](#_Toc52051167)

[1.2. Motivation 6](#_Toc52051168)

[1.2.1. Statement of the Problem 7](#_Toc52051169)

[1.3. The Objective of the Thesis 7](#_Toc52051170)

[1.3.1. General Objective 7](#_Toc52051171)

[1.3.2. Specific Objectives 7](#_Toc52051172)

[1.4. Research Methodology 8](#_Toc52051173)

[1.4.1. Data Collection 8](#_Toc52051174)

[1.4.2. Literature Review 8](#_Toc52051175)

[1.4.3. Evaluation 8](#_Toc52051176)

[1.4.4. Implementation Tools 9](#_Toc52051177)

[1.5. Scope and Limitation of the Study 9](#_Toc52051178)

[1.5.1. Scope of the Study 9](#_Toc52051179)

[1.5.2. Limitations of the Study 9](#_Toc52051180)

[1.6. Organization of the Thesis 9](#_Toc52051181)

[2. Literature review and Related work 11](#_Toc52051182)

[2.1. Introduction 11](#_Toc52051183)

[2.2. Inside GAN 12](#_Toc52051184)

[2.2.1. GAN Training 13](#_Toc52051185)

[2.2.2. Conditional GAN 14](#_Toc52051186)

[2.3. Variational Autoencoders 15](#_Toc52051187)

[2.4. Image-to-Image Translation 15](#_Toc52051188)

[2.5. Video to video translation 18](#_Toc52051189)

[2.6. Problems in Translation Networks 18](#_Toc52051190)

[2.7. Temporal Information 19](#_Toc52051191)

[2.7.1. Optical flow 20](#_Toc52051192)

[2.7.2. Pose Estimation 21](#_Toc52051193)

[2.7.3. 3D convolutional tensor 22](#_Toc52051194)

[2.7.4. Recurrent temporal 22](#_Toc52051195)

[2.8. Related Works 22](#_Toc52051196)

[2.9. Related work summary 24](#_Toc52051197)

[3. Materials and Methods 27](#_Toc52051198)

[3.1. Overview 27](#_Toc52051199)

[3.2. Dataset 27](#_Toc52051200)

[3.3. Development tools 29](#_Toc52051201)

[3.4. Design tools 29](#_Toc52051202)

[3.5. Prototype development framework 30](#_Toc52051203)

[3.5.1. TensorFlow 30](#_Toc52051204)

[3.5.2. OpenCV 31](#_Toc52051205)

[3.5.3. MATLAB Deep Network Designer 31](#_Toc52051206)

[3.6. Baseline Works 31](#_Toc52051207)

[3.7. Feature Extraction Algorithm 32](#_Toc52051208)

[3.8. Evaluation Methods 33](#_Toc52051209)

[3.8.1. Human Evaluation Study 33](#_Toc52051210)

[3.8.2. Inception Score 33](#_Toc52051211)

[4. Proposed loss function. 34](#_Toc52051212)

[4.1. Overview 34](#_Toc52051213)

[4.2. Model Architecture. 34](#_Toc52051214)

[4.3. Model Learning Functions 35](#_Toc52051215)

[4.3.1. Proposed Network Learning Function. 35](#_Toc52051216)

[4.4. Temporal warping 38](#_Toc52051217)

[4.5. Temporal aware Discriminator 38](#_Toc52051218)

[4.6. Training Pseudocode 39](#_Toc52051219)

[5. Implementation of the Proposed work 41](#_Toc52051220)

[5.1. Overview 41](#_Toc52051221)

[5.2. Working Environment. 41](#_Toc52051222)

[5.3. Environmental Setup 41](#_Toc52051223)

[5.4. Implement Cycle-GAN 42](#_Toc52051224)

[5.5. Temporal Predictor Network Implementation 43](#_Toc52051225)

[5.6. Feature Preserving Loss Implementation 44](#_Toc52051226)

[5.7. Temporal aware Discriminator Network Implementation 45](#_Toc52051227)

[5.8. Temporal information Implementation 46](#_Toc52051228)

[5.9. Experiment Class 46](#_Toc52051229)

[6. Evaluation, Results, and Discussion 48](#_Toc52051230)

[6.1. Overview 48](#_Toc52051231)

[6.2. Video to video Translation 48](#_Toc52051232)

[6.2.1. Flower to Flower 48](#_Toc52051233)

[6.2.1. Sunset to Day 52](#_Toc52051234)

[6.2.2. Face to Face 55](#_Toc52051235)

[7. Conclusion and Future work 58](#_Toc52051236)

[7.1. Conclusion 58](#_Toc52051237)

[7.2. Limitation and Future work 59](#_Toc52051238)

[References 60](#_Toc52051239)

[Appendix 65](#_Toc52051240)

[Appendix A: Loss function 65](#_Toc52051241)

[Appendix *B:* Residual Blocks: 65](#_Toc52051242)

[Appendix C: Result on Different epochs 66](#_Toc52051243)

[Appendix D: Cycle-GAN Training pseudocode: 69](#_Toc52051244)

[Appendix E: Cycle-GAN with Feature Preserving Training pseudocode: 70](#_Toc52051245)

[Appendix F: Inception Score: 71](#_Toc52051246)

# LIST OF TABLES

[Table 1 generator goal vs discriminator goal 13](#_Toc52050623)

[Table 2 Cycle-GAN generator and discriminator operation. 17](#_Toc52050624)

[Table 3 previous works summary on the video to video translation. 24](#_Toc52050625)

[Table 4 Training Dataset Sample (from Obama - Trump and Flower Datasets) 28](#_Toc52050626)

[Table 5 Viper Dataset Sample Examples 29](#_Toc52050627)

[Table 6 Training pseudocode 40](#_Toc52050628)

[Table 7 lists of experimental classes. 47](#_Toc52050629)

[Table 8 IS score and Human evaluation study Result on flower Dataset 49](#_Toc52050630)

[Table 9 IS score and Human evaluation study Result on Viper Dataset 54](#_Toc52050631)

[Table 10 Obama to Trump Inception Score and Human evaluation Study. 57](#_Toc52050632)

# LIST OF FIGURES

[Figure 1‑1(a) input image. (b) style image. (c) output image 3](#_Toc52051085)

[Figure 1‑2 Cycle Gan vs Recycle GAN 5](file:///C:\Users\thisiskiru\Documents\GitHub\desktop-tutorial\Temporal%20cycle%20consistency.docx#_Toc52051086)

[Figure 2‑1 GAN framework structure GAN framework consists of two networks: Discriminator (𝐷) and Generator (𝐺) 11](file:///C:\Users\thisiskiru\Documents\GitHub\desktop-tutorial\Temporal%20cycle%20consistency.docx#_Toc52051087)

[Figure 2‑2 cGAN Architecture 14](file:///C:\Users\thisiskiru\Documents\GitHub\desktop-tutorial\Temporal%20cycle%20consistency.docx#_Toc52051088)

[Figure 2‑3 Amharic to English language translation using google translator (Example). 16](#_Toc52051089)

[Figure 2‑4 (A) pair shoe dataset sample from Pix2pix, (B) Sunny to Rainy translation from input and output image 17](file:///C:\Users\thisiskiru\Documents\GitHub\desktop-tutorial\Temporal%20cycle%20consistency.docx#_Toc52051090)

[Figure 2‑5 Detection of the optical flow in 3 consecutive images. 20](file:///C:\Users\thisiskiru\Documents\GitHub\desktop-tutorial\Temporal%20cycle%20consistency.docx#_Toc52051091)

[Figure 2‑6 pose extraction 21](file:///C:\Users\thisiskiru\Documents\GitHub\desktop-tutorial\Temporal%20cycle%20consistency.docx#_Toc52051092)

[Figure 3‑1 Deep Learning Framework comparison. 30](#_Toc52051093)

[Figure 3‑2 Benchmark Analysis on EfficientNet-B7 32](#_Toc52051094)

[Figure 4‑1 Generator Network Architecture 34](file:///C:\Users\thisiskiru\Documents\GitHub\desktop-tutorial\Temporal%20cycle%20consistency.docx#_Toc52051095)

[Figure 4‑2 Temporal Discriminator Network 39](#_Toc52051096)

[Figure 6‑1 flower to flower translation result 50](#_Toc52051097)

[Figure 6‑2 weight Vanishing problem on CC+CP+TD 51](#_Toc52051098)

[Figure 6‑3 CC+CP+TD with gradient penalty 52](#_Toc52051099)

[Figure 6‑4 Sunset to Day translation Output Result 53](#_Toc52051100)

[Figure 6‑5 Comparison between Cycle-GAN with this thesis work 55](#_Toc52051101)

[Figure 6‑6 Obama to Trump Translation Result 56](#_Toc52051102)

[Figure 6‑7 RC Trump Generated image sequences 57](#_Toc52051103)

# LIST OF EQUATIONS

[eq.( 2.1) 12](#_Toc52013984)

[eq.( 3.1) 33](#_Toc52013985)

[eq.( 4.1) 36](#_Toc52013986)

[eq.( 4.2) 36](#_Toc52013987)

[eq.( 4.3) 37](#_Toc52013988)

[eq.( 4.4 ) 38](#_Toc52013989)

[eq.( 4.5 ) 38](#_Toc52013990)

[eq.( 4.6) 39](#_Toc52013991)

# List of Abbreviations

|  |  |
| --- | --- |
| 2D | 2-Dimensional |
| 3D | 2-Dimensional |
| AED | Annotation Edit Distance |
| AI | Artificial intelligence |
| ANN | Artificial neural network |
| API | Application programming interface |
| CC | Cycle Constraint |
| CC+CP | Cycle Constraint plus feature preserving |
| CC+CP+TD | Cycle Constraint plus feature preserving plus Temporal Aware Discriminator |
| CGAN | Conditional generative adversarial network |
| CGI | Computer-generated imagery |
| CNN | Convolutional neural network |
| Conv-nets | Convolutional neural |
| CPU | Central processing unit |
| CYCLEGAN | Unpaired Image-to-Image Translation using Cycle-Consistent |
| DL | Deep Learning |
| DX | Discriminator X |
| DY | Discriminator Y |
| EfficientNET-B7 | EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks |
| FCN | Fully Convolutional Networks |
| FID | Fréchet Inception distance |
| flownet2 | FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks |
| GAN | Generative adversarial networks |
| GPU | graphics processing unit |
| GX | Generator X |
| GY | Generator X |
| HFR | High Frame Rate |
| IDE | Integrated development environment |
| IoU | Intersect over Union |
| IS | inception score |
| L1 | Manhattan Distance or L1 Norm |
| L2 | Euclidean distance |
| LSTM | Long short-term memory |
| mIoU | mean Intersect over Union |
| ML | Machine learning |
| MoCycle-GAN | Mocycle-GAN: Unpaired Video-to-Video Translation |
| MPI | Max Planck Institute |
| P(z) | Noise Data Distribution |
| PIX2PIX | Image-to-Image Translation with Conditional Adversarial Nets |
| RC | ReCycle-GAN |
| RC+TD | ReCycle-GAN plus Temporal Discriminator |
| ReCycle-GAN | Recycle-GAN: Unsupervised Video Retargeting |
| RGB | Red Green Blue |
| RNN | recurrent neural network |
| sec | Second |
| TPU | Tensor Processing Unit |
| VAE | Variational Autoencoder |
| 𝑋 → 𝑌 | x to y |
| 𝑌 → 𝑋 | y to x |

# Abstract

Generative Adversarial Networks (GANs) is a deep learning method that is developed for synthesizing data. Area of applications for which it can be used are image-to-image translations, Video to video translation, and video retargeting. However, to train the model there is a need for large amounts of complex paired data which is hard to find. To collect dataset, especially when we need a paired dataset is time-consuming and expensive. One way to overcame this problem is to collect datasets in one domain and translate it to another domain using image translation techniques to make it a paired dataset. Various research has leveraged enormous in image translation by the use of GANs on an unpaired dataset. As far as video translation is concerned, current GAN-based approaches do not entirely leverage space-time knowledge in videos.

This research examines the idea of using GANs for the utilization of spatial-temporal information in a video by extending the unpaired video-to-video translations model (ReCycle-GAN) to enhance spatial-temporal video translation. In particular, previous methods suffer from Object disappearance, Object dislocation, and flickering Artifacts. To Mitigate these issues, this work proposes to adds feature preserving loss and temporal aware discriminator to the network Cycle GAN and ReCycle GAN to generate more temporal consistent videos. Extensive qualitative and quantitative assessments demonstrate the notable success of the proposed system against existing methods. Experiments have shown that this research learns more Spatio-temporal information from the video. This paper concludes that Adding feature preserving constraints and temporal aware discriminator to the Cycle GAN and ReCycle-GAN models does improve temporal coherency of output video.

***Keywords:*** Cycle GAN, ReCycle GAN, Spatial-temporal information, Unsupervised Video to Video translation

CHAPTER ONE

# Introduction

## Background

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 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 , the output variable 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 is known; this is referred to as 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 .

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. Alternatively, supervised learning techniques can be used to predict labels to every unlabeled . 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]. However, 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 DL's potential 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 a 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.*



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

Source: Adapted from [6]

Last year (2018), all Artificial intelligence news [7] headlines were screaming about a paint drawn 100% by AI sold $432,500 (fascinating, isn't it?). (Because of style transfer data scientist does not need to buy a hundred-thousand-dollar painting for decorating his living room while he can have one when he is home siting in Infront of his laptop, marvelous)

Perhaps the first successful neural style transfer paper was published in 2015 by [8]**.** After this work, many researchers came with a more realistic synthetic image. pix2pix [9] 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. [10] 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.

## Backgrounds of the Study

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.

### 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. 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.

Figure 1‑2 Cycle Gan vs Recycle GAN

Source: Adapted from[11]

The second Network is entitled the **Discriminator**. This model aims to recognize if an input data is ‘real’ — goes to the original dataset — or if it is ‘fake’ — generated by a falsifier Generator Network. In this scenario, a Discriminator is corresponding to the law enforcement agent (or an art expert), which tries to spot artworks as truthful or fraud. Successful training of a GAN requires reaching an equilibrium state between two opposing objectives, unlike CNN or Long Short-Term Memory (LSTM) where the training objective is to minimize or maximize the value of a single loss function.

#### Conditional GAN

The conditional GAN [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.

#### 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 [12] they lack generalization and does not work well if there is occlusion. The second approach use paired image to image translation such as pix2pix -in an image it takes a pixel, then converts to another pixel. [9] use conditional GAN [13], learn a mapping between paired input to the output image. The third category is unsupervised and unpaired data domain transfer like Cycle-GAN [11], which enforces cycle consistency for the unpaired image.

The recent state of artwork work ReCycle-GAN by Bansal [11] motivated by [10] proposes video retargeting via spatiotemporal constraint though directly synthesizing future frames via temporal predictor to preserve temporal continuity. Bansal et al., clams video to video translation 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 **temporal cycle consistency constraints also proposes to introduce Spatiotemporal video to video** translation for better realistic results.

## Motivation

Recent deep learning achievement has been done because of the enormous 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 significant impact on computer vision and deep learning society.

Even though this work focused on the unpair dataset, there is one significant 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 [11], [14] explicitly for a video to video translation.

### 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 and respectively, cycle constraint enables an image to image translation in mutually frontward and backward mapping. There are two mapping functions mapping from domain and correspondingly form target domain to source and vice versa. where is input video frame at time and is a synthetic frame in domain same is true in domain. Cycle consistency constraint so then as well as so then .

Besides the preservation of cycle consistency in each frame this work-study mapping temporal consistency between consecutive frames in both domains. Meaning let optical flow between and is and optical flow between is , then, temporal cycle consistency need to enforce motion consistency via minimizing the difference between . Recycle-GAN [11] 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.

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

## The Objective of the Thesis

### 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 [11] ReCycle-GAN.

### Specific Objectives

The following specific objectives is 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.
* Preprocess the dataset in order to enhance its quality.
* Extract temporal information from the video.
* Add learning constraints to the Cycle-GAN and recycle-GAN networks.
* Design a deep learning video translation model using Keras and tensorflow framework.
* Train the model using a proper dataset.
* Test the trained model with a test set.
* Assess the performance of the model.

## Research Methodology

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

### 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.

### 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.

### 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.

### 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.

## Scope and Limitation of the Study

### Scope of the Study

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 constraints to the ReCycle-GAN network.
* Blend spatial information to temporal information to improve the consistency of video to video translation.

### Limitations of the Study

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.

## 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

# Literature review and Related work

## 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]](#footnote-1). The generative model learns to mimic data while the discriminative model learns to determine whether a sample is from the model distribution or the data distribution .

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

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.

## 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](#_Inside_GAN), which are represented by diﬀerentiable functions concerning each network’s input and parameters. The discriminator is deﬁned by a function where (observed variable) is the input which is a real dataset. gives the likelihood that came from (real distribution) rather than (fake distribution). It is a binary classiﬁer with two classes, when is real the probability is 1 and when 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 where (latent variable) is the input and generates an image as its output . 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]:

|  |  |
| --- | --- |
|  | eq.( 2.1) |

Equation 1 Adversarial loss function

The generator can be seen as a kind of reverse CNN. It takes an -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.



### 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 seeks to get as real as possible to 1 (label for the positive class). Meaning 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 that are alike from the real data in the training dataset. Mathematically, the Generator strives to produce fake examples such that is as close to 1 as possible.

Table 1 generator goal vs discriminator goal

|  |  |
| --- | --- |
| ***Generator*** | such that is as close to 1 as possible. |
| ***Discriminator*** | 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)

1. Train the Discriminator:
   1. Take a random mini-batch of real examples: .
   2. Take a mini-batch of random noise vectors z and generate a mini-batch of fake examples:
   3. Compute the classification losses for and , and backpropagate the total error to update to minimize the classification loss.
2. Train the Generator:
3. Take a mini-batch of random noise vectors z and generate a mini-batch of fake examples:
4. Compute the classification loss for and backpropagate the loss to update 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

### 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 the composite association between the latent space input to the generator in order to mimic the real dataset and the generated images[5], [15].

Figure 2‑2 cGAN Architecture

Mirza et al propose The conditional generative adversarial network, or cGAN [13] 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]](#footnote-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 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 [9], [11]. The next section will discourse about Image to image translation.

## 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.

## Image-to-Image Translation

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****”* [16]I was wondering too winter to summer Absolutely fascinating.

Recent advancements in GANs [5] empowers style transfer models to create realistically looking [8]–[10], [17] adapted image (2-4 B show image to image transfer from sunny to rainy). 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 primary difficulty in the image to image translation is they need paired data set for training, but in reality, doing so is very expensive and not scalable, but some work achieves good results. pix2pix[9] 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 it 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 [10] which was invented to learn bidirectional mapping in the absence of paired training data via Cycle consistency loss. *Cycle Consistency loss* utilizes to learn transformation between two domains in a frontward and backward fashion. Cycle consistency constraint is not a new idea; in fact, very old news in natural language processing. The following example gives a simple illustration. Assume using language translation from Amharic (ኣማርኛ) to English in both directions. When the user input “ስም አበበ ይባላል፡፡” the model should generate “My name is Abebe” perhaps if the user translates “My name is Abebe” to Amharic back again it should generate the original text “ስም አበበ ይባላል፡፡”. Meaning the difference between the original text and regenerated text should be minimum. 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 us see it in bit detail using a table 2.

Table 2 Cycle-GAN generator and discriminator operation.

|  |  |  |
| --- | --- | --- |
|  | Translate from to |  |
|  | Translate from to |  |
|  | Classify real and fake | 1 for , 0 for |
|  | Classify real and fake | 1 for , 0 for |

The loss function of the network could be formulated as:

Meaning translate a given image are and reconstructed image the difference should be the minimum . Input image 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

Source: Adapted from [9]

## **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 [18], [19] effect in the output video. To overcome the flickering effect, Chen et al. [19] consider temporal information along with spatial information. Specifically, they exploit previous frame optical flow to warp the current frame towards imposing temporal constraints. Let see what temporal information and different mechanism to extract is. But before that it worth brief discussion about the problem in current approaches.

## Problems in Translation Networks

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

Let say we have two generators and to translate from one domain to another domain and two discriminators and , where trained to translate from to and from to . and discriminators and to classify between real and fake in both domains. Video and are sample videos from respective domain and . where are the ith frame of video . Each frame may contain various objects. Objects in a frame can be seen as a group of connected pixels.

* **Object disappearance**: is a problem object in a given video frame in domain shall also appear in translated appear in another domain image. meaning if a car appears in is should also appear in Mathematically,
* ***Object dislocation***[[3]](#footnote-3): happen when an object in frame from a domain changes its position when translated in domain . Object dislocation also can be seen as an abrupt object movement. Mathematically,
* ***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.

## Temporal Information

Temporal refers to time-domain, wherein our case, it can be seen as a relation between the 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 forward approaches basically fail because they cannot consider both domains. Temporal information for video can describe a phenomenon in a particular pixel location with position change in time.

For a video to video translation, we have various options to represent motion information. The next section would discuss those topics. For illustration, The extraction of time knowledge can be split into two separate groups. One explicit temporal information extraction: this kind of network operates in such a way that the model extracts temporal information directly, such as optical flow and pose estimation. Then the model imposes temporal information on the generated frame. The other tacit does not explicitly collect temporal knowledge but aims to learn temporal dimension through specially modelled learning layers of the model. Examples could be 3D Conv-nets, RNNs, and temporal constraint models. Indeed, some of the works A close up of text on a black background

Description automatically generatedhave been done blend the above techniques, such as Park et al. [14].

Figure 2‑5 Detection of the optical flow in 3 consecutive images.

Source: Adapted from [20]

### Optical flow

Optic flow is the change of structured pixels with specific intensity in successive images, or in other words, Optical flow is the motion of objects among successive frames, caused by the relative movement among the object and camera. This make optical flow an ideal for encoding temporal information[11], [14], [21].

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 where  is the motion in the  direction,  is the motion in the direction, and  is a neural network that takes in two consecutive frames  (frame at time  ) and  (frame at time as input.

Computing optical flow with deep neural networks [22], [23] requires vast amounts of training data which is principally hard to obtain. This is because marking optical flow video footage requires a detailed finding of the precise motion of each point of a frame to the precision of the subpixels. To address the issue of labeling training data, many research works, [22], [24], [25] 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 [26], an open-source CGI movie with optical flow labeling rendered for numerous sequences, and Flying Chairs [24], a dataset of numerous chairs hovering around random backgrounds both generated from the virtual world using Computer Graphics.

### 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. Mainly human pose estimation is used in transferring motion from one person to another as used [27], to transfer motion between different domain videos specifically for animating static image by driving motion as shown in figure 2-6 [28] and facial expression transfer [29] between source and the target person.

Figure 2‑6 pose extraction

Source: Adapted from[27]

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.

### 3D convolutional tensor

The 3D convolutional tensor mechanism is one of the orthodox methods, which basically does not consider temporal information explicitly. Since it considers presenting video scene [30] as a 3-dimensional tensor meaning it takes the whole video as input, and the network eventually learns the relation between consecutive frames to preserve temporal consistency implicitly. In due course, this approach is not used frequently because of two fundamental reasons requires a high-efficiency machine, and the network becomes an entirely black box, which means hard to tune parameters basically done in training Deep Learning models.

### Recurrent temporal

Recurrent neural networks or RNNs are a type of neural network inherently ideal for analyzing data from time-series, and other sequential figures make it ideal for video analysis. Possibly it overcomes the black-box nature of 3D Conv nets by adding an additional parameter to tune the network. Recent works consider using LSTM (Long Short Time Memory.) which takes into account all previous frames as an input to minimize temporal residual error [21].

## Related Works

In the previous section, we discussed the temporal information (motion information) extraction mechanism. However, 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 problem listed in section 2.6. the results output video[18], [19].

To overcome the flickering effect, Chen et al. [19] consider temporal information along with spatial information. Specifically, they exploit previous frame optical flow to warp the current frame to impose temporal constraints, but this paradigm prone to occlusion and fast illumination change (since optical flow does not consider newly introduced pixels in given frame scene).

Another fine work by Chen et al. [31] 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 [11] ReCycle-GAN further extends 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 [21] 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*.*

## Related work summary

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

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, IS | Framewise image to image translation. |
| Recycle-GAN: Unsupervised Video Retargeting: ReCycle-GAN [11] | Viper, face, and flower datasets (more than 10,000 images) | Cycle-GAN with recurrent temporal predictor | Recurrent temporal predictor(pix2pix) | Generator Network | Human evaluation, IoU, pixel accuracy, Average class accuracy,IS | Temporal predictor basically fails to correctly predict, and no cycle consistency temporal cycle considered |
| MoCycle-GAN: Unpaired Video-to-Video Translation MoCycle-GAN [31] | 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[29] | 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, AED, and FID | No random input sizes. |
| Unsupervised Video-to-Video Translation Dina [30] | 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 [14] | 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 [32] 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[21] | 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 does not consider temporal cycle continuity |

As shown in the above table, researchers design complex architectures used in previous works to learn a mapping a domain to domain transfer in an unsupervised manner. However, those works do not consider the content difference between a real and translated video, which is a significant problem that causes object dislocation and disappearance. ReCycle-GAN does not consider the power of the temporal aware discriminator network. This thesis work was done to add the above two learning losses to the ReCycle-GAN network to improve the generated video even further.

CHAPTER THREE

# Materials and Methods

## 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.

## 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 [11]. 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).
* ***አዲስ (Adiss):*** is a local dataset for video retargeting containing two very popular politicians Prime minister Abiy Ahmed and Debretsion Gebremicahel making an interview press conference. Frame size is 256 x 256, and around 5000 images are included.

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

|  |  |
| --- | --- |
| Flower1 |  |
| Flower2 |
| Obama |  |
| Trump |
| Abiy |  |
| Debretsion |

Table 5 Viper Dataset Sample Examples

|  |  |  |  |
| --- | --- | --- | --- |
| ***Rain*** | ***Snow*** | ***Sunset*** | ***Day*** |
|  | | | |

## Development tools

For this research, numerous types of development tools are used to design and implement the proposed thesis work. The development tools section gives a description and justification of these development tools. These tools include prototype development tools and platforms, UML Modeling tools, and other relevant tools to the research. The following sections give a brief detail about these development tools.

## Design tools

Design tools are mediums that are used for the creation, presentation, and interpretation of design concepts. Edraw Max [33] 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 [33].

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

## Prototype development framework

### TensorFlow

TensorFlow is an open-source software library optimized for maximum-performance numerical modelling and processing. Its modular architecture can be easily implemented on a range of platforms such as Central Processing Units (CPUs), Graphical Processing Units (GPUs), Tensor Processing Units (TPUs). It can also be mounted on personal computers, clusters, handheld devices, and edge devices. Tensorflow Supports artificial learning, deep learning, and versatile numerical computing [34] The following diagram displays the power score of the deep learning system based on application, popularity, and interest [35].

The following diagram demonstrates the power score of the deep learning system based on application and popularity. As shown in the below 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.



Figure 3‑1 Deep Learning Framework comparison.

Source: Adapted from [36]

### OpenCV

OpenCV is an open-source computer vision software library intended to provide a shared infrastructure for image processing and computer vision applications [37]. It has Python, Java, C++, and MATLAB interfaces and supports nearly any operating system as well. OpenCV was developed for image processing, meaning that and feature and data structure was developed with the image processing engineers in mind.

### MATLAB Deep Network Designer

MATLAB deep network designer [38] is an application developed by MATLAB which developed for easy Design, Visualize, and train deep learning networks using drag & drop simple user interactive mechanism. This tool is a relief for AI developers, especially for complex network deep architectures and GAN networks. This even further helps Developers to track and debug errors on the early design stage.

## Baseline Works

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 understand the Spatio-temporal constraint effect better.
* 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.

## Feature Extraction Algorithm

Several states of art Deep CNN based architectures have been proposed over the last decades for the classification of images. These different state of art CNN based feature extraction architecture include ResNet, Inception, Xception, EfficientNet, and others.



Figure 3‑2 Benchmark Analysis on EfficientNet-B7

Source: Adapted from[39]

The above figure shows the top1 accuracy vs the number of parameters. The x-axis represents the number of trainable parameters. EfficientNet-B7. Other architectures like VGG have more than 155 parameters million and ResNet has round 60 million. EfficientNet-B7 has less number of parameters and operations compared to most architectures, which makes it able to run fast on different devices that have less computing. Perhaps, Other architectures like ResNet-50 have a fewer number of parameters and operations as shown in the figure above, but their accuracy fails as well. Based on the above observation, EfficientNet-B7[39] is used in this research for feature extraction in the task of computing feature map of a given image.

## Evaluation Methods

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

### Human Evaluation Study

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

### Inception Score

|  |  |
| --- | --- |
|  | eq.( .) |

The inception score is a commonly used evaluating algorithm for GANs. It uses a pre-trained inception V3 network (trained on ImageNet) to extract the features of both generated and real images. The IS (inception score) [40] for short, **measures the variety and the quality of the created images**. The superiority of the model is good if it has a high inception score. Further detail and limitations are found in [Appendix F](#_Appendix_F:_Inception).

CHAPTER FOUR

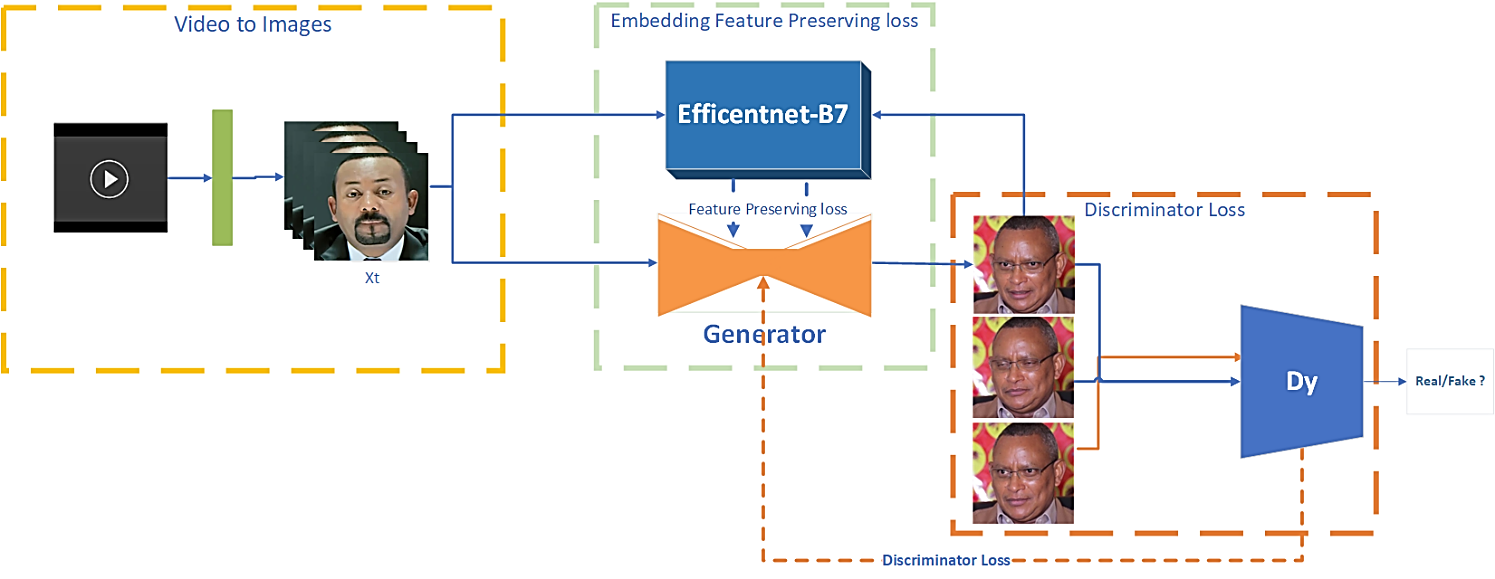
# Proposed loss function.

## 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 central building stones. This chapter can be ideally portioned into three major sections; the first introduces model Architecture to translate a given domain image into another domain—the second deals with Network optimizing loss functions. The last explains training Pseudocode to train our model.

## Model Architecture.

The model architecture has directly influenced by the architecture defined in *“Unpaired Image-to Image Translation using Cycle-Consistent Adversarial Networks”* [10] and *“Recycle-GAN: Unsupervised Video Retargeting”* [11] for Learning Domain Translation.



Adjustments have Generator Network Architecture been 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.

## Model Learning Functions

The key objective of this thesis work is to optimize the use of space-time knowledge. 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, , to a sequence of domain changed images, , With the exclusion of problems listed in section [2.5](#_Problems_in_Translation). The function is then to acquire the mapping of . Note that our model uses sequential unpaired image data as input during training.

### Proposed Network Learning Function.

Because we follow the GAN architecture, the vanilla adversarial loss is also used in our work, called . And the cycle consistency loss in Cycle-GAN [10] is adopted. Besides, the recurrent loss and the recycle loss in Recycle-GAN [11] are also leveraged. Meanwhile, this work introduces constrain to impel the model and improve the whole translation. The full loss function of our work is as follows:

|  |
| --- |
|  |

Where and 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.

Keeping in mind 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:

|  |  |  |
| --- | --- | --- |
|  |  | eq.( 4.1) |

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

#### Identity Loss

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

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

|  |  |
| --- | --- |
|  | eq.( 4.2) |

So, the full loss would be:

***where***: are generators,,and are discriminators respectively both domain and are samples from both domain datasets.

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

#### Feature Preserving Loss

Indeed, classic cycle-consistency does not essentially assurance the transformation to be semantically consistent. This is, as a result, it does not consider any semantic correspondence during the translation, and thus the system can accomplish textbook cycle-consistency (i.e., = 0) only if the inverse mapping recovers the original contents, regardless of how incorrect the forward mapping was.

|  |  |
| --- | --- |
|  | eq.( 4.3) |

***where:*** *mNET stand for pre-trained EfficientNet-B7*

By adding the above loss, we inspire the network to minimize the **Object Disappearance** problem list in [section 2.5](#_Problems_in_Translation) to have consistent semantics earlier and afterwards the translations. This thesis work uses EfficientNet-B7[39] as a feature extractor that enforces the content information that appears in the original image also should appear on translate. For example, if a person and a dog appear in image A so does in translated image A2B, albeit the style 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 [11], we adopt a recurrent temporal predictor to predict frames in the future based on the past frame details. The repeated deficit is as follows:

|  |  |
| --- | --- |
|  | eq.( 4.4 ) |

Where is a prediction of given and as concatenated input.

#### Recycle Loss

Merging image generator [10] and temporal prediction network. The recycle loss[11] across domains and time can be described as:

|  |  |
| --- | --- |
|  | eq.( 4.5 ) |

## 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](#_Model_Architecture.). Notice that the time dynamics of the translated videos should be close to those of the source videos. This thesis study uses flownet2 to measure optical flow [41]. The temporal warping is defined as;

## Temporal aware Discriminator

To improve visual quality further, a discriminator takes two consecutive generated images to decide whether it is real or fake is used. The Discriminator architecture and the output stay the same with Patch GAN [9] instead, the differences are just the input and the number of channels. Rather than differentiating between single frames, the discriminator network is designed in a way to observes three constitutive of synthesized frames and three constitutive of the real frames. This approach makes the discriminator network optimal because it takes into account the temporal nature of the video generation issue such as **Object Dislocation**.

|  |  |
| --- | --- |
|  | eq.( 4.6) |



Figure 4‑2 Temporal Discriminator Network

## Training Pseudocode

Training algorithms for this thesis work have been introduced in this section as this study compares earlier research, Cycle-GAN and ReCycle-GAN. Their training algorithms have present in [Appendix D and E](#_Appendix_D:_Cycle). As discussed in the preceding section, the previous approach does not consider content translation, which leads to Object Dislocation and Object Disappearance problem. However, this work emphasizes minimizing the content difference between fake and real images, as shown in training pseudocode **line-12.** On the other hand, this work modified the patch-GAN discriminator to make it a temporal aware network, **line-4,** and **line-13,** which care about the relation among consecutive three frames.

Table 6 Training pseudocode

|  |  |
| --- | --- |
| This thesis work Training pseudocode: | |
| 1 |  |
| 2 | Train D: |
| 3 |  |
| **4** |  |
| 5 |  |
| 6 | . |
| 7 | Train G: |
|  | *,* |
| 8 |
|  |
| 9 |  |
| 10 |  |
| 11 |  |
| **12** | ***feature\_preserving\_loss =*** |
| **13** |  |
|  | . |

CHAPTER FIVE

# Implementation of the Proposed work

## Overview

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

## Working Environment.

This section explains 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.
  + Operating system: Windows 10
  + Processor: Intel ® Core™ i7-2300QM CPU @ 2.00GHz
  + Graphics: Intel ® Graphics 3000
  + Primary Memory (RAM): 8.00 GB
  + System Type: 64-bit Operating System, x64-based Processor
* Desktop: The desktop computer is used for developing a video for video translation.
  + Operating system: windows 10
  + Processor: Intel ® Core™ i5-4580 CPU @ 3.29GHz x 4
  + Graphics: Intel ® HD Graphics 4600
  + GPU: GeForce RTX 2070 Super 6 GB RAM
  + Primary Memory (RAM): 14.00 GB
  + 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. Furthermore, TensorFlow-GPU 2.2 used. In the next section list of experiments class conducted for evaluating the proposed hypothesis are discussed.

## Environmental Setup

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

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

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

## Implement Cycle-GAN

A Cycle-GAN is made up overall of two GAN architectures: a generator and a discriminator. The generator architecture contains two separate models, Generator AB and Generator BA. In the same manner, the discriminator architecture contains an additional two architectural models, Discriminator A, and Discriminator B. table 7 shows convolutional layers of Cycle-GAN architecture.

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

G\_A2B = module.ResnetGenerator(input\_shape=(256,256, 3))  
G\_B2A = module.ResnetGenerator(input\_shape=(256,256, 3))

The discriminator network is equivalent to the discriminator's architecture in a Patch GAN network[9]. Basically, it takes an image of the shape of (256, 256, 3) and predicts whether the image is real or fake. This network composed of 5 convolutional layers denotes a 4 × 4 Convolution-Instance Normalization with LeakyReLU layer and stride 2. After the last layer, apply a convolution to produce a 1-dimensional output. The slope of leaky in leakyReLU was 0.2.

D\_A = module.ConvDiscriminator(input\_shape=(256,256, 3))  
D\_B = module.ConvDiscriminator(input\_shape=(256,256, 3))

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.

Table 7 Cycle GAN architecture convolutional layers

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

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 10e -5. Every network used ADAM optimizer with the momentum term set to 0.5 and the learning set to 0.0002. The Cycle GAN architecture is composed of the above four independent networks. The Generator's objective is to diminish the adversarial loss function against an adversary Discriminator, which constantly tries to maximize it. Similar to other network types of GAN is no different. The learning function has to be explicitly defined in order for the network to learn to translate the image.

self.combined = tf.keras.Model(inputs=[img\_A, img\_B], outputs=[valid\_A, valid\_B,reconstr\_A, reconstr\_B,img\_A\_id, img\_B\_id,img\_A\_id, img\_B\_id])

*#define loss function*  
d\_loss\_fn, g\_loss\_fn = gan.get\_adversarial\_losses\_fn(adversarial\_loss\_mode)  
cycle\_loss\_fn = tf.losses.MeanAbsoluteError()  
identity\_loss\_fn = tf.losses.MeanAbsoluteError()  
G\_loss = (A2B\_g\_loss + B2A\_g\_loss) + (A2B2A\_cycle\_loss + B2A2B\_cycle\_loss) \* cycle\_loss\_weight + (A2A\_id\_loss + B2B\_id\_loss) \* identity\_loss\_weight

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 and its authors' implementations on GitHub.

## Temporal Predictor Network Implementation

This thesis work uses Recycle-GAN temporal predictor network Px and Py for video retargeting, which identical to the pix2pix [9] generator network. However, the input layer has been modified to receive two successive previous images.

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

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

As shown in the code snip, *train\_p* function takes six argument variables. A, A\_1, and A\_2 are in domine A and the rest in domain B. Since pix2pix needs paired dataset A\_1 and A\_2 concatenated as an input, the network predicts A\_p, which is predicted frame-based given inputs. The loss is the L1 distance between A\_P and A, which is used to update the gradient weights.

*#input temporal predictor network*  
from temporal\_predictor import Generator   
Px = Generator(inputs)  
Py = Generator(inputs)

P\_optimizer = keras.optimizers.Adam(learning\_rate = 2e-4, beta\_1 = 0.5)  
*#A\_1, A\_2, B\_1 and B\_2 are the previous two frames in Domain A and Domain B*  
**@tf.function**  
def **train\_P**(A, A\_1, A\_2, B, B\_1, B\_2):  
 with tf.GradientTape() as pt:   
 A\_p = Px([A\_1, A\_2],training = True)  
 B\_p = Py([B\_1, B\_2],training = True)   
 xl1\_loss = P\_loss\_fn(A, A\_p)  
 Px\_loss = xl1\_loss \* LAMBDA  
 yl1\_loss = P\_loss\_fn(B, B\_p)  
 Py\_loss = yl1\_loss \* LAMBDA  
 P\_loss = (Px\_loss + Py\_loss)\* args.cycle\_loss\_weight  
*#update gradient weight*  
 P\_grad = pt.gradient(P\_loss, Px.trainable\_variables +Py.trainable\_variables)  
 P\_optimizer.apply\_gradients(zip(P\_grad, Px.trainable\_variables + Py.trainable\_variables))  
 return A\_p, B\_p, {'Px\_loss': Px\_loss, 'Py\_loss': Py\_loss}

## Feature Preserving Loss Implementation

As discussed in the previous sections, feature preserving loss aims to minimize content information deference between real and the translated fake images. To do so efficientnet-b7 pre-trained model is imported. Since the aim is to extract the feature map the input image, the last four layers are removed, as shown in code snip. Using a pre-trained EfficientNetB7 model, the new Keras model has been created. Another method was then defined to measure the content of two pictures.

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)

in order to compute feature map new function, *get\_content\_feature* is defined, which returns a feature map of the input images. Then compute feature preserving loss between the real image and fake image pair sets has been computed then the loss has been used to update network weight. Meaning the content loss would be the L1 distance between *M\_B, M\_B2A, and M\_A, M\_A2B*.

def **get\_content\_features**(a,b):  
 return mNet(a), mNet(b)

M\_A, M\_A2B = get\_content\_features(A, A2B)  
M\_A\_A2B = identity\_loss\_fn(M\_A, M\_A2B)  
M\_B, M\_B2A = get\_content\_features(B, B2A)  
M\_B\_B2A = identity\_loss\_fn(M\_B, M\_B2A)

## Temporal aware Discriminator Network Implementation

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

def **build\_discriminator**(n):  
 inputA = tf.keras.Input(shape = (256,256,3))  
 inputB = tf.keras.Input(shape = (256,256,3))  
 inputC = tf.keras.Input(shape = (256,256,3))  
 h = tf.keras.layers.Concatenate()([x, y, z])  
   
 d1 = conv2d(h, 64, 4, 2)  
 d2 = conv2d(d1, 128, 4, 2)  
 d3 = conv2d(d2, 256, 4, 2)  
 d4 = conv2d(d3, 512, 4, 2)   
 d5 = conv2d(d4, 1, 4, 1)   
 x = tf.keras.Model(img,d5,name=n)

return x

The Discriminator architecture and the output stay the same with Patch GAN [9]; instead, the differences are just the concatenated input and the number of channels. This enforces the network to strictly focus on the relation among generated images and its relation with two previous images.

A\_d\_logits = D\_A((A,A\_1,A\_2), training=True)  
 B2A\_d\_logits = D\_A((B2A,B2A\_1,B2A\_2), training=True)  
 B\_d\_logits = D\_B((B,B\_1,B\_2), training=True)  
 A2B\_d\_logits = D\_B((A2B,A2B\_1,A2B\_2), training=True)

## 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. flownet2 [[5]](#footnote-5) is used as a temporal extractor [22].

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*

## Experiment Class

To evaluate the essence of temporal information for video translation testing the initial hypothesis is mandatory. Five different classes of experiments are conducted, as shown below on the table for each dataset group. The first three experiments focus on video translation on flower and viper datasets, while the rest two are basically for video retargeting on Obama-Trump and (አዲስ) Adiss Datasets.

The first class is all about vanilla Cycle-GAN image translation on a given sequence of images, considering the spatial domain only. The second is regarding consider using feature preserving loss. The third one includes temporal Discriminator build up on the second experiment. The fourth experimental class uses vanilla ReCycle-GAN aiming video retargeting, and the last one merges ReCycle-GAN with temporal discriminator.

Table 8 lists of experimental classes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Notation*** | ***Experiment*** | ***Training Epochs*** | ***Dataset*** | ***Model used*** |
| CC | CycleGAN baseline implementation in TF-Keras with settings according to suggestions from the original article. | 20 epochs | Flower dataset and Viper dataset | Cycle-GAN |
| CC+CP | CycleGAN baseline generator trained with additional feature preserving loss | 20 epochs | Flower dataset and Viper dataset | Cycle-GAN and EfficientNet-B7 |
| CC+CP+TD | CycleGAN baseline generator trained with additional feature preserving loss and temporal Discriminator Network | 20 epochs | Flower dataset and Viper dataset | Cycle-GAN, flownet2, and Temporal aware discriminator. |
| RC | ReCycle-GAN baseline implementation in TF-Keras with settings according to suggestions from the original article | 30 epochs | Obama trump dataset and Adiss Dataset | ReCycle-GAN |
| RC+TD | ReCycle-GAN with temporal Discriminator | 30 epochs | Obama trump dataset and Adiss Dataset | ReCycle-GAN & temporal Discriminator |

Total of 10 experiments are conducted.

CHAPTER SEVEN

# Evaluation, Results, and Discussion

## 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? Moreover, this thesis work proposes a hypothesis that “adding temporal consistency constrain would improve temporal consistency between successive frames.”.*

## Video to video Translation

The video to video translation takes a video from the scene as an input to generate an equivalent video in other domains with the consideration of preserving temporal information. This work conducts different training experiments to explain the qualitative and quantitative outcomes of comparing the baselines on which the study is based 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](#_Proposed_work). The models compared in the evaluation are shown in Table 7.

### Flower to Flower

Figure 6-1 demonstrates this method's synthesized frames on the Flower dataset.[[6]](#footnote-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 relatively small. Our algorithm can generally preserve the consistency of a sense and content information based on the given input image. The translated flower in each target field retains a continuity for much of the time, with input flower at a different domain. Table 8 shows the inception score of experimental runs of the network.

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Methods | Flower | | | | |
| IS | | | Human evaluation study[[7]](#footnote-7) | |
| Real dataset | | 1.165 0.030 | 1.248±0.055 |  | |
|  | | Domain A | Domain B | Domain A | Domain B |
| CC | | 1.022±0.002 | 1.102±0.031 | 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.

Based on Quantitative results, the simplest model *CC*, which only trained only on the Cycle-loss, has the lowest score among the models, indicating that the cycle GAN architecture is not complex enough for the video to video translation. Since CC only considers the spatial domain, the translation lacks knowledge of the temporal domain. The CC+CP model shows a slight improvement in the inception score compared to CC, which indicates that in the model CC+CP feature preserving loss does not have much impact.

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 stability brought by the spatial-temporal constraint. Figure 6-1 shows that even if Temporal Continuity can be preserved by model CC+CP+TD, it contains many artifacts because of the vanishing gradient problem. It does not really have much weight changes after some epochs (meaning the gradient becomes very low-approximate to zero, multiplication of a small number further minimizes the loss) so any gradient update does change almost nothing in backpropagation.The discriminator network of CC+CP+TD becomes too complicated to be tricked by the generator network, As seen in figure 6-2 below, the discriminator loss become slightly similar to zero, and the generator loss will escalate to one.

|  |  |
| --- | --- |
| Input |  |
| CC |
| CC+CP |
| CC+CP+TD |
| Input |  |
| CC |
| CC+CP |
| CC+CP+TD |

Figure 6‑1 flower to flower translation result

The real images labled Input that the synthetic images are based on. The second-row is the result of Cycle-GAN, Third-row shows Cycle-GAN with Feature preserving loss, the last include Temporal Aware discriminator and Temporal warping.

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. However, we ware searching for the Nash equilibrium of the two networks (Generator and Discriminator) to balance each other. In other hand, Even though the Inception score of CC+CP outperforms vanilla Cycle-GAN, the Human evaluation study shows CC excel CC+CP. Indeed, The CC+CP network even amplifies the flickering effect.

Vanishing gradient and gradient explosion problem minimized by applying gradient penalty to the discriminator network. Applying the gradient penalty to the CC+CP+TD improves the generated output video quality, as shown in figure 6-3 below. (but Human evaluation study and Inception score experiment the gradient penalty not included for a fair comparison.)

|  |  |
| --- | --- |
| Discriminator loss |  |
| Generator loss |  |

Figure 6‑2 weight Vanishing problem on CC+CP+TD

Top Discriminator network, and bottom Generator network,

Red: CC+CP+TD, Blue: CC+CP, Orange: CC

|  |  |
| --- | --- |
| result Output |  |
| Generator loss |  |
| Discriminator loss |  |

Figure 6‑3 CC+CP+TD with gradient penalty

Number of examples (top) Output sample (middle) Generator loss,

and (bottom) Discriminator loss

### 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 video to Sunset video and vice versa. It takes around 1.6 sec per image and a total of 136.8 hours to train. The task of Sunset to Day is shown to explain the influence of our exploitation of the proposed solution of this thesis; therefore, more focus on the video quality improvements shown in Figure. 6-4 and figure 6-5,

|  |  |
| --- | --- |
| Input |  |
| CC |
| CC+CP |
| CC+TD |
| Input |  |
| CC |
| CC+CP |
| CC+TD |

Figure 6‑4 Sunset to Day translation Output Result

Eventually, this work positively improves visual quality as confirmed in IS and Human evaluation study results (Table 9). 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.

Table 10 IS score and Human evaluation study Result on Viper Dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Methods | Day to Sunset | | | |
| IS | | Human evaluation study[[8]](#footnote-8) | |
| Real data | *3.56s±0.21* | *3.81±0.44* |
|  | Day | Sunset | Day | Sunset |
| 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%** |

Bold values indicate the best results in the experiments.

CC+CP performs a much better in Sunset to Day datasets Compared to the flower translation as shown in table 9 above. I suppose it is because the Efficientnet-B7 feature preserving network trains on ImageNet. I did not think there were substantial flower blooming (in fact it only contains 1197 images of flowers, which is around 0.0084% of the entire dataset.) instances in training so the network might not be able to extract adequate features in flower video. Hence, the network performed badly due to this cause.

CC+CP+TD was impacted by the vanishing of gradient problem in the tiny dataset as seen in the flower to flower translation, and 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 detailed information, but our model generates a decent result. Figure 6-5 shows the positive inpact toward content translation.

|  |  |  |
| --- | --- | --- |
| Input | CC | CC+TD+CP |
|  | | |

Figure 6‑5 Comparison between Cycle-GAN with this thesis work on Sunset to Day

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

### Face to Face

In this experiment, we evaluate Obama to Trump translation using the Recycle-GAN and ReCycle-GAN with temporal discriminator. Both approaches are capable of accessing the stylistic facial gestures of Donald Trump[[9]](#footnote-9) ***(Please note that the photos are very minimal in their representation*)**. Nevertheless, mouth motion slightly Differ, as shown in the above figure 6-7. For example, in Trump to Obama translation, our model fetches trump mouth movement more reasonably than the comparative model Recycle GAN.

|  |  |
| --- | --- |
| Input |  |
| RC |
| RC+TD |
| Input |  |
| RC |
| RC+TD |

Figure 6‑6 Obama to Trump Translation Result

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

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 cannot be implemented since this work focuses on video Retargeting). 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 base work.

Table 11 Obama to Trump Inception Score and Human evaluation Study.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Methods | Obama to Trump | | | |
| IS | | Human evaluation study[[10]](#footnote-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% |

Bold values indicate the best results in the experiments.

RC model has been suffered from image flipping problem as shown below in figure 6-8, perhaps inherited from the Convolutional neural network. (but for pair comparison in figure 6-8, RC output images have been aligning according to the input.) however, RC+TD could maintain input video alignment. On the other hand, the temporal discriminator network positively impacts video retargeting based on qualitative and quantitative evaluations discussed above; however, it also forces the model to make the L1 distance between consecutive frames become very small, limiting motion change.



Figure 6‑7 RC Trump Generated image sequences

CHAPTER SEVEN

# Conclusion and Future work

## 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 that have a relation between them. This approach is non-trivial since the underlying flickering problem effect is in the output video.

The purpose of this study was to improve temporal coherence for the video to video translation by adding constraints to the GAN network learning function trained on the unpaired dataset. Which 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, 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.

Differ from early approchs,which gives focus only to the Generated image look real or fake based on Spatial information only. However, 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 Combines the above two losses to preserve temporal information.

Inpact of dataset,shall be discussed

Compared with baseline works Cycle-GAN[10], and ReCycle-GAN[11], 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.

## Limitation and Future work

The proposed method does not 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. 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 considerably 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] “Attempts on Real Time Style Transfer – mc.ai.” [Online]. Available: https://mc.ai/attempts-on-real-time-style-transfer/. [Accessed: 22-Sep-2020].

[7] “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].

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] “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].

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

[26] 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.

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

[28] 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.

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

[30] D. Bashkirova, B. Usman, and K. Saenko, “Unsupervised Video-to-Video Translation,” no. Nips, 2018.

[31] 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.

[32] 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.

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

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

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

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

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

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

[39] 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.

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

[41] 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, Evaluation Metrics and Residual Blocks

The loss function is used to calculate the error of an event, which is used to determine the error between the output of 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 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 overall mean 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.

Inception Score: The IS takes an image list and returns the resulting score, which is a floating number. The score is an indication of how realistic the output of a GAN is, the score allows us to measure two criteria:

* Are images diverse (e.g. each image produced is of a different kind)?
* Checks the created images quality?

The score will be high if both things are right. If either is wrong, the score is low. Better results will be able to produce many different images through the GAN network. Ideally, the result could be between null and infinite, but in fact, the result does not land at an infinite number of floats.

The statistical formula known as the Kullback-Leibler divergence is used to produce the Inception score. The KL difference is a measure of how similar/different two distributions of probabilities are. When the Generated distributions are different, the KL divergence is high.

IS limitation

* The score is limited by what the Inception classifier network classifier can detect, which has a direct link to the training dataset (imageNet). meaning If the network is learning to generate something not present in the classifier's training data
* Does not consider Time serious data, Is focus on similarity or difference of the network generated image distribution score result is always small on data like videos because Adjacent frames are quite similar.

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 the residual block. In the residual block, the output of the lower layer of the network feeds into the input of the layer.

****

## Appendix B: Result on Different epochs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 30th epoch | 25th epoch | 20th epoch | 15th epoch | 10th epoch | 5th epoch |
|  | | | | | |

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 20th epoch | 15th epoch | | 10th epoch | | 5th epoch | | 1st epoch | |
|  | | | | | | | | |
| 45th epoch | | 40th epoch | | 35th epoch | | 30th epoch | | 25th 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: |
|  |
| Train D: |
|  |
|  |
|  |
| . |
| Train G: |
|  |
|  |
|  |
|  |
| . |

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

|  |
| --- |
| Cycle-GAN with Feature Preserving Training pseudocode: |
|  |
| Train D: |
|  |
|  |
|  |
| . |
| Train G: |
|  |
|  |
|  |
| ***feature\_preserving\_loss =*** |
|  |
| . |

## Appendix F: Inception Score:

1. Some authors see GAN in other perspective rather adversarial: collaboration of two networks to Mimic a give real data distribution. [↑](#footnote-ref-1)
2. conditioning variable *C* could be any type of information. Like Image, tabular information or…. [↑](#footnote-ref-2)
3. Object disappearance and object dislocation in a situation like face to face translation wouldn't be a question. [↑](#footnote-ref-3)
4. These present papers are only substantial papers that directly relate to thesis work, and all are from 2017 onwards. [↑](#footnote-ref-4)
5. This thesis work uses Flownet2 because it is commonly used in previous works and its tensor-pack implementation is available. [↑](#footnote-ref-5)
6. It takes around 1.6 sec per image and totally training takes about 57.8 hours. [↑](#footnote-ref-6)
7. Human Evaluation User study for flower translation found at: <https://forms.gle/dG3jo9iVskvXxLWLA> [↑](#footnote-ref-7)
8. Human Evaluation User study for Day to Sunset found at: <https://forms.gle/xbkt9aFxA4YmNFnH6> [↑](#footnote-ref-8)
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> [↑](#footnote-ref-9)
10. Human Evaluation User study for Day to Sunset found at: <https://forms.gle/ydauVZbixebUJVYJA> [↑](#footnote-ref-10)