A PRELIMINARY REPORT ON

" IMAGE TO IMAGE SYNTHESIS USING CYCLEGANS"

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

Anshula Awasthi	**18U101**
Raj Dilip Mesta	**18U119**
Chaitanya Sanjeev Wagh	**18U136**
Yuvraj Singh	**18U142**

DEPARTMENT OF COMPUTER ENGINEERING

STES'S SMT. KASHIBAI NAVALE COLLEGE OF ENGINEERING

VADGAON BK, OFF SINHGAD ROAD, PUNE 411041

SAVITRIBAI PHULE PUNE UNIVERSITY

2020-2021

Sinhgad Institutes

CERTIFICATE

This is to certify that the project report entitled

" Image to Image Synthesis using CycleGans "

Submitted by

Anshula Awasthi	Exam No : **18U101**
Raj Dilip Mesta	**18U119**
Chaitanya Sanjeev Wagh	**18U136**
Yuvraj Singh	**18U142**

is a bonafide student of this institute and the work has been carried out by him/her under the supervision of **Prof. P. N. Mahalle** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

(Dr. P. N. Mahalle)
Guide
Department of Computer Engineering

(Dr. P. N. Mahalle)
Head,
Department of Computer Engineering

(Dr. A. V. Deshpande)
Principal,
Smt.Kashibai Navale College of Engineering Pune – 41

Place : Pune

Date :

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely fortunate to have got this all along the completion of my project work. Whatever we have done is only due to such guidance and assistance and we would not forget to thank them.

We respect and thank our project Guide Dr. P. N. Mahalle, who took keen interest in our project work and guided us all along, by providing all the necessary information for developing a good project. We also thank him for giving me an opportunity to do the project work and providing us all support and guidance which help us to complete our project on time.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staff of the Computer department which helped us in successfully completing our project work. Also, we would like to extend our sincere regards to all the non-teaching staff of the Computer department for their timely support.

ANSHULA AWASTHI
RAJ DILIP MESTA
CHAITANYA SANJEEV WAGH
YUVRAJ SINGH

ABSTRACT

In the artificial intelligence sector, deep learning has achieved great success, and several deep learning models have been created. One of the deep learning models is Generative Adversarial Networks (GAN), and it has become a recent research hotspot. Generative modeling is an unsupervised task in machine learning involving the automatic discovery and learning of the regularities or patterns in input data in such a way that the model can generate or produce new examples that could have been plausibly extracted from the original dataset. GANs have now been widely studied due to the immense potential for applications, including image and vision computing, video and language processing, etc. In this report, we have presented a CycleGAN model to translate photos of horses to zebras, and back again.. The whole project is divided into three tasks. 1.To load and prepare the image translation dataset for modeling 2.to train a pair of CycleGAN generator models for translation 3.To load saved CycleGAN models and use them to translate photographs.

Keywords: Deep Learning; Artificial intelligence; Generative Adversarial Networks (GAN); CycleGAN;

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Image-to-image translation involves creating, with a particular alteration, a new synthetic version of a given image, such as converting a zebra landscape into a horse. Usually, training a model for translation of image-to-image involves a large dataset of paired examples.. These datasets, including images of paintings by long-dead artists, can be difficult and costly to plan and in some cases impossible.

CycleGAN is a methodology that requires automated training without paired examples of image-to-image translation models. The models are trained in an unsupervised manner employing a collection of images from the source and target domain that don't have to be related in any way. The CycleGAN extends the GAN architecture. Two generator models and two discriminator models are trained simultaneously in this.

Using a training set of aligned image pairs, image-to-image conversion is a class of vision and graphics problems where the objective is to learn the mapping between an input image and an output image.[1] Examples of image-to-image translation include: Translating zebra landscapes to horse landscapes (or the reverse). Translating paintings to photographs (or the reverse). Translating horses to zebras (or the reverse). Traditionally, a dataset consisting of paired examples includes training and an image-to-image translation model. That is, a huge dataset of several input image X examples (e.g. zebra landscapes) and the same image with the desired change that can be used as a predicted output image Y (e.g. horse landscapes). A limitation is the need for a paired training dataset. These datasets, such as images of different scenes under different circumstances, are difficult and costly to prepare.

It can be difficult and costly, however to obtain paired training data. It can be even more difficult to obtain input-output pairs for graphics tasks such as artistic stylization, because the desired output is extremely complex, usually requiring artistic authoring. For many tasks, like object transfiguration (e.g., zebra <-> horse), the desired output is not even well-defined.[1]. This is known as the unpaired image-to-image conversion problem.

CycleGAN is an effective method of unpaired image-to-image conversion. CycleGAN is an approach using the generative adversarial network, or GAN, model architecture to train image-to-image translation models.

1.1 Motivation

Deep learning models are getting more attention in the area of computer vision and recognition tasks.Deep learning based methods are able to exploit the hidden relations in an image data and learn better features for representation of raw input data.

The CycleGAN does not require a dataset of paired images, unlike other GAN models for image translation. For example, if we are interested in translating photographs of oranges to apples, we do not require a training dataset of oranges that have been manually converted to apples. This enables a translation model to be built on issues where there might be no training datasets, such as translating paintings into photographs. Therefore using CycleGANs are more beneficial than traditional GANs.

1.2 Problem Definition

We present an approach for learning to translate an image from a source domain X to a target domain Y in the absence of paired examples with the help of Cycle Generative Adversarial Networks.

Chapter 2

Literature Survey

Table 2.1: Literature Survey

Sr. no	Author/Paper name	Proposed System	Analysis
1.	"Unpaired Image to Image Translation using Cycle Consistent Adversarial Network/" , by Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros[1]	In this paper, they present a method that can learn to do the same: capturing special characteristics of one image collection and figuring out how these characteristics could be translated into the other image collection, all in the absence of any paired training examples. They seek an algorithm that can learn to translate between domains without paired input-output examples.	The algorithm can be applied to a wide range of applications, including collection style transfer, object transfiguration, season transfer and photo enhancement. We also compare against previous approaches that rely either on hand-defined factorizations of style and content, or on shared embedding functions, and show that our method outperforms these baselines.

Sr. no	Author/Paper name	Proposed System	Analysis
2.	"Image-to-Image Translation with ConditionalAdversarial Networks", Phillip Isola Jun-Yan Zhu Tinghui Zhou Alexei A. Efros[2]	They formed networks that not only learn the mapping from input image to output image, but also learn a loss function to train this mapping. This makes it possible to apply the same generic approach to problems that traditionally would require very different loss formulations.	They demonstrate that this approach is effective at synthesizing photos from label maps, reconstructing objects from edge maps, and colorizing images, among other tasks
3.	"Generative-Adversarial Networks" by Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza[3]	They proposed a new framework for estimating generative models via an adversarial process, in which they train simultaneously two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G..	This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to 1/2 everywhere.

Sr. no	Author/Paper name	Proposed System	Analysis
4.	"Learning from Simulated and Unsupervised Images through Adversarial Training" Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind[4]	They propose Simulated+Unsupervised (S+U) learning, where the task is to learn a model to improve the realism of a simulator's output using unlabeled real data, while preserving the annotation information from the simulator. They developed a method for S+U learning that uses an adversarial network similar to Generative Adversarial Networks (GANs), but with synthetic images as inputs instead of random vectors	They made several key modifications to the standard GAN algorithm to preserve annotations, avoid artifacts, and stabilize training: (i) a 'self-regularization' term, (ii) a local adversarial loss, and (iii) updating the discriminator using a history of refined images. We show that this enables generation of highly realistic images, which we demonstrate both qualitatively and with a user study.

Sr. no	Author/Paper name	Proposed System	Analysis
5.	"Improved Techniques for Training GANs" by Tim Salimans, Ian Goodfellow, Wojciech Zaremba	They present a variety of new architectural features and training procedures that we apply to the generative adversarial networks (GANs) framework. They focus on two applications of GANs: semi-supervised learning, and the generation of images that humans find visually realistic. Unlike most work on generative models, their primary goal is not to train a model that assigns high likelihood to test data, nor do they require the model to be able to learn well without using any labels.	Using their new techniques, they achieved state-of-the-art results in semi-supervised classification on MNIST, CIFAR-10 and SVHN. The generated images are of high quality as confirmed by a visual Turing test: our model generates MNIST samples that humans cannot distinguish from real data, and CIFAR-10 samples that yield a human error rate of 21.3%. They also present ImageNet samples with unprecedented resolution and show that our methods enable the model to learn recognizable features of ImageNet classes.

2.1 Related work :

Generative Adversarial Networks (GANs) The key to GANs' success is the idea of an adversarial loss that forces the generated images to be, in principle, indistinguishable from real images. This is particularly powerful for image generation tasks, as this is exactly the objective that much of computer graphics aims to optimize. We adopt an adversarial loss to learn the mapping such that the translated image cannot be distinguished from images in the target domain.

Image-to-Image Translation The idea of image-to-image translation goes back at least to Hertzmann et al. 's Image Analogies, who employ a nonparametric texture model on a single input-output training image pair. More recent approaches use a dataset of input-output examples to learn a parametric translation function using CNNs. Our approach builds on the "pix2pix" framework of Isola et all which uses a conditional generative adversarial network [14] to learn a mapping from input to output images. However, unlike these prior works, we learn the mapping without paired training examples

Neural Style Transfer is another way to perform image-to-image translation, which synthesizes a novel image by combining the content of one image with the style of another image (typically a painting) by matching the Gram matrix statistics of pre-trained deep features. Our main focus, on the other hand, is learning the mapping between two domains, rather than between two specific images, by trying to capture correspondences between higher-level appearance structures. Therefore, our method can be applied to other tasks, such as painting→ photo, object transfiguration, etc. where single sample transfer methods do not perform well.

Cycle Consistency The idea of using transitivity as a way to regularize structured data has a long history. In visual tracking, enforcing simple forward-backward consistency has been a standard trick for decades . In the language domain, verifying and improving translations via "back translation and reconciliation" is a technique used by human translators. More recently, higher-order cycle consistency has been used in structure from motion , 3D shape matching, co-segmentation, dense semantic alignment, and depth estimation.here in this project we use a cycle consistency loss as a way of using transitivity to supervise CNN training. In this work, we are introducing a similar loss to push G and F to be consistent with each other.

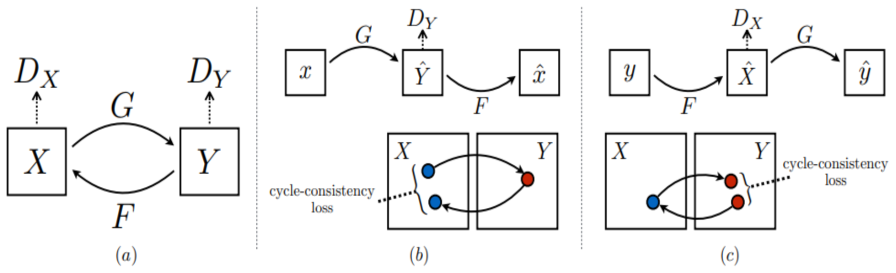

Fig 2.1   Cycle-Consistency Loss

Our model contains two mapping functions G : X → Y and F : Y → X, and associated adversarial discriminators DY and DX. DY encourages G to translate X into outputs indistinguishable from domain Y , and vice versa for DX, F, and X. To further regularize the mappings, we introduce two "cycle consistency losses" that capture the intuition that if we translate from one domain to the other and back again we should arrive where we started: (b) forward cycle-consistency loss: x → G(x) → F(G(x)) ≈ x, and (c) backward cycle-consistency loss: y → F(y) → G(F(y)) ≈ y

Chapter 3

Software Requirements Specification

3.1 Introduction

The software requirements are description of features and functionalities of the target system. The functional requirements of our project is two system features i.e CycleGan Model and UI Window. The non-functional requirements contribute to the system performance that it should return the accurate translated image that the user wants. The System requirements are the dataset which we are going to provide as an input to CycleGan Model and software and hardware requirements.

3.2 Functional Requirements

3.2.1 System Feature 1 : Uploading Image on UI

Description and Priority

Capturing special characteristics of uploaded images and figuring out how these characteristics could be translated into the other image.

Stimulus / Response Sequences

The input to this feature is an image given from the available dataset . This input is further processed and given to the CycleGan model.

Functional Requirements

The UI sends the uploaded image to the CycleGan model for translation.

3.2.2 System Feature 2 :CycleGan Model

Description and Priority

The feature used to translate the uploaded image using two mapping functions and two adversarial discriminators to distinguish between the generated and original images.

Stimulus / Response Sequences

The input to this feature is the original image which the user wants to translate.

Functional Requirements

This feature then returns the translated image to UI Window for the user to get their required image.

3.3 Nonfunctional Requirements

3.3.1 Performance Requirements

The user should get the required translated image wherever an image is uploaded. The labelling must be accurate. All the elements must be labelled whether missing or present. The model should be able to distinguish between generated and uploaded images accurately. The response time of the system should be as short as possible.

3.3.2 Software Quality Attributes

Reliability : The model should be reliable to give the resulting image with high resolution and as per the user wants.

Maintainability : As this is a real time system, maintenance should be done in less time and the system should have easy maintainability.

Reusability : As this is a deep learning model it should have reusable capabilities with the help of transfer learning.

Resource Utilization : As we know, for processing images larger computational power is required. So to perform well, a well defined system should utilize all available resources efficiently.

3.4 System Requirements

3.4.1 Database Requirements

1.Dataset

We note that decent results can often be obtained even on small datasets. Our facade training set consists of just 400 images, and the day to night training set consists of only 91 unique webcams. On datasets of this size, training can be very fast: the results took less than two hours of training on a single Pascal Titan X GPU. At test time, all models run in well under a second on this GPU.

3.4.2 Software Requirements

1. Windows operating system
2. Python
3. Anaconda Environment

3.4.3 Hardware Requirements

1. Minimum 8 GB RAM
2. Minimum 2GB GPU
3. Processor minimum i5

3.4.4 Analysis Models: SDLC Model to be applied

Agile Model :

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working soft- ware products. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on vari- ous areas like :

- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing
- Acceptance Testing

At the end of the iteration, a working product is displayed to the customer and impor- tant stakeholders.

Following are the Agile Manifesto principles :

- Individuals and interactions In Agile development, self-organization and motiva- tion are important, as are interactions like co-location and pair programming.

- Demo working software is considered the best means of com- munication with the customers to understand their requirements, instead of just depending on documentation.

- Customer collaboration As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.

- Responding to change Agile Development is focused on quick responses to change and continuous development.

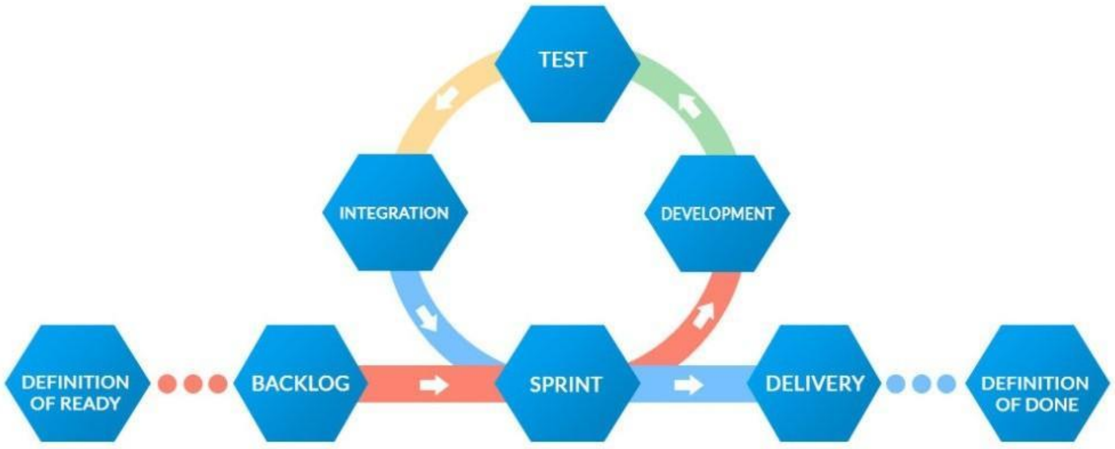

Figure 3.1: Agile model

- **Planning :**

  In this phase,we planned the various timelines required for our project. As this domain was relatively untouched by us in the past, the planning phase in this project took longer than other projects. In this phase, we planned how we will approach this project? Technologies required to complete this project.Also technologies required for this project were new to us. So, in addition to project work we had planned how we will understand particular technology. Also, requirements from this project were gathered.

- **Requirements Analysis :**

  In this phase, requirements collected in the planning phase were analyzed. In this project the main requirement is to detect violence in a video in minimum time. Also there were other requirements such as generating alerts, making entries in databases about violent incidents. Those requirements were analyzed thoroughly.

- **Design :**

  According to the requirement phase, the system was designed to satisfy the requirements. While designing, every requirement was considered and a system was developed to get optimal performance.

- **Coding :**

  As per system design, various models will be constructed independently. E.g. Alert Module, Deep learning model, GUI etc.

- **Unit Testing :**

  Those modules constructed in the coding phase will be tested as a unit.

- **Acceptance Testing :**

  Those modules constructed in the coding phase will be integrated and the whole system will be tested for acceptance on various parameters.

Chapter 4

System Design

We present a model that can learn to do the same: capturing special characteristics of one image collection and figuring out how these characteristics could be translated into the other image collection, all in the absence of any paired training examples. Our model includes two mappings G : X → Y and F : Y → X. In addition, we introduce two adversarial discriminators DX and DY , where DX aims to distinguish between images {x} and translated images {F(y)}; in the same way, DY aims to discriminate between {y} and {G(x)}. Our objective contains two types of terms: adversarial losses for matching the distribution of generated images to the data distribution in the target domain; and cycle consistency losses to prevent the learned mappings G and F from contradicting each other. It is also demonstrated that the performance improvement is statistically significant.

4.1 System Architecture

Consider the problem where we are interested in translating images from zebra to horse and horse to zebra.

We have two collections of photographs and they are unpaired, meaning they are photos of different locations at different times; we don't have the exact same scenes in horse and zebra.

- Collection 1: Photos of zebra landscapes.
- Collection 2: Photos of horse landscapes.

We will develop an architecture of two GANs, and each GAN has a discriminator and a generator model, meaning there are four models in total in the architecture.

The first GAN will generate photos of horse given photos of zebras, and the second GAN will generate photos of zebras given photos of horses.

- GAN 1: Translates photos of zebra (collection 1) to horse (collection 2).
- GAN 2: Translates photos of horse (collection 2) to zebra (collection 1).

Each GAN has a conditional generator model that will synthesize an image given an input image. And each GAN has a discriminator model to predict how likely the generated image is to have come from the target image collection. The discriminator and generator models for a GAN are trained under normal adversarial loss like a standard GAN model.

We can summarize the generator and discriminator models from GAN 1 as follows:

- Generator Model 1:
  - Input: Takes photos of zebra (collection 1).
  - Output: Generates photos of horse (collection 2).
- Discriminator Model 1:
  - Input: Takes photos of horse from collection 2 and output from Generator Model 1.
  - Output: Likelihood of image is from collection 2.

Similarly, we can summarize the generator and discriminator models from GAN 2 as follows:

- Generator Model 2:
  - Input: Takes photos of horse (collection 2).
  - Output: Generates photos of zebras (collection 1).
- Discriminator Model 2:
  - Input: Takes photos of zebra from collection 1 and output from Generator Model 2.
  - Output: Likelihood of image is from collection 1.

So far, the models are sufficient for generating plausible images in the target domain but are not translations of the input image. Each of the GANs are also updated using cycle consistency loss. This is designed to encourage the synthesized images in the target domain that are translations of the input image.

Cycle consistency loss compares an input photo to the Cycle GAN to the generated photo and calculates the difference between the two, e.g. using the L1 norm or summed absolute difference in pixel values. There are two ways in which cycle consistency loss is calculated and used to update the generator models each training iteration.

The first GAN (GAN 1) will take an image of a zebra landscape, generate an image of a horse landscape, which is provided as input to the second GAN (GAN 2), which in turn will generate an image of a zebra landscape. The cycle consistency loss calculates the difference between the image input to GAN 1 and the image output by GAN 2 and the generator models are updated accordingly to reduce the difference in the images.

This is a forward-cycle for cycle consistency loss. The same process is related in reverse for a backward cycle consistency loss from generator 2 to generator 1 and comparing the original photo of horse to the generated photo of horse.

- Forward Cycle Consistency Loss:
    - Input photo of zebra (collection 1) to GAN 1
    - Output photo of horse from GAN 1
    - Input photo of horse from GAN 1 to GAN 2
    - Output photo of zebra from GAN 2
    - Compare photo of zebra (collection 1) to photo of zebra from GAN 2

- Backward Cycle Consistency Loss:
    - Input photo of horse (collection 2) to GAN 2
    - Output photo of zebra from GAN 2
    - Input photo of zebra from GAN 2 to GAN 1
    - Output photo of horse from GAN 1
    - Compare photo of horse (collection 2) to photo of horse from GAN 1

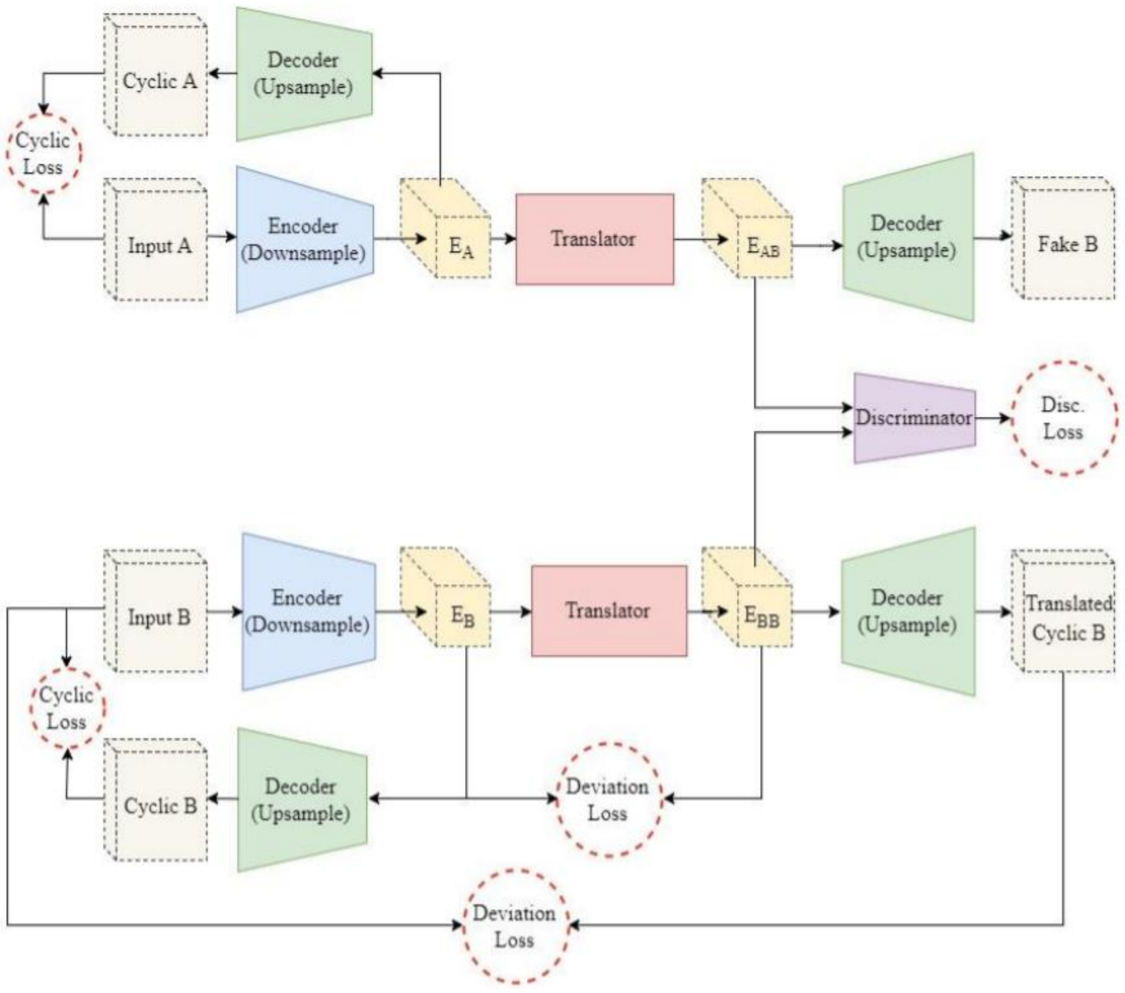

Figure 4.1: System Architecture

4.2 Data Flow Diagrams

The figure below shows a context Data Flow Diagram that is drawn for Image to Image Synthesis. It contains a process (shape) that represents the system to model, in this case, the ”Image to Image Synthesis”. It also shows the participants who will interact with the system, called the external entities. Here Frame is one entity who will interact with the system. In between the process and the external entities, there is data flow (connectors) that indicate the existence of information exchange between the entities and the system.

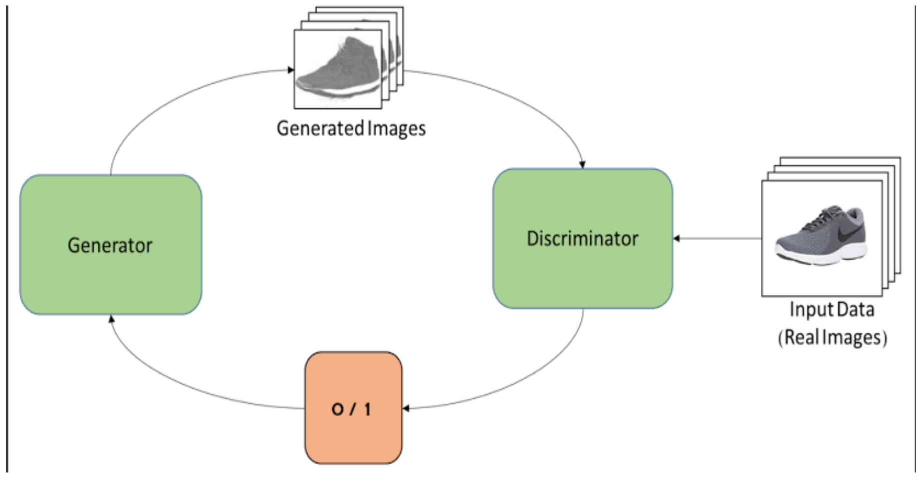

Figure 4.2: Data Flow Level 0 Diagram

The model works by taking an input image from domain DA which is fed to our first generator Generator A→B whose job is to transform a given image from domain DA to an image in target domain DB . This new generated image is then fed to another generator Generator B→A which converts it back into an image, CyclicA , from our original domain DA.

In this level, we build the generator and discriminator used for further levels.

Building the generator : The generator has three components Encoding, Transformation and Decoding. The encoder is used to extract the features from an image. Then in transformation these layers are used for combining different nearby features of an image and then based on these features making decisions about how we would like to transform that feature vector/encoding from DA to that of DB. Then in decoding we do the exact opposite of encoding, we will build back the low level features back from the feature vector.

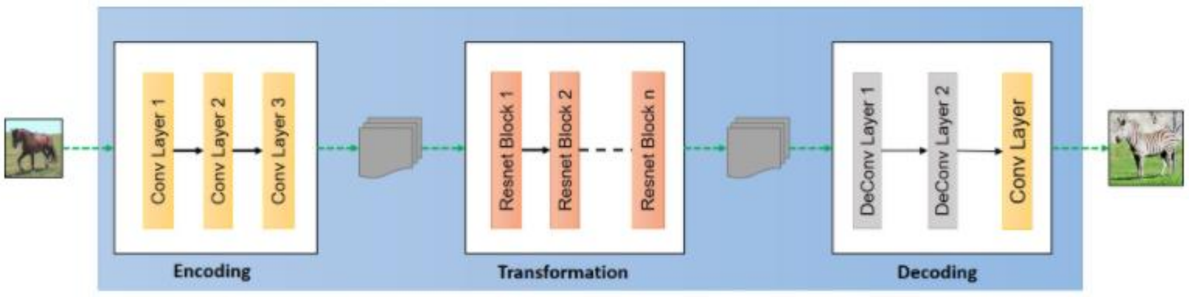

Fig. 4.3(a) Generator

Building the Discriminator : The discriminator would take an image as an input and try to predict if it is an original or the output from the generator, which is a convolution network in our case.

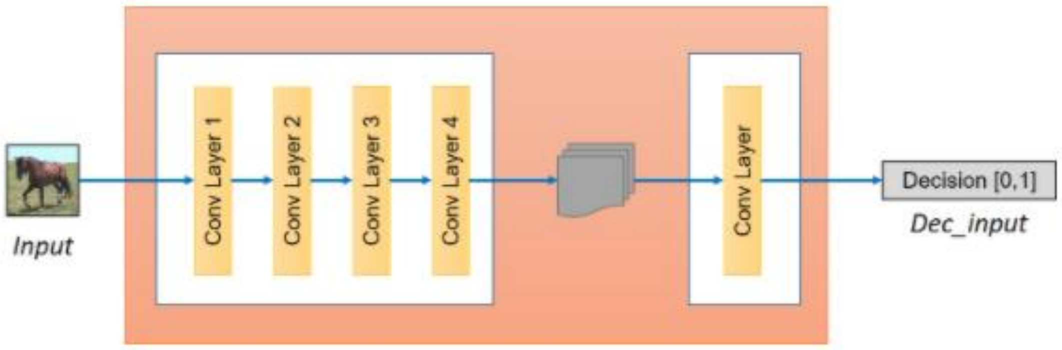

Fig. 4.3(b) Discriminator

Figure 4.3: Data Flow Level 1 Diagram

As you can see in below figure, two inputs are fed into each discriminator(one is original image corresponding to that domain and other is the generated image via a generator) and the job of discriminator is to distinguish between them, so that discriminator is able to defy the generator and reject images generated by it. While the generator would like to make sure that these images get accepted by the discriminator, it will try to generate images which are very close to original images in Class DB . (In fact, the generator and discriminator are actually playing a game whose Nash equilibrium is achieved when the generator's distribution becomes same as the desired distribution)

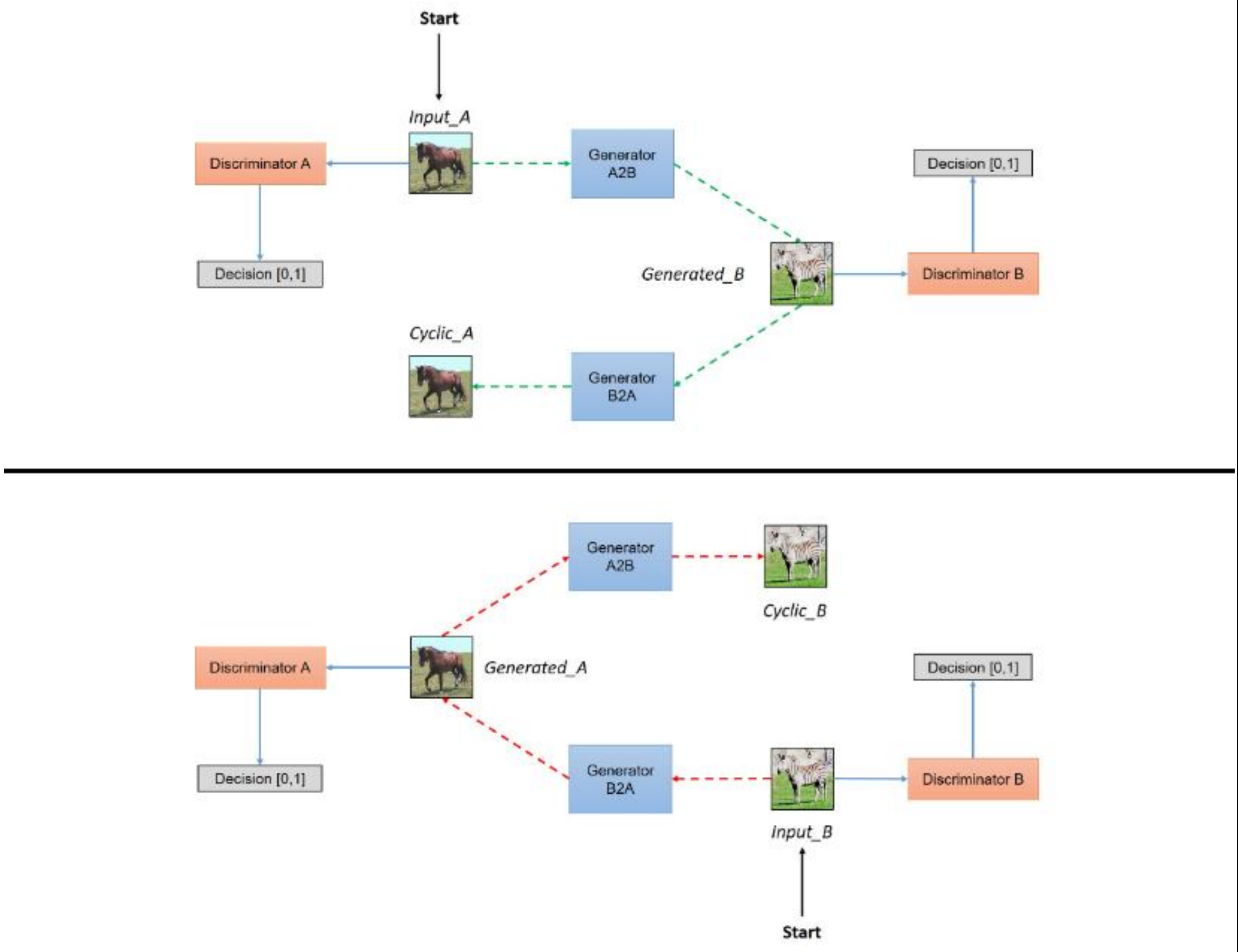

Figure 4.4: Data Flow Level 2 Diagram

4.3 UML Diagrams

4.3.1 Use case Diagram

Following figure shows the use case diagram for image to image synthesis, in which actor role is played by users. Users has assigned a role to select input image and then get the translated image after training is done. On other side we have the admin who has done the model preparation for users. Model preparation includes the preparation of dataset training and testing of dataset after all op- erations the users can see the output according to provided given test cases.

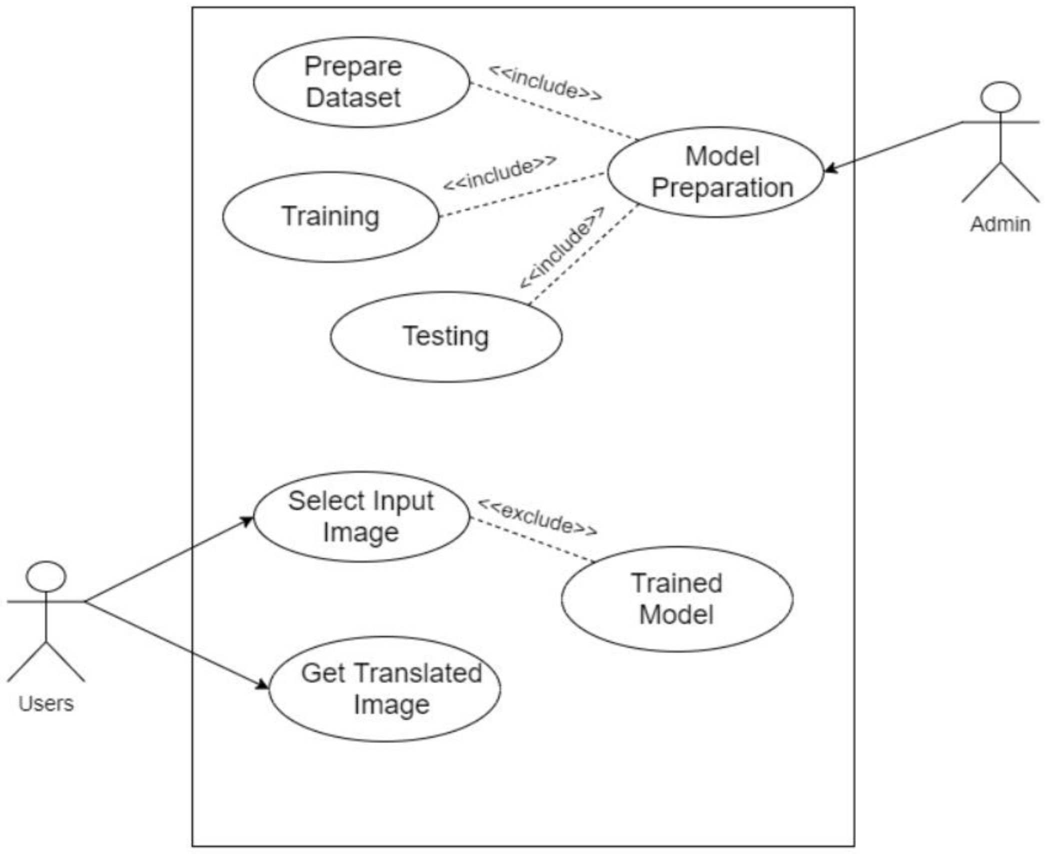

Figure 4.5: Use Case Diagram

4.3.2 Sequence Diagram

Following diagram is describing the sequence of actions taken. First of all, users need to upload images and submit that to the UI Window. The UI Window will take the image as an input and it will pass that to a CycleGan Model. The CycleGan model will initialize the pre-trained model and by applying that model on the input image it will get results. The Translation process will take place and will return the image to UI Window. Users will receive a loaded transformed image on UI Window.

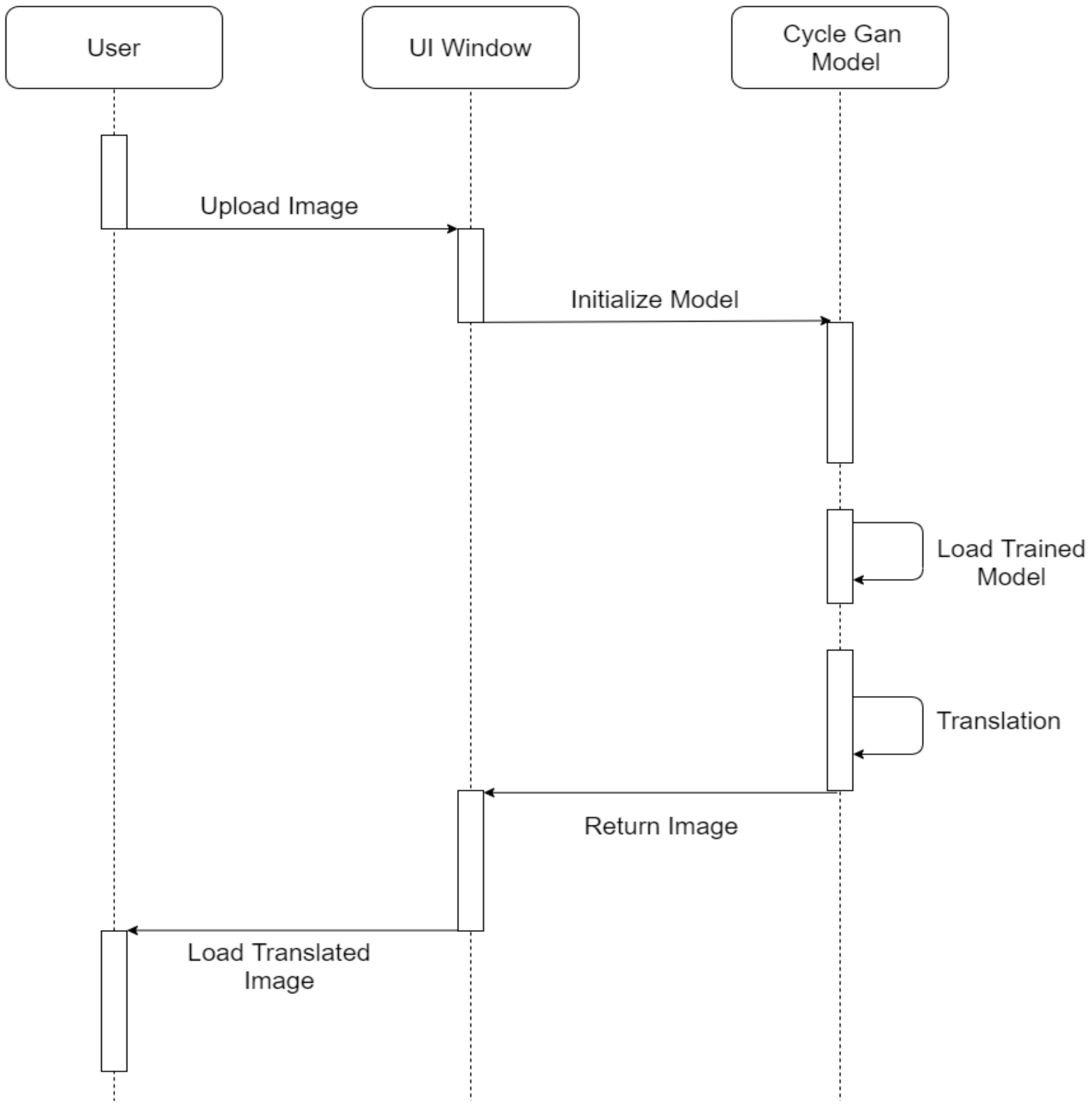

Figure 4.6: Sequence Diagram

4.3.3 Flow Chart

Flowchart of the proposed framework. Without loss of generality, we use gray images in the example. For each input image, we use CycleGAN to synthesize a frontal face. Besides, facial key points are extracted and used to align the facial landmarks. In this way, the result of CycleGAN is refined and an improved result is computed.

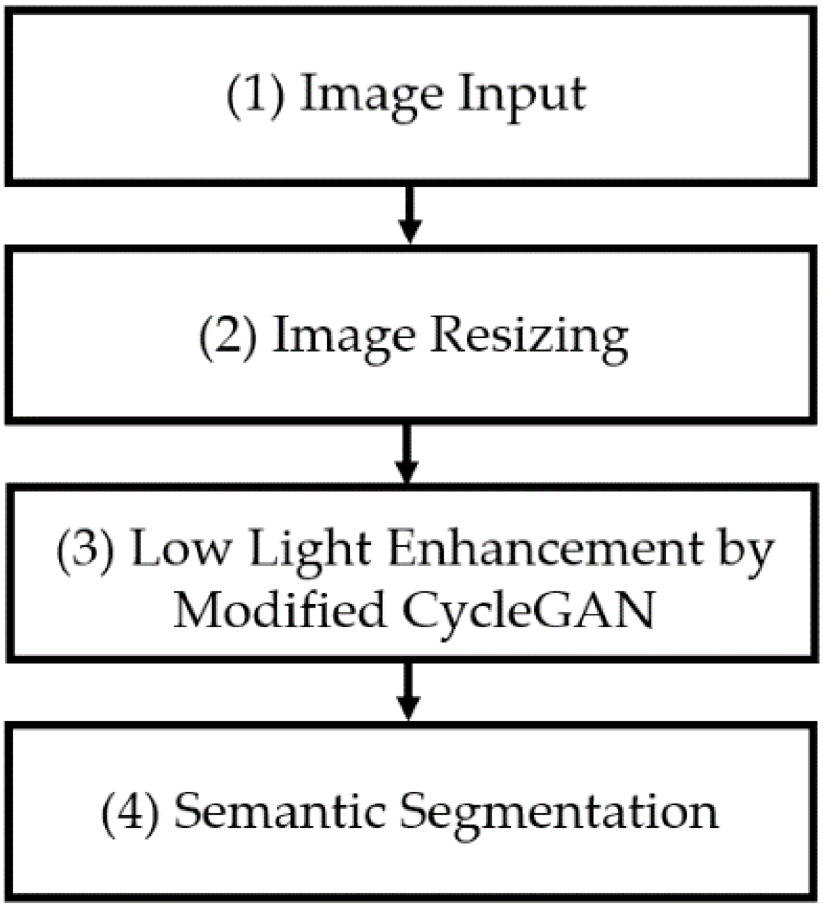
(1) Image Input
(2) Image Resizing
(3) Low Light Enhancement by Modified CycleGAN
(4) Semantic Segmentation

Figure 4.7: Flow Chart

4.3.4 Activity Diagram

Following diagram is describing the activity of components involved in the system. First of all, users need to select an input image and upload that on the UI. The image will be then preprocessed and all important features would be extracted. Then the trained model will be initialized for the given parameters. The image would be then translated to another domain and then returned to the UI. Then the image will be displayed to the user.

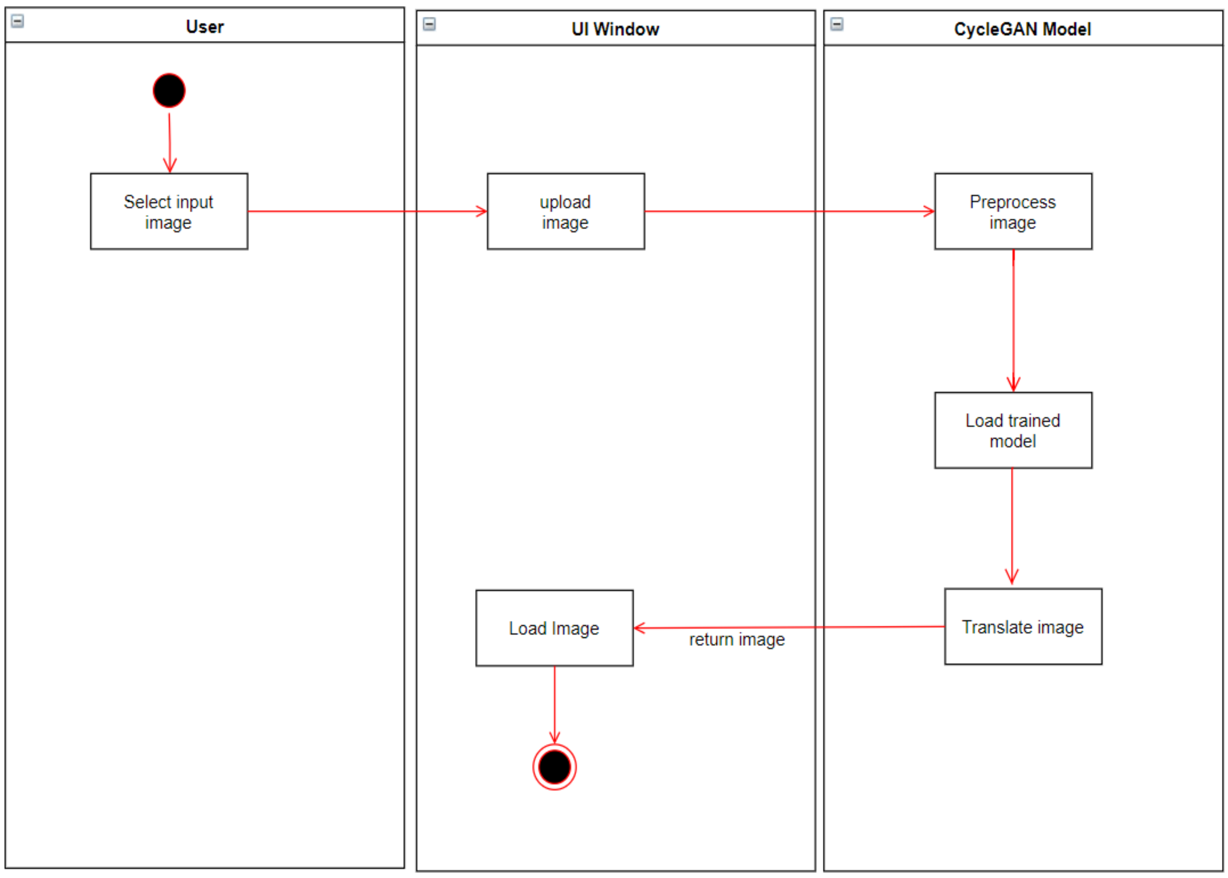

Figure 4.8: Activity Diagram

Chapter 5

Other Specification

5.1 Advantages

5.1.1 Image to image translation

Overall, the results produced by CycleGAN are very good — image quality approaches that of paired image-to-image translation on many tasks. This is impressive, because paired translation tasks are a form of fully supervised learning, and this is not.

5.1.2 Dataset Preparation

There is no need to prepare a huge dataset of paired examples manually.

5.1.3 Accurate results

It works well on tasks that involve color or texture changes, like day-to-night photo translations, or photo-to-painting tasks like collection style transfer.

5.1.4 Photo generation from paintings

For painting→photo, we find that it is helpful to introduce an additional loss to encourage the mapping to preserve color composition between the input and output.

5.2 Limitations

5.2.1 Hardware limitations

For smooth working of this system a system should satisfy minimum requirements stated above,unless this system will give below average performance.It is extremely expensive to train due to complex data models. Moreover deep learning requires expensive GPUs and hundreds of machines. This increases cost to the users.

5.2.2 Less geometric changes

We explored tasks that require geometric changes, with little success.The learned translation degenerates into making minimal changes to the input.This failure might be caused by our generator architectures which are tailored for good performance on the appearance changes. Handling more varied and extreme transformations, especially geometric changes, is an important problem for future work.

5.2.3 Poor results compared to paired method

We observe a lingering gap between the results achievable with paired training data and those achieved by our unpaired method. In some cases, this gap may be very hard – or even impossible – to close. Resolving this ambiguity may require some form of weak semantic supervision.

5.3 Applications

CycleGAN has been demonstrated on a range of applications including season translation, object transfiguration, style transfer, and generating photos from paintings.

- **Painting to Image Translation:** For painting→photo, we find that it is helpful to introduce an additional loss to encourage the mapping to preserve color composition between the input and output.

- **Photograph enhancement** : Photograph enhancement refers to transforms that improve the original image in some way.

- **Object transfiguration :** The model is trained to translate one object class from ImageNet to another (each class contains around 1000 training images).

- **Collection style transfer** : Style transfer refers to the learning of artistic style from one domain, often paintings, and applying the artistic style to another domain, such as photographs.

- **Season transfer** : it refers to the translation of photographs taken in one season, such as summer, to another season, such as winter.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

Image-to-Image translation involves the controlled modification of an image and requires large datasets of paired images that are complex to prepare or sometimes don’t exist. CycleGAN is a technique for training unsupervised image translation models via the GAN architecture using unpaired collections of images from two different domains. In this report, we proposed method for image-to-image transformation called as CSGAN. The CSGAN is based on the Cyclic-Synthesized loss. Ideally, the cycled image should be similar to the synthesized image in a domain. The Cyclic-Synthesized loss finds the error between the synthesized and cycled images in both the domains. By adding the Cyclic-Synthesized loss to the objective function (i.e., other losses such as Adversarial loss and Cycle-consistency loss), the problem of unwanted artifacts is minimized. The performance of proposed CSGAN is validated over two benchmark image-to-image translation datasets and the outcomes are analyzed with the recent state-of-the-art methods. The thorough experimental analysis, confirms that the proposed CSGAN outperforms the state-of-the-art methods. The results in this project suggest that cycle adversarial networks are a promising approach for many image-to-image translation tasks, especially those involving highly structured graphical outputs. These networks learn a loss adapted to the task and data at hand, which makes them applicable in a wide variety of settings.

6.2 Future Scope.

The performance of the proposed method is also either better or comparable over other datasets. In future we want to extend our work towards optimizing the generator and discriminator networks and to focus on unpaired datasets i.e., towards unsupervised learning.The technology used in this project can be further modified and used in various other domain such as Object transfiguration, which can be achieved when model is trained to translate one object class from ImageNet to another. Also Collection Style transfer can be achieved which refers to the learning of artistic style from one domain, often paintings, and applying the artistic style to another domain, such as photographs. It can also be modified for Painting to Image Translation in which we introduce an additional loss to encourage the mapping to preserve color composition between the input and output.

References

[1] “Unpaired Image to Image Translation using Cycle Consistent Adversarial Network” , by Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros

[2] “Image-to-Image Translation with ConditionalAdversarial Networks“, Phillip Isola Jun-Yan Zhu Tinghui Zhou Alexei A. Efros

[3]“Generative-Adversarial Networks” by Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza

[4]"Learning from Simulated and Unsupervised Images through Adversarial Training" Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind

[5] "Improved Techniques for Training GANs" by Tim Salimans, Ian Goodfellow, Wojciech Zaremba

[6] J. Donahue, P. Krahenb ¨ uhl, and T. Darrell. Adversarial ¨ feature learning. arXiv preprint arXiv:1605.09782, 2016. 5

[7] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. arXiv preprint arXiv:1612.05424, 2016. 3

[8] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In ICCV, pages 2650–2658, 2015. 2

[9] L. A. Gatys, M. Bethge, A. Hertzmann, and E. Shechtman. Preserving color in neural artistic style transfer. arXiv preprint arXiv:1606.05897, 2016. 3

[10] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. CVPR, 2016. 3, 6, 8

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, 2014. 2, 3, 4, 5

[12] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science, 313(5786):504–507, 2006. 4

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to image translation with conditional adversarial networks. In CVPR, 2017. 2, 3, 4, 5

[14] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In ECCV, pages 694–711. Springer, 2016. 2, 3, 4

[15] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunning- ´ ham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. arXiv preprint arXiv:1609.04802, 2016.

[16] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. arXiv preprint arXiv:1612.07828, 2016. 3, 4, 5

[17] J. Zhao, M. Mathieu, and Y. LeCun. Energy Based generative adversarial network. arXiv preprint arXiv:1609.03126, 2016. 2

[18] T. Zhou, P. Krahenbuhl, M. Aubry, Q. Huang, and A. A. Efros. Learning dense correspondence via 3d-guided cycle consistency. In CVPR, pages 117–126, 2016. 2, 3

[19] J.-Y. Zhu, P. Krahenb ¨ uhl, E. Shechtman, and A. A. Efros. ¨ Generative visual manipulation on the natural image manifold

[20] L. Karacan, Z. Akata, A. Erdem, and E. Erdem. Learning to generate images of outdoor scenes from attributes and semantic layouts. arXiv preprint arXiv:1612.00215, 2016. 3

Plagiarism Detector

Date: December, 04 2020

PLAGIARISM SCAN REPORT

0%	100%	351	2389
Plagiarised	Unique	Words	Characters

Exclude Url : None

Content Checked For Plagiarism

Image-to-image translation involves creating, with a particular alteration, a new synthetic version of a given image, such as converting a zebra landscape into a horse. Usually, training a model for translation of image-to-image involves a large dataset of paired examples.. These datasets, including images of paintings by long-dead artists, can be difficult and costly to plan and in some cases impossible. CycleGAN is a methodology that requires automated training without paired examples of image-to-image translation models. The models are trained in an unsupervised manner employing a collection of images from the source and target domain that don't have to be related in any way. The CycleGAN extends the GAN architecture. Two generator models and two discriminator models are trained simultaneously in this. Using a training set of aligned image pairs, image-to-image conversion is a class of vision and graphics problems where the objective is to learn the mapping between an input image and an output image.[1] Examples of image-to-image translation include: Translating zebra landscapes to horse landscapes (or the reverse). Translating paintings to photographs (or the reverse). Translating horses to zebras (or the reverse). Traditionally, a dataset consisting of paired examples includes training and an image-to-image translation model. That is, a huge dataset of several input image X examples (e.g. zebra landscapes) and the same image with the desired change that can be used as a predicted output image Y (e.g. horse landscapes). A limitation is the need for a paired training dataset. These datasets, such as images of different scenes under different circumstances, are difficult and costly to prepare. It can be difficult and costly, however to obtain paired training data. It can be even more difficult to obtain input-output pairs for graphics tasks such as artistic stylization, because the desired output is extremely complex, usually requiring artistic authoring. For many tasks, like object transfiguration (e.g., zebra horse), the desired output is not even well-defined.[1]. This is known as the unpaired image-to-image conversion problem. CycleGAN is an effective method of unpaired image-to-image conversion. CycleGAN is an approach using the generative adversarial network, or GAN, model architecture to train image-to-image translation models.

Plagiarism Detector