CARTOON FACE TO HUMAN FACE TRANSLATION USING CONTOUR LOSS BASED CYCLEGAN

A DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

MASTER OF TECHNOLOGY
IN
INFORMATION SYSTEMS

Submitted by:

MAYANK SINGHAL
2K19/ISY/11

Under the supervision of

RITU AGARWAL
ASSISTANT PROFESSOR

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi college of Engineering)
Bawana Road, Delhi-110042

JULY, 2021

CANDIDATE'S DECLARATION

I, Mayank Singhal, 2K19/ISY/11 hereby certify that the work which is presented in the M. Tech Thesis/Dissertation entitled "**CARTOON FACE TO HUMAN FACE TRANSLATION USING CONTOUR LOSS BASED CYCLEGAN**" in fulfilment of the requirement for the award of the Degree of Master of Technology in Information Systems and submitted to the Department of Information Technology, Delhi Technological University, Delhi is an authentic record of my own, carried out during a period from August 2020 to June 2021, under the supervision of **Ms. Ritu Agarwal.**

The matter presented in this report has not been submitted by me for the award of any other degree of this or any other Institute/University.

Place: Delhi — Mayank Singhal

Date: July 29, 2021 — (2K19/ISY/11)

CERTIFICATE

I hereby certify that the M. Tech Thesis/Dissertation titled "**CARTOON FACE TO HUMAN FACE TRANSLATION USING CONTOUR LOSS BASED CYCLEGAN"** which is submitted by **Mayank Singhal**, Roll No. **2K19/ISY/11** Information Systems, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi
Date: July 29, 2021

Ms. Ritu Agarwal
SUPERVISOR
ASSISTANT PROFESSOR
DEPARTMENT OF INFORMATION TECHNOLOGY

ACKNOWLEDGEMENTS

I am very thankful to **Ms. Ritu Agarwal** (Assistant Professor, Department of Information Technology) and **Mr. Kapil Sharma** (Head of Department, Information Technology (M. Tech)) and all the faculty members of the Department of Information Technology at DTU. They all provided me with immense support and guidance for the project.

I would also like to express my gratitude to the University for providing us with the laboratories, infrastructure, testing facilities and environment which allowed us to work without any obstructions.

I would also like to appreciate the support provided to us by our lab assistants, seniors and our peer group who aided us with all the knowledge they had regarding various topics.

ABSTRACT

Cartoon to Human Translation transforms a 2D vector cartoon face to a Real Human Face. The mapping is based on semantic similarity of both the input domains. This is an image→mage translation problem that finds its applications in the entertainment and animation industry. Cartoon movies evolved from 2D animations in 1930 and became more life-like with timeline. In image synthesis, audio, and other sorts of data, Generative Adversarial Networks have demonstrated promising outcomes. They also produce excellent results when translating images to images.

In this research, a CycleGAN based methodology for generating target Human Faces from source Cartoon Faces is proposed, preserving the facial characteristics i.e. face shape, eyebrow alignment and hair style. In order to improve the mapping we have used contour loss along with cycle consistency loss in our model and patch discriminator is used with L2 norm.

CONTENTS

TABLE OF FIGURES

Chapter 1
INTRODUCTION

Cartoons are illustrations which are non-realistic or semi-realistic, can be hand-drawn or computer generated. When these cartoon illustrations are presented sequentially to exhibit an action, it becomes animation. Illustrations are the oldest method of communication, predating languages and writing. Before civilizations turned the sounds into letters, illustrations were already being used to express emotions and to pass on those to others. Sculptures, carvings, monuments, artifacts etc. have been a form of human expression before pages and books were introduced. It will not be wrong to conclude that these forms of expressions are ubiquitous in every era since their origin and have shaped cultures and civilizations. In the present era, it is possible to encode the information in bits and bytes and save it, reproduce it and even generate it digitally. Treating the information as bits and bytes has opened and created several new areas of research such as *Information Theory*[1] , Artificial Intelligence[2], Machine Learning[3] etc

The first digital computer was ABC[4] and was also the basis of the computers we use today. Though images were analogously being produced since the early nineteenth century, it wasn't until 1975 when the first digital image was taken. The work in the field resulted in the production of movies and animations. The first animated movie was "Snow White and the Seven Dwarfs" and "Rendering of the planned highway" was the first vector animation.

With timeline the technology improved and so did the animations and their applications. Their applications are in printed media, advertising, digital media and entertainment industry. In Cartoons and Animations, a living character can be created without the involvement of humans as actors. Such productions are likely to cost less than the productions where humans as actors are involved. In animated movies, the objective is to make realistic human-like characters using cartoons. The proximity to realism is directly proportional to the complexity involved in generating a cartoon. In some applications, there is a need to reproduce the old classical work for new audiences with new technology; reproduction of the movie "Titanic" in 3D, old black & white movies

in color, 2D animations to life-like 3D. However manually manipulating each and every frame of the animations can be very laborious and requires substantial artistic skills. Therefore, special techniques that can transform the frames of animation is helpful and doesn't require an artist's supervision. Here we bring into light one such transformation i.e Cartoon Face to Human Face. The transformations have been done using algorithms dedicated to specific styles and involve a lot of human supervision. Recently, deep learning techniques have been used for the task and have produced remarkable results, specifically the generative models like Generative Adversarial Networks (GANs)[5].

In this work, an effort is made to improve the image generation technique, precisely in bringing realism in the task of generating a Human avatar from a Cartoon avatar. The proposed technique is deep learning[6] based technique that does not require any human input for the generation task.

1.1 MOTIVATION

Cartoons are illustrations which are non-realistic or semi-realistic, can be hand-drawn or computer generated. When these cartoons exhibit motion it becomes animation. Their applications are in printed media, advertising, digital media and entertainment industry. In Cartoons and Animations, a living character can be created without the involvement of humans as actors. Such productions are likely to cost less than the productions where humans as actors are involved. In animated movies, the objective is to make realistic human-like characters using cartoons. The proximity to realism is directly proportional to the complexity involved in generating a cartoon. In some applications, there is a need to reproduce the old classical work for new audiences with new technology; reproduction of the movie "Titanic" in 3D, old black & white movies in colour, 2D animations to life-like 3D. However manually manipulating each and every frame of the animations can be very laborious and requires substantial artistic skills. Therefore, special techniques that can transform the frames of animation is helpful and doesn't require an artist's supervision. Here we bring into light one such transformation i.e. Cartoon Face to Human Face.

1.2 OBJECTIVE

This thesis extends an existing CycleGAN [7] model to a new model that uses contour loss. The original implementation of CycleGAN [7] produces poor results for Cartoon Face to Human Face translation due to substantial difference in input domains (i.e. Cartoon Faces are high level abstractions of face with clear edges and non-natural facial colours and style, the input is a 2D vector image that lacks face texture unlike real human faces). The generator in the original implementation takes an input from the domain A as source and generates an image belonging to the target domain B. In the middle layers of the generator the source image from domain A is compressed and is represented in latent dimension and then that embedding is upsampled to get the target image in domain B. Similarly for the second generator, in the middle layers of the generator the source image from domain B is compressed and is represented in latent dimension and then that embedding is upsampled to get the target image in domain A. In both the cases the enforcement of the embedding of both the domains in latent dimension with respect to similar facial features should preserve the contour based features while translation. This suggestion stood for itself after the experimentation. The original implementation of CycleGAN [7] lacks this consideration.

A contour loss based CycleGAN [7] is proposed in the paper. Our implementation inputs a set of 2D Cartoon vector faces and Human faces for training. The training data was easy to obtain as there was no need of pairing Cartoon Faces and Human Faces, thus reducing our task to unpaired image→image translation [8]. This paper introduces:

(1) A dedicated CycleGAN based approach that learns to translate Cartoon Faces to Real faces using unpaired sets. In comparison to current methods, the model generates Human Faces with more expressive face shapes and hairstyles.

(2) We propose contour loss, a simple and effective loss that enhances the visual similarity between source and target domains irrespective of domain differences without compromising the stability of GAN training.

1.3 ORGANIZATION OF DISSERTATION

Chapter two gives a brief about the topic and how its need is generated also highlighting how useful it could be. It also includes a literature review of tools and technology in Image to Image translation.

Chapter three describes our approach of solving the problem. A novel model has been proposed, its architecture and training is discussed. Dataset and its cleaning to achieve the purpose have been stated.

Chapter four presents the results of the model under different settings and outputs at several epochs. The models output is compared with state of the art technologies.

In Chapter five, thesis is directed towards a conclusion and further ideas for future work have been proposed.

Chapter 2

BACKGROUND AND RELATED WORK

2.1 LITERATURE REVIEW

In image translation task [8], manipulating each frame is tedious and time consuming. The manpower used is directly proportion to the expenses, therefore classical method of manipulating each frame is not an optimized solution. We have tools and techniques for manipulation but the speed of the process is decided by the speed of humans. Humans are smart but not fast, a technique that lies at the horizon is needed. One such technique is Machine Learning. An objective is defined and a mathematical model follows it on classical computers. An expansion of the technique is Deep Learning that uses models that imitate human brain [6].

Machine learning is a sort of data analysis that uses artificial intelligence to create analytical models [3]. It is a field of Artificial Intelligence considering the idea of machines learning from hidden patterns in data and based on those learned patterns take a decision without human brain intervention. Thanks to advancements in computing technology, the machine learning in new era is completely different from machine learning in olden days. It was motivated by pattern recognition [9] and the idea that computing devices may learn to perform tasks without being explicitly taught how to do so.

Machine learning models make several iterations over a dataset called epochs that helps models in evolving independently because in the process they are exposed to new data. They make reliable, repeatable decisions and outcomes by using previous computations. Discriminative and Generative models are two types of machine learning models.

Discriminative Learning [10] is a type of statistical classification model that is commonly employed in supervised machine learning. Generative modelling, also known as conditional models, learns the border between classes or labels in a dataset. It has a proclivity for modelling the combined probability of data points and can use probability estimations and maximum likelihood to produce new examples.

Susceptibility to outliers is less in case of discriminative models, unlike generative models.

Some discriminative models:

- Logistic regression
- Support vector machine
- Decision tree
- Random forests

Generative Machine Learning [11] model is a type of statistical model that is capable of producing new data instances. This model is commonly used to estimate probabilities, model data points, and differentiate between classes using these probabilities. Generic models can handle more complex tasks than discriminative models since they generally rely on Bayes theorem. Unsupervised machine learning uses descriptive modelling to characterise phenomena in data, allowing computers to grasp the real world.

Some Generative models:

- Naive Bayes or Bayesian networks
- Gaussian Mixture Model (GMM)
- Hidden Markov model

2.2 AUTOENCODERS

Autoencoder's [12] output layers has the same dimensions as they have in input. The reason for being so is that the data is copied to output layers from in an unsupervised manner from input layers. Replicator neural network is other name of autoencoder [12].

Each dimension of input to the autoencoder is reconstructed While making the replica, there is a reduction in input size, which results in a representation that is relatively smaller. In middle layers of the network there are lesser number of units than the input and output layers. The output is recreated using this compressed input representation.

2.2.1 ARCHITECTURE OF AUTOENCODERS

An auto-encoder is made up of three parts. An architecture of autoencoder is illustrated in figure 2.1.

Encoder: It is a completely-connected, feedforward neural network that has the ability of input image compression into a latent space representation before encoding it in a lower dimension [12].

Code: It is the output of the auto encoder, which is fed as input to the discriminator.

Decoder: Like the encoder, the decoder is a feedforward network with a structure identical to the encoder. This network is in charge of reassembling the input from the code to its original dimensions [12].

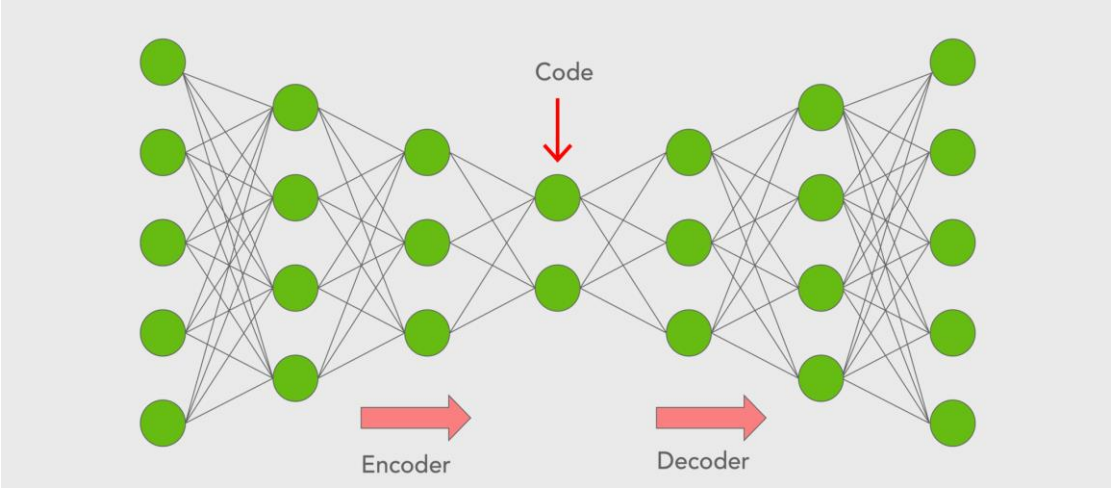

Fig. 2.1 Architecture of AutoEncoder.

The resultant of encoder is Code which is the compressed representation of the input. The original input is then decompressed by the decoder. The autoencoder's principal goal is to provide the output that resembles the input. It's worth noting that the decoder's architecture is the inverse of the encoder's. This isn't a requirement, but it's common practise. The only stipulation is that the input and output dimensions must be identical.

2.2.2 TYPES OF AUTOENCODERS

There are many different types of autoencoders, and a few of them are described briefly below.

Convolutional Autoencoders (CAE): They have the ability of encoding the information as a series of fundamental signals prior to reconstructing the source input from the fundamental signals. CAE can be used to alter the image geometry or produce reflectance. In such class of autoencoder, convolution layers are used as encoder layers, while decoder layers are termed as deconvolution layers. The deconvolution side is referred to as upsampling or transpose convolution.

Variational Autoencoders: Similar to GAN's, such autoencoders can produce new images [13]. Several assumptions are made by variational autoencoder about the distribution of latent variables. The probability distribution of the latent vector of a variational autoencoder generally matches the training data much better than that of a normal autoencoder. VAEs are suitable for any type of art generation since their generation behaviour is far more versatile and configurable than GANs.

Deep Autoencoder: A deep autoencoder [14] constitutes two deep belief networks which are symmetrical. The encoding half of the network is represented by one network, and the decoding half is represented by the other network. They are able to learn more complex features since they have more layers than a standard autoencoder. Restricted Boltzmann machines, make up the layers.

2.2.3 APPLICATIONS OF AUTOENCODERS

- Compression of data
    - Lossy compression: The autoencoder's output is not identical to the input, but it is a near but degraded representation. They are not the best option for lossless compression .
- Denoising an image
- Reduction of Dimensionality
- Extraction of Features
- Image Generation
- Colorization of an image

2.3 GENERATIVE ADVERSARIAL NETWORKS

GANs, or Generative Adversarial Networks [5], are a type of generative modelling that uses convolutional neural networks and other deep learning approaches. Generative modelling falls under the category of unsupervised learning task in machine learning that requires automatically finding regularities and learning patterns in incoming data so that the model may be used to produce or output new instances that could have been chosen from the original dataset. GANs use an amazing way of training a generative model by as a supervised problem using two different neural networks, the generator that produces new images and other one to discriminate as real or fake. These adversaries are trained till discriminator reaches 50% accuracy (in ideal case) suggesting that the generator model is producing credible examples.

GANs, or Generative Adversarial Networks, are generative models based on deep learning. GANs are a model architecture for training a generative model in general, and deep learning models are most commonly used in this architecture. An architecture of GAN is illustrated in figure 2.2.

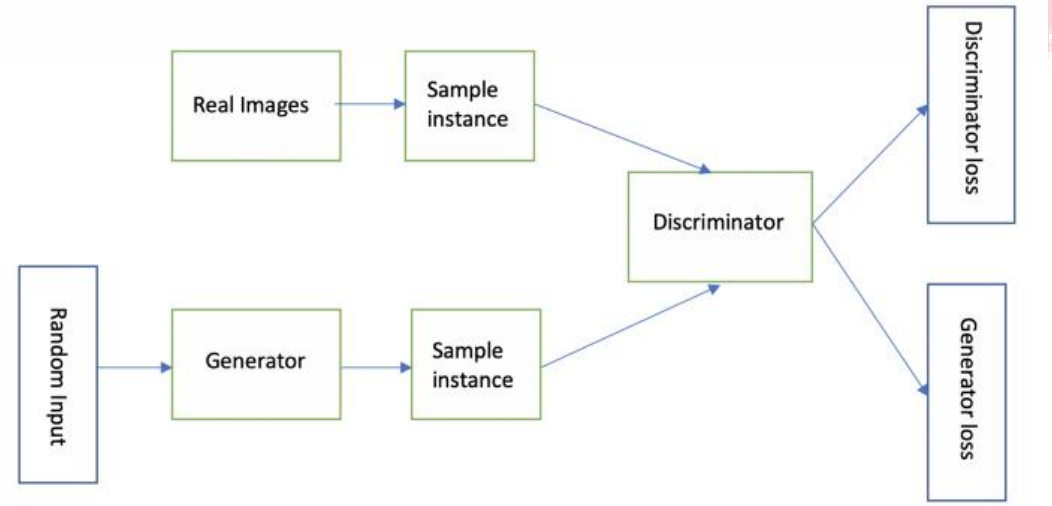

Fig. 2.2 Architecture of Generative Adversarial Network

2.3.1 GAN ARCHITECTURE

The GAN model has a generator that is trained in such a manner that it produces images that are similar to images on which it is trained and a discriminator model whose task is to identify the images produced by generator and the original source [5].

Generator - The model is used to produce fresh credible instances from the domain of the problem.

Discriminator - Model for determining if instances are genuine (from the domain) or not (generated).

2.3.1.1 GENERATOR MODEL

The generator model [5] maps points from lower dimension space, generally random gaussian distribution to a point the desired distribution. While training the points from the random distribution are said to be latent points. The weights of the generator learn to give the latent points a meaning with respect to the source input.

In other words, a latent space enables compression of observed raw data, such as the distribution of input data. The generator model in GANs provides meaning to points in a latent space, allowing fresh points to be pulled from the latent space as input and used to generate new and varied output examples. Figure 2.3 depicts the architecture of a discriminator.

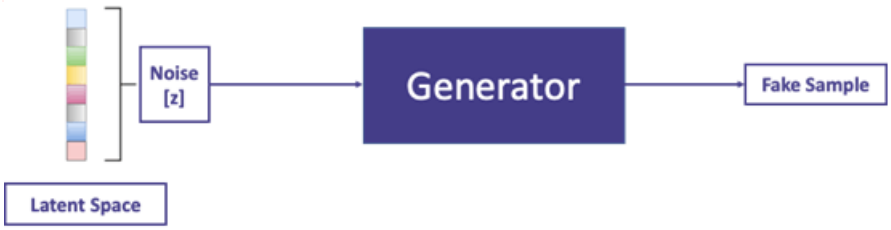

Fig. 2.3 Input and Output of Generator.

2.3.1.2 DISCRIMINATOR MODEL

The discriminator model differentiates between generated and real data points as real or fake[5]. The training data set provided the real-world example. In transfer learning, generator can be useful because during the training procedure, the model learns to extract features from the domain.

Fig. 2.4 Input and Output of Discriminator.

2.3.2 GAN AS A TWO PLAYER GAME

Although generative modelling is an unsupervised learning problem, the GAN design brilliantly frames the generative model's training as a supervised learning problem [5]. The discriminator and generator models are both trained at the same time. The generator generates a batch of samples, which are subsequently sent to the discriminator, together with genuine domain occurrences, to be classified as true or false. As a result, the two models are in competition, are antagonistic in the sense of game theory, and are playing a zero-sum game.

The discriminator parameters are changed or unchanged when it efficiently distinguishes between actual and false samples, whereas the generator is penalised with significant model parameter adjustments. The generator is rewarded or the model parameters are not modified when it deceives the discriminator, whereas the discriminator is penalised and its model parameters are updated.

The generator is able to produce immaculate copies from the input domain after a certain point, and the discriminator fails to differentiate, so it always predicts "uncertain" (e.g., 50 percent for real and fake). This is only an idealised scenario, and

it is not necessary to reach this step in order to arrive at a feasible generator model. In figure 2.5, you can see a complete GAN model.

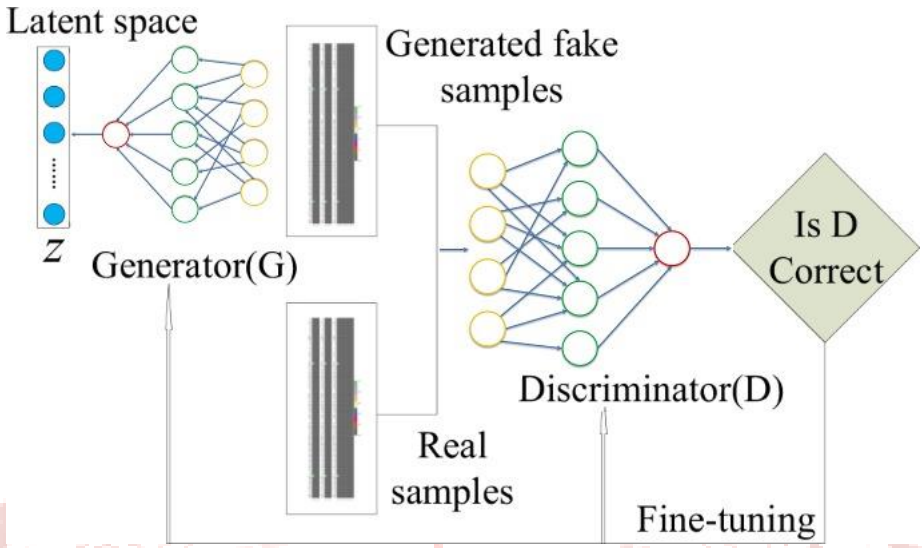

Fig. 2.5 A complete model of GAN.

Convolutional Neural Networks [16], are employed as the generator and discriminator models in GANs that usually deal with picture data in DCGANS or Convolutuional GANs. When the generator's input, the latent space, is used to model picture data, it produces a compressed representation of the collection of images or photographs that were used to train the model

2.3.3 OBJECTIVE FUNCTION

A GAN is made up of two parts: (1) a generator and (2) a discriminator. The discriminator D_ϕ is a function that distinguishes samples from the real dataset and the generator. G_θ is a directed latent variable model that deterministically generates samples x from z. G_θ and D_ϕ are represented graphically in the image below. x stands for samples (from data or a generator), z stands for our noise vector, and y stands for the discriminator's prediction about x.

Both the generator and the discriminator play a two-player minimax game in which the generator minimises a two-sample test objective ($p_{data} = p_\theta$) while the discriminator maximises it ($p_{data} \neq p_\theta$). Intuitively, the generator seeks to deceive the discriminator as much as possible by generating samples that seem identical to p_{data}.

The GAN objective can be phrased as follows:

$$\min_{\theta} \max_{\emptyset} V(G_\theta, D_\emptyset) = \mathrm{E}_{x\sim p_{data}}[\log D_\emptyset\,(x)] + \mathrm{E}_{z\sim p_{(z)}}[\log(1 - D_\emptyset\,(G_\theta(z)] \quad (1)$$

We know that the discriminator maximises this function with respect to its parameters ϕ, and that it performs binary classification with a fixed generator G_θ: it assigns probability 1 to data points from the training set $x\sim p_{data}$, and probability 0 to generated samples $x\sim p_G$. The best discriminator in this situation is:

$$D^*_G(x) = \frac{p_{data}(x)}{p_{data}(x) + p_G(x)} \quad (2)$$

The generator, on the other hand, minimises this goal for a fixed discriminator D_ϕ. And inserting in the optimal discriminator $D^*_G(.)$ into the overall goal $V(G_\theta, D^*_G(x))$ reduces to:

$$2D_{\mathrm{JSD}}[p_{data}, p_G] - log4 \quad (3)$$

The Jenson-Shannon Divergence, commonly known as the symmetric form of the KL divergence, is referred to as the D_{JSD}:

$$D_{\mathrm{JSD}}[p, q] = \frac{1}{2}\left(D_{KL}\left[p, + \frac{p+q}{2}\right] + D_{KL}\left[q, + \frac{p+q}{2}\right]\right) \quad (4)$$

The JSD has the advantage of $D_{JSD}[p,q]=D_{JSD}[q,p]$, which means it meets all of the KL's qualities. The ideal generator for the GAN goal becomes $p_G=p_{data}$ with this distance metric, and the best objective value we can get with optimal generators and discriminators $G^*(.)$ and $D^*_{G^*}(x)$ is -log4.

2.3.4 GAN TRAINING ALGORITHM

The model is trained as follows:

Do the following for epochs 1,...,N:

1. Sample minibatch of size m from data: $x^{(1)},\ldots,x^{(m)}\sim D$
2. Sample minibatch of size m of noise: $z^{(1)},\ldots,z^{(m)}\sim p_z$
3. On the generator parameters θ , take a gradient descent step.

$$\nabla_\theta V(G_\theta, D_\emptyset) = \frac{1}{m}\nabla_\theta \sum_{i=1}^{m} log(1 - D_\emptyset\,(G_\theta(z^{(i)})) \quad (5)$$

4. Using the discriminator parameters ϕ , take a gradient ascent step:

$$\nabla_\emptyset V(G_\theta, D_\emptyset) = \frac{1}{m}\nabla_\emptyset \sum_{i=1}^{m} \left[log\ D_\emptyset\left(x^{(i)}\right) + log\left(1 - D_\emptyset\left(G_\theta\left(z^{(i)}\right)\right)\right)\right] \quad (6)$$

2.4 RESNETS

ResNet, stands for Residual Network, are also a type of neural network [18]. To address a complex problem, we expect deeper layers to solve the problem. But, deeper networks face the problem of vanishing gradients and learning becomes stagnant after few epochs. This problem is solved using residual networks that use skip connections which make possible training of deeper models without the problem of vanishing gradients. When it comes to image identification, the first layer may learn to recognise edges, the next one to recognise textures, the next to recognise objects, and so on. The conventional Convolutional neural network model, on the other hand, has been discovered to have a maximum depth threshold. The error % on training and testing data for a 20 layer Network and a 56 layer Network is shown in figure 2.6.

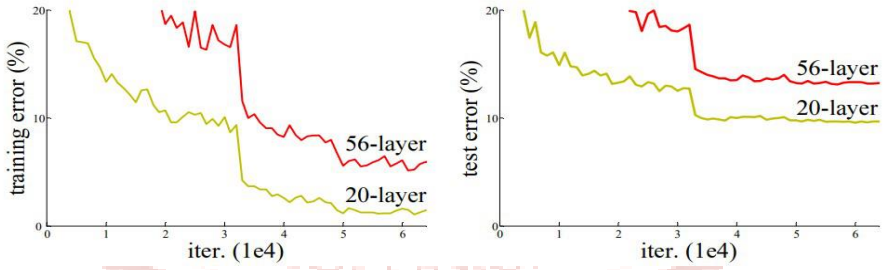

Fig. 2.6 Training and Test error with respect to iteration.

In both training and testing data, we can see that a 56-layer network has a greater error percent than a 20-layer network. This means that when more layers are placed on top of a network, its performance degrades. The reasons may be several ranging from optimization objective to design of the network to vanishing gradients. The 56-layer network has the worst error percent on both training and testing data, which doesn't happen when the model is overfitted [19].

2.4.1 RESNET BLOCK

The emergence of ResNet or residual networks [18] , which are made up of Residual Blocks as shown in figure. 2.7, has relieved the challenge of training very deep networks.

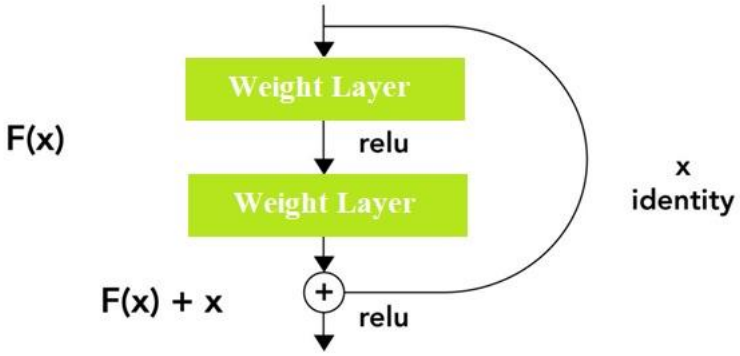

Fig. 2.7 A resnet block with skip connection.

The first thing we notice is that there exists a direct connection that bypasses several levels (which may vary according to the model) in between. The core of residual blocks is a link known as ‘the skip connection . The output of the layer is no longer the same due to this skip connection. Without this skip link, the input 'x' is multiplied by the layer's weights, then a bias term is added.

ResNet's skip connections solve the problem of vanishing gradient in deep neural networks by allowing the gradient to flow through a different channel. These connections also aid the model by allowing it to learn the identity functions, ensuring that the higher layer performs at least as well as the lower layer, if not better.

Assuming there is a shallow network and a deep network that use the function H to map an input 'x' to an output 'y' (x). We want the deep network to perform equivalent to the shallow network, without compromising with the performance. One method to achieve this is for additional layers in a deep network to learn the identity function, and so their output equals inputs. An example of shallow and deep network is shown in figure 2.8.

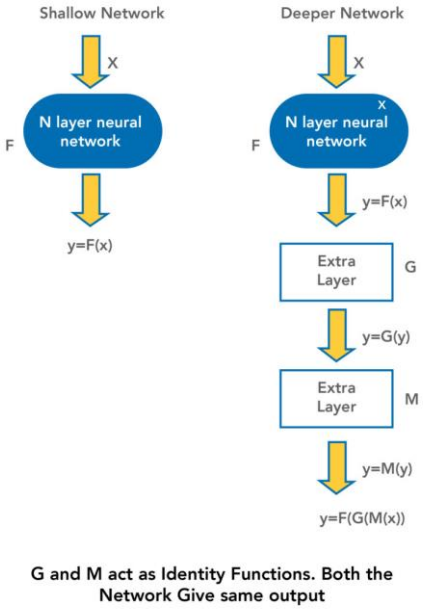

Fig. 2.8 Shallow and Deeper Network.

2.4.2 RESNET ARCHITECTURE

Comparing ResNets to simple layered neural networks, the performance of neural networks with extra layers has been considerably enhanced by utilising ResNet [18], as illustrated in figure 2.9 of error percent.

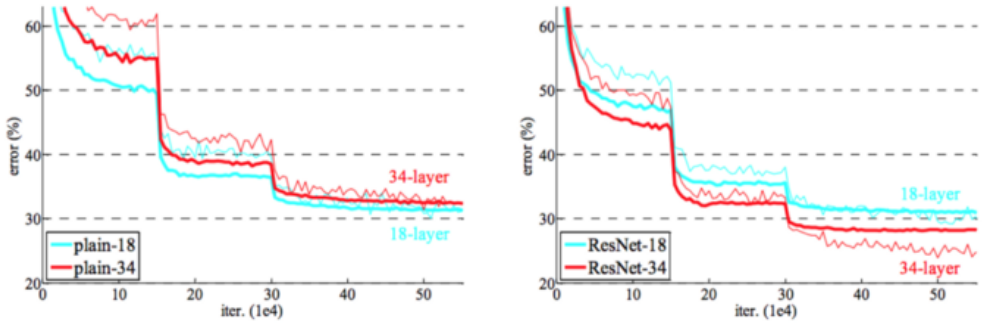

Fig. 2.9 Comparison of ResNet and Plain Networks with deep layers.

The difference is noticeable in 34-layer networks, with ResNet-34 having a significantly lower error % than plain-34. It's also worth noting that the percentage of error for plain-18 and ResNet-18 are almost identical.

The ResNet network uses a 34-layered simple network architecture [20] influenced by VGG-19, after which the skip connection is implemented. These skip links, as shown in figure 2.10, transform the design into the residual network.

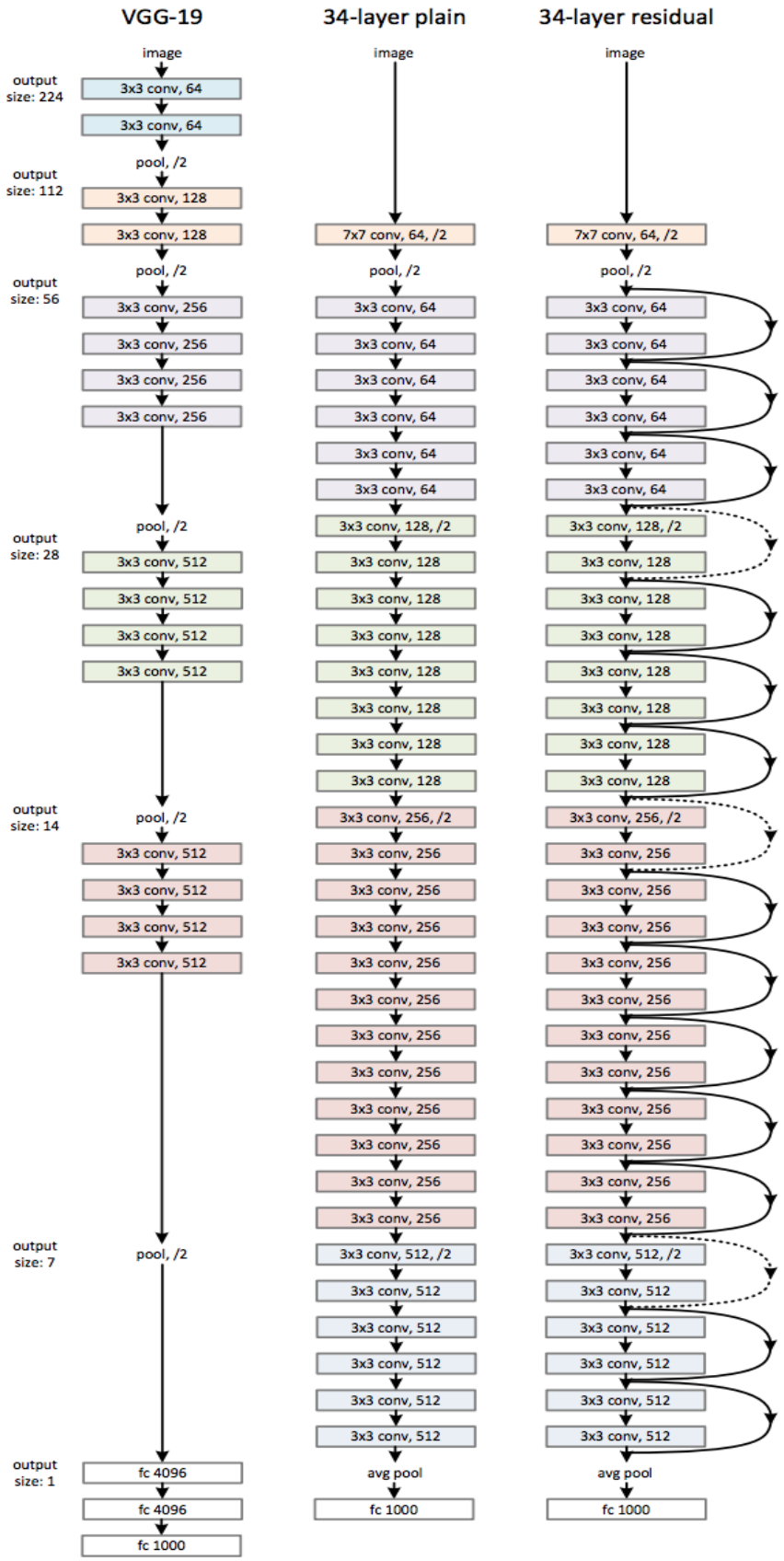

Fig. 2.10 ResNet Architecture

2.5 CYCLE GAN

CycleGAN [7] is a technique that makes possible image translation with non-paired source and target images. The training is done in an unsupervised fashion with a set of photos from the source and target domains that are not related to each other. This novel technique is relatively effective in yielding visually stunning results in a variety of domains, including converting horse photos to zebra and vice versa. Images translation entails the controlled change of an image and necessitates the preparation of vast datasets of paired images, which can be difficult or impossible to find. Season translation, item transfiguration, style transfer, and photo generation from paintings are just a few of the tasks CycleGAN [7] has been used for.

2.5.1 IMAGE TO IMAGE TRANSLATION

Image→image translation [8] is an image synthesis problem that necessitates the creation of a new image from a controlled alteration of an existing one.

The following are some applications of image→translation:

1. Taking summer sceneries and turning them into winter landscapes (or the reverse).
2. Creating photographs from paintings.
3. Horses → zebras translation.

Traditionally, a dataset of matched instances is required to train an image→image translation model. That is, a huge dataset containing several instances of input photos X and the same image with the intended alteration that can be utilised as an expected output image Y. Preparing these datasets, such as images of different scenes under varied situations, is difficult and expensive. In many cases, such as iconic artworks and their accompanying pictures, the datasets do not exist. As a result, strategies for training an image→image translation system that do not require paired samples are in high demand. Any two unconnected image collections can be used with the general features collected from each collection and employed in the image translation process. This is termed as the unpaired image-to-image translation problem.

2.5.2 UNPAIRED IMAGE TO IMAGE TRANSLATION

CycleGAN [7] uses the generative adversarial network model architecture to train image→image translation models.

The GAN architecture is a method of training two image synthesis models. It's the task of discriminator to decide whether the input image is real or fake (based on a dataset), The generator takes random point as input and generates new images from the domain In a game, both models are updated so that the generator can fool the discriminator and can better recognise produced images [7].

The first generator uses photos from the first domain to create images for the second, while the second generator uses photos from the second domain to make images for the first. The discriminators are then used to update the generator models in order to determine how realistic the generated images are. This extension may be adequate for producing realistic images in each domain, but it is insufficient for producing translations of the input images.

Cycle consistency is a further enhancement to the CycleGAN [7] architecture. This is the idea of feeding a first-generation image into a second-generation generator, with the second-output generation's matching the first-generation image. The converse is also true: a second generator output can be used as an input to the first generator, with the result matching the second generator's input.

Cycle consistency is supported by the CycleGAN [7] by adding an additional loss to quantify the distance between the generated output and the original picture, and vice versa. This normalises the generator models, driving the image production process in the new domain toward image translation.

2.5.3 CycleGAN ARCHITECTURE

Inspired from GAN, CycleGAN consists of Generator and Discriminator. Unlike GAN, CycleGAN consists of a pair of Generator and Discriminator. Identity loss ensures if the generator takes as input the source domain, it doesn't make changes to it. Cycle consistency loss ensures that translation of images from source to domain can be reverted back.

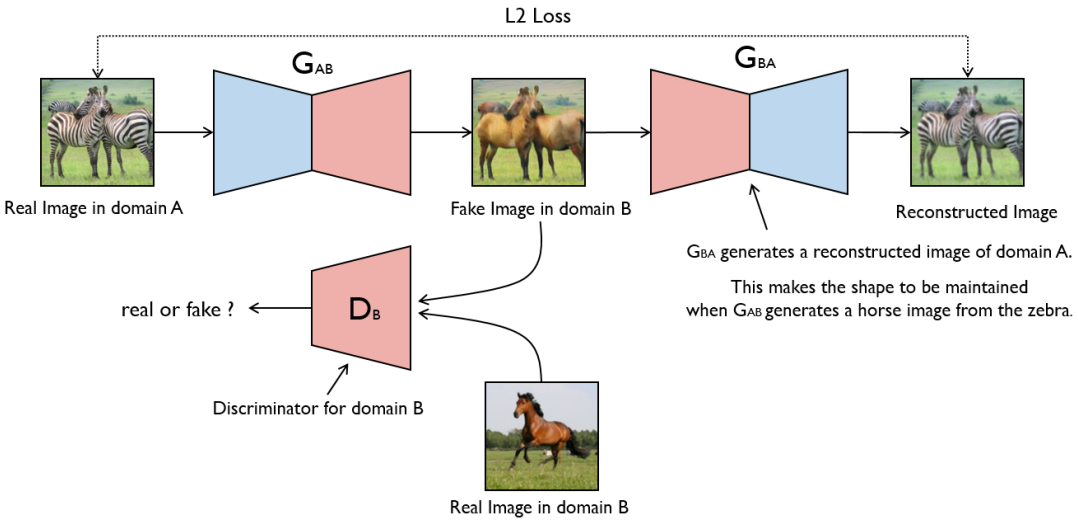

Fig. 2.11 CycleGAN architecture.

2.5.4 APPLICATION OF CycleGAN

Style Transfer: Style transfer [21] is the process of taking an artistic style from one domain, such as paintings, and transferring it to another, such as photos. The CycleGAN [7] is demonstrated by photographing landscapes in the artistic styles of Monet, Van Gogh, Cezanne, and Ukiyo-e.

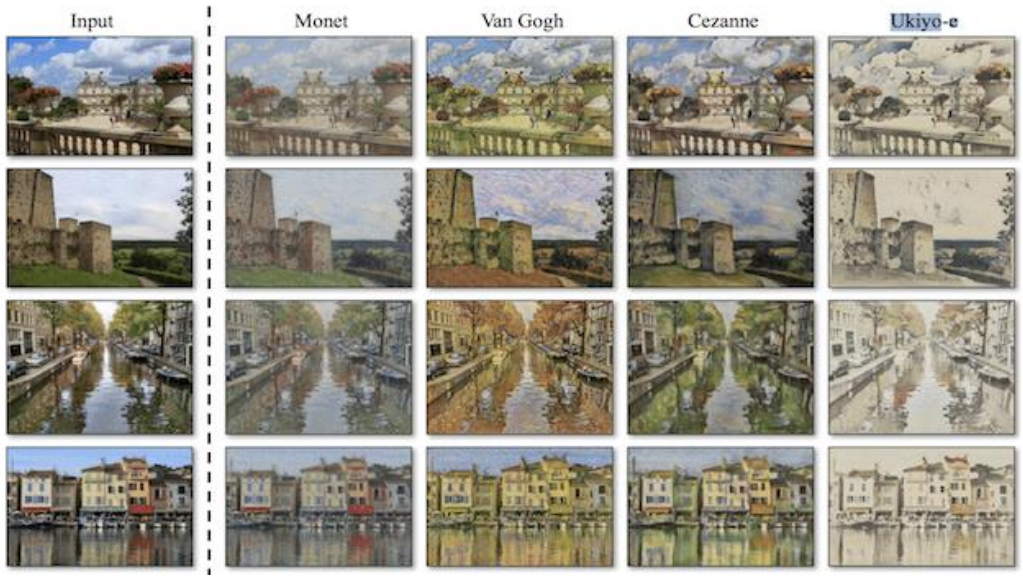

Fig. 2.12 Translation of Real scenes into paintings by Monet, Van Gogh, Cezanne, Ukiyo-e (Left to Right)

Object Transfiguration

The transition of objects from one class, such as dogs, into another class, such as cats, is known as object transfiguration. The CycleGAN [7] is seen converting pictures of horses into zebras and the other way around. Given that both horses and zebras are identical in size and structure, except for their colouring, this type of transformation makes sense. An example of translation is shown figure 2.13.

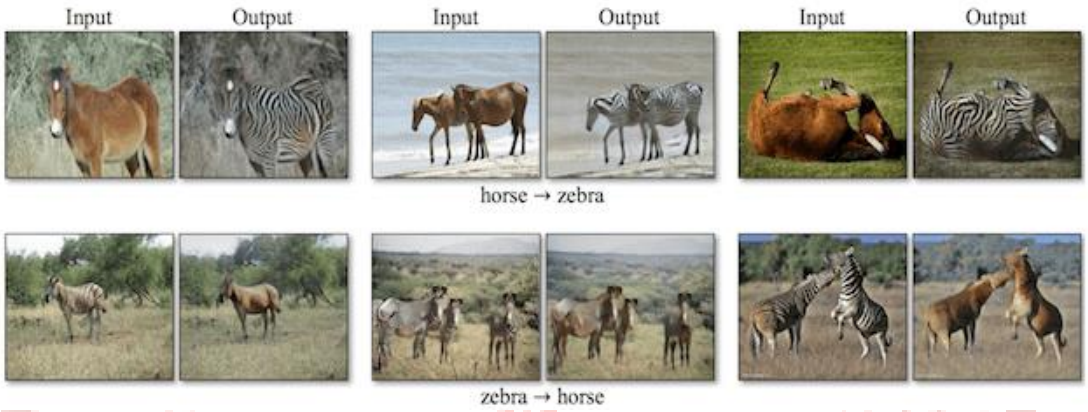

Fig. 2.13 Translation of Horse to Zebra and Vice Versa.

The CycleGAN is also demonstrated on translating photographs of apples to oranges, as well as the reverse: photographs of oranges to apples.Again, this transfiguration makes sense as both oranges and apples have the same structure and size. An example of such translation is shown in figure. 2.14.

Fig. 2.14 Translation of Apple to Oranges and Vice Versa.

Chapter 3
PROPOSED APPROACH

3.1 METHODOLOGY

We solve the problem using a CycleGAN [7] based model that uses Contour loss. Let the cartoon images represent the domain A and real human faces represent the domain B. Although both domains are faces and have features in common, they visually appear non-similar. Our task of transformation of images from domain A to domain B should be based on some similarity rather than random mapping. Semantics being the same, the model is developed to encourage the transformation based on semantics like hairstyle, facial shape, eyebrows shape and face gesture. We utilised a model that leverages cycle consistency loss [7] because our technique is unpaired image to image translation and an additional contour loss is added that improves the translation and produces better results. The model performs semantic style transfer [22] from domain A to domain B. Similar to cycleGAN [7], our model consists of a pair of generators namely G & F and discriminators DA and DB.Unlike to cycleGAN [7] G & F are encoder-decoder models.

$$G(x) = d_1(e_1(x)) \text{ And } F(y) = d_2(e_2(y)) \quad (14)$$

The training objective for the proposed model can be decomposed into adversarial loss, identity loss, cycle consistency loss and contour loss.

- Adversarial Loss: This is a state-of-the-art GAN objective [5] that makes the transformed distribution p(d1(e1(A))) as close as possible to the target distribution p(B). The generator (d1(e1(A))) is paired against the discriminator DB and DB strives to differentiate between generated samples and real samples, from domain B. Generally, binary cross-entropy is used but for the purpose of image→image translation mse performs better. Adversarial loss is described in eq. 15 and eq. 16.

$$L_{GAN}(e_1, d_1, D_B) = E_{b\sim p_{data}(B)}[(D_B(b) - 1)^2] + E_{a\sim p_{data}(A)}[\left(D_B\left(d_1(e_1(a))\right)\right)^2] \quad (15)$$

$$L_{GAN}(e_2, d_2, D_A)$$
$$= E_{a\sim p_{data}(B)}[(D_A(a) - 1)^2]$$
$$+ E_{b\sim p_{data}(B)}[\left(D_A\left(d_2(e_2(b))\right)\right)^2] \quad (16)$$

Collectively, Adversarial loss can be written as:

$$L_{GAN}(e_1, d_1, e_2, d_2, D_A, D_B) = L_{GAN}(e_1, d_1, D_B) + L_{GAN}(e_2, d_2, D_A) \quad (17)$$

- Identity Loss: If the generator receives samples from the target domain as input, the mode should prevent any changes to the input. Identity loss is utilised to encourage this.

$$L_{identify}(e_1, d_1, e_2, d_2) = E_{b\sim p_{data}(B)}[\,||\, d_1(e_1(a)) - a \,||_1\,] +$$
$$E_{b\sim p_{data}(B)}[\,||\, d_2(e_2(b)) - b \,||_1\,] \quad (18)$$

- Cycle Consistency Loss: It encourages the synthesized image in the target domain that are to be translations of the input image.Cycle consistency loss offers transitivity in supervised CNN training, i.e., for an image a in domain A, the image translation cycle should be able to return a to the original image.

$$a \rightarrow G(a) \rightarrow F(G(a)) \sim a$$

This is a forward cycle. Similarly we have a backward cycle.

$$b \rightarrow F(b) \rightarrow G(F(b)) \sim b$$

Collectively, Cycle consistency loss is:

$$L_{cyc}(e_1, d_1, e_2, d_2) = E_{a\sim p_{data}(A)}[\,||\, d_2(e_2(d_1(e_1(a)))) - a \,||_1\,] +$$
$$E_{b\sim p_{data}(B)}[\,||\, d_1(e_1(d_2(e_2(b)))) - b \,||_1\,] \quad (19)$$

- Contour Loss: It encourages the inputs from both the domains to be represented in lower dimensions based on similarity of face shape, hairstyle, and other characteristics. The domain A input is encoded in lower dimension and the embedding is passed on to domain B decoder. The loss is calculated and gradients are updated keeping decoderB untrainable. Symmetrically, embeddings for domain B are preserved.

$$L_{contour}(e_1, d_1, e_2, d_2) = E_{a\sim p_{data}(A)}[\,||\, d_2(e_1(a)) - a\,||_1\,] + E_{b\sim p_{data}(B)}[\,||\, d_1(e_2(b)) - b\,||_1\,] \quad (20)$$

The weighted total of the above-mentioned losses is the entire loss function: -

$$L(e_1, d_1, e_2, d_2, D_{B,} D_{A,}) = L_{GAN}(e_1, d_1, e_2, d_2, D_A, D_{B,}) + L_{identify}(e_1, d_1, e_2, d_2) + L_{cyc}(e_1, d_1, e_2, d_2) + L_{contour}(e_1, d_1, e_2, d_2) \quad (21)$$

3.2 ARCHITECTURE

An encoder plus a decoder make up the Generator. The encoder is a sequence of downsample convolution blocks [23] to encode the image input followed by ResNet convolutional blocks [20] for the image transformation. Batch normalization is replaced with Batch-Instance Normalization [24], this allows removal of instance-specific contrast from the content image hence simplifying the generation and improving resultant images. The input images of dimensions 128x128 were encoded into an embedding of dimension 25x25x1024.

The decoder network is a sequence of upsampling convolutional blocks and also uses Instance Normalization [24]. It upsamples the embedding and produces an image of source dimensions i.e. 100x100.

Discriminator used here is patch Discriminator [25]. It takes an input image and predicts whether it is fake or real. A patch discriminator discriminates an image based on average on nxn patches of the source image. The parameter n is the size of the patch. PatchGAN takes the effective receptive field into consideration, where single output activation maps to an nxn patch of the image. In experiments, we tested for different patches and obtained impressive results with 17*17 patch discriminator. A smaller patch acts as a local discriminator for the whole image. The architecture of proposed model is illustrated in figure 3.1.

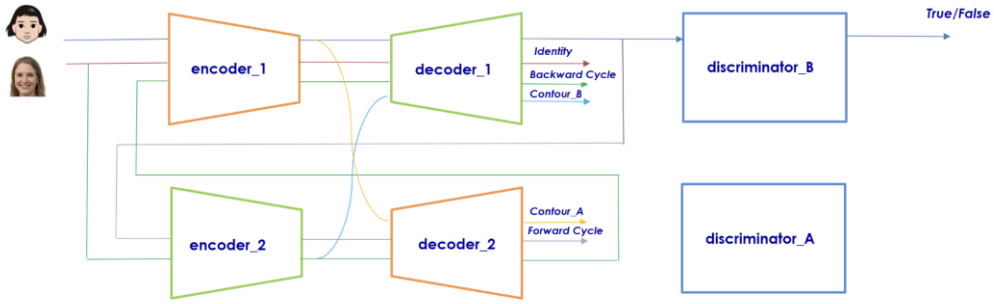

Fig. 3.1 Illustration of the proposed method

3.3 DATASETS

CartoonSet [26] is a dataset that has been released publicly for the research in the domain. For Cartoon faces we have used this dataset. The dataset is a collection of vector cartoon faces. A face in the dataset is parameterised by 16 attributes i.e 12 facial attributes and 4 color attributes. The number of choices per attribute is in the range between 3 and 111, resulting in 250 cartoon components and 100 million possible combinations. figure 3.2 illustrates samples from the dataset.

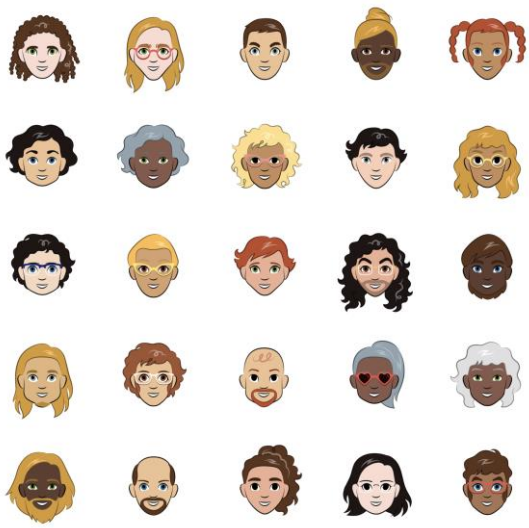

Fig. 3.2 CartoonSet10K

All the features don't correlate to human facial features and filtering has been done to remove unnatural skin and hair colours. Faces with glasses have been ignored to learn mapping of eyes and eyebrows. Since cartoon images have sharp edges and human faces don’t, Gaussian Filter with radius value 2 has been applied to smoothen the

edges[27] . Of 10,000 images from CartoonSet, 1792 images with natural skin and features have been selected. Of 1792 images 1200 images were selected for training. figure 3.3 illustrates samples after filtering.

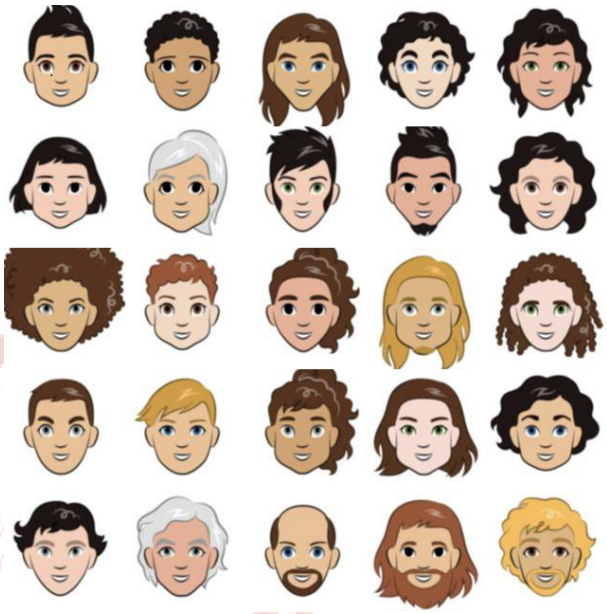

Fig. 3.3 CartoonSet10K after filtering.

Human Face Dataset, Face Mask Lite Dataset has been used for the purpose. The dataset consists of 10000 human faces with mask on and another 10000 images without the mask. For the purpose, without masks images were used. This dataset is generated by StyleGAN 2 [28]. StyleGAN2 generates state-of-the-art results in unconditional data driven Generative modelling as shown in figure 3.4.

Fig. 3.4 Human Face Dataset by StyleGAN2

Filtering was done by removing side views and images with extra features like cap and glasses. To reduce the noise, matting of the images has been done. Images after matting are shown in figure 3.5.

Fig. 3.5 Human Face dataset after Matting.

Chapter 4

RESULTS AND COMPARISONS

The proposed model is an image to image translator that translates cartoon faces to human faces based on facial features. The source domain is CartoonSet, of which 1792 images are selected with natural face and hair tints, and target domain is Human Face dataset. From selected images, 1200 are used for training. All the images in the Human Face dataset are generated by styleGAN2 [28]. As a preprocessing step, the images from the cartoon dataset are filtered, blurred to smoothen the edges and zoomed to reduce the noise [27]. The images from the Human Face dataset are filtered, and matted to remove the background noise. Equal number of training images from the target domain are used. For the purpose of training, we resized input images to 128x128 dimensions. In figure 4.1, along the rows we see the output of different models that perform the task of Cartoon face to Human Face translation

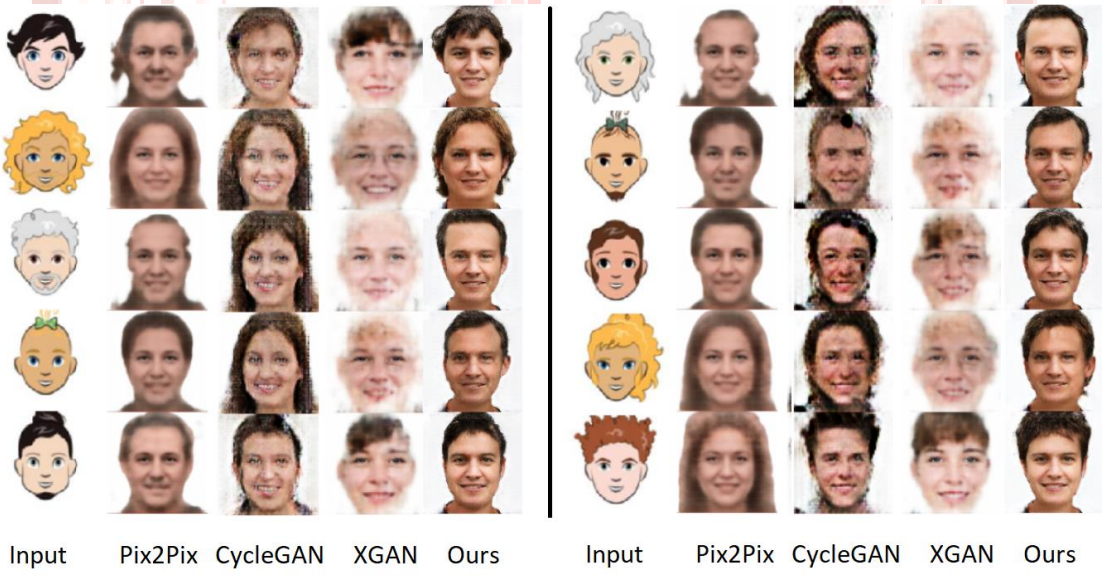

Fig. 4.1 Comparison of Faces generated by Pix2Pix, CycleGAN, XGAN and Proposed Model.

In this section we compare the output of the proposed model with other models. Although all the models are able to generate plausible human faces, we observe that faces generated by Pix2Pix [8] are nowhere similar to the input. The generated faces are diverse in face shapes and hairstyles but output seems to be more of a random mapping than based on semantics. Moreover the faces are not crisp and have gradual smoothing along the face edges and hair texture is not close to the natural texture.

CycleGAN [7] performs a better job in semantic mapping than Pix2Pix GAN [8], but the model suffers mode collapse as can be seen from third and fourth rows in figure 18. The output is almost similar, input being different. Moreover the output is noisy and lacks photo-realistic details.

XGAN [29] produces blurry images and poorly performs in generating the correct hairstyle. Though diversity in skin colour and hair colours can be seen. Facial features are diverse when facial expressions are taken into consideration. Still, the images don't seem to be photo-realistic when compared to the output of the proposed model.

The proposed model generates crisp images with diversities in hairstyle, facial expressions. The images are realistic when compared to the baselines models. The skin tones and hair texture are natural and the model also learns the mapping of face shapes from cartoon to real domain.

Various models were run and tested to produce the best quality images. From stae-of-the art U-NET generator to different orders of patch discriminators. Patch discriminators are different from traditional discriminators in sense that they don’t predict for complete image as real or fake, instead different patches of the images are classified as real or fake and average of all the patches are taken. In figure 4.2 output of the model with 142 patch discriminator is shown at different epochs. The images generated after 500 epochs are not crisp but are able to capture the semantic properties of the input. The outputs after 1000 epochs are crisp and capture wide range of features. Since the model is stable at 1000^{th} epoch further epochs also shown an improvement in the output.

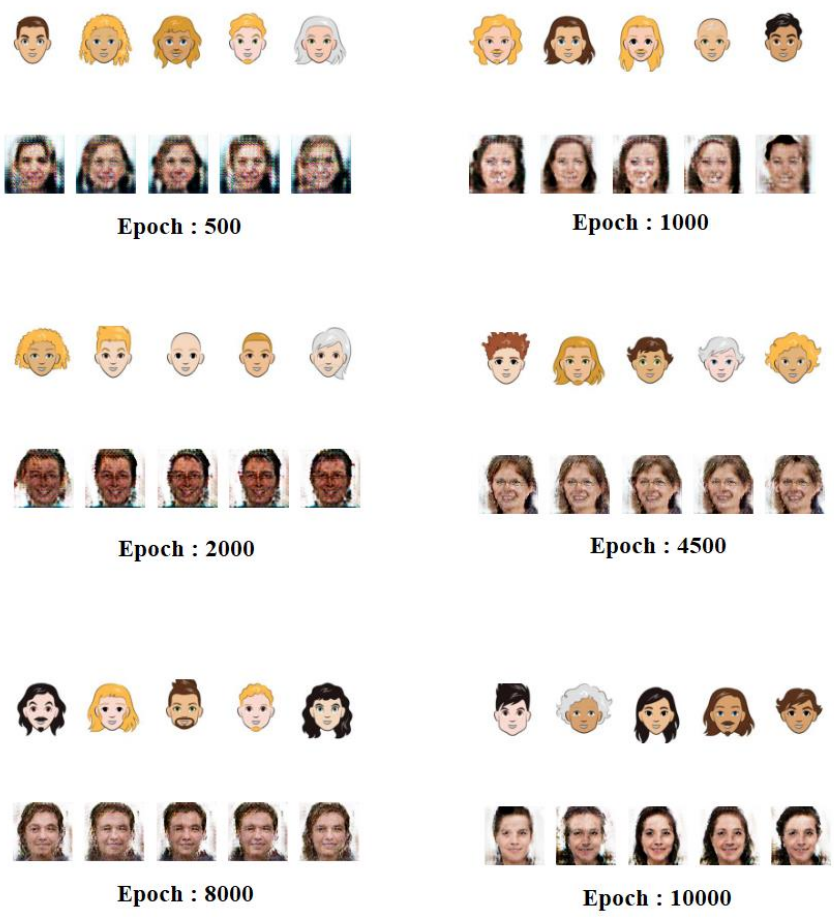

Fig. 4.2 Output with 142 patch Discriminator.

In figure 4.3 output of the model with 70 patch discriminator is shown at different epochs. The images generated after 500 epochs are not crisp but are able to capture the semantic properties of the input. The outputs after 5500 epochs are crisp and capture wide range of features. Since the model is stable at 9000th epoch further epochs also shown an improvement in the output. As compared to 142 patch discriminator the outputs of 70 patch discriminator are better.

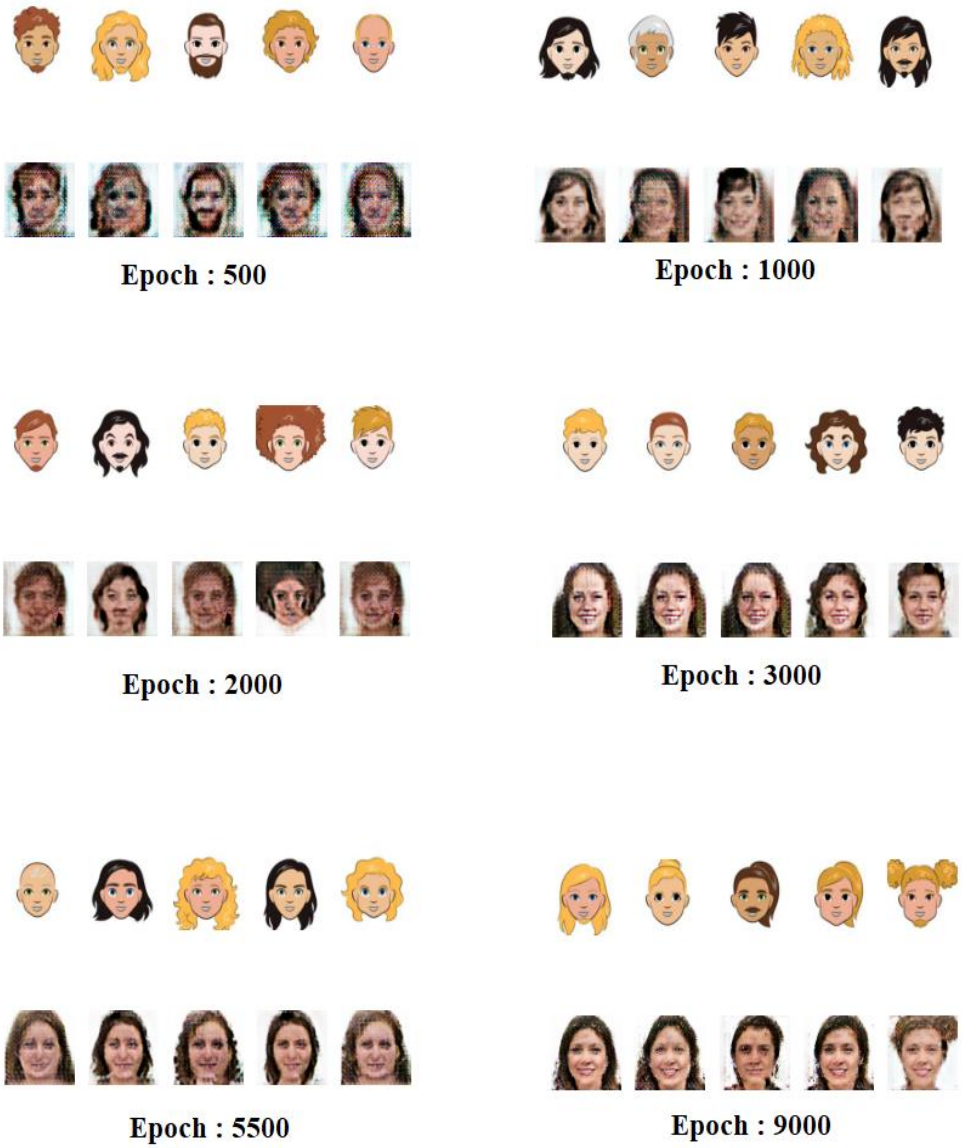

Fig. 4.3 Output with 70 patch Discriminator.

In figure 4.4 output of the model with 70 patch discriminator is shown at different epochs. The images dimensions for this model were 128*128, since it has larger number of features, the model was slow to train. The image quality improves with larger dimensions since model has lot more features to capture.

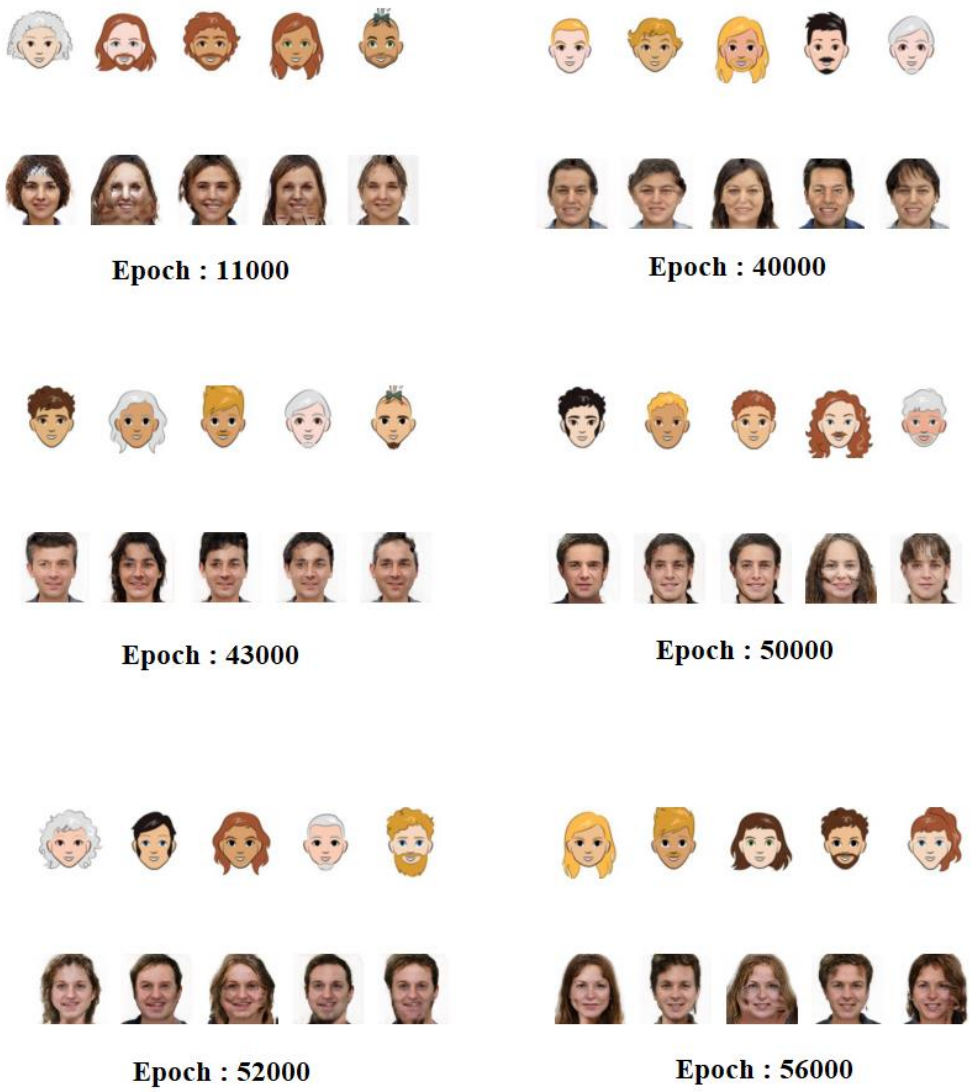

Fig. 4.4 128*128 dimension output by proposed model with 70 patch discriminator.

U-NET is a symmetric encoder decoder structure with skip connections and has shown significant results in semantic classification and segmentation. The model was opted as a replacement for encoder – decoder architecture in cycleGAN but the results were not very impressive. An architecture with resnet blocks is better able to encode images in lower dimension.

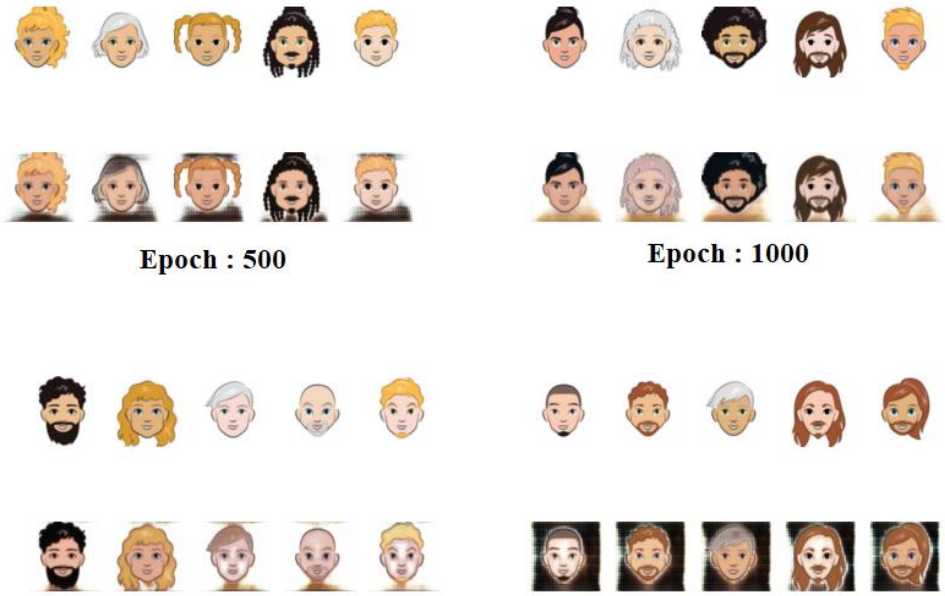

Fig. 4.5 Output by UNETmodel as generator.

Chapter 5
CONCLUSION

We have proposed contour loss based cycleGAN, a model for image→image translation based on semantics. The model performs better when compared with other cartoons to real translation techniques from the literature.. The translation is more feature oriented than random mapping. The facial features are crisp and more precisely preserved during translation. The contour loss, significantly improves the mapping of images, without destabilizing the GAN training process, produces better results than XGAN, pix2pix GAN and cycleGAN.

The model captures semantics like hairstyle, face-shape. The translation of cartoon faces to human faces is a translation if the features of the original source are transformed in the target domain. In our model the features on which the transformation takes place are hairstyles, eyebrow alignment, face shape, smile and eyes position. However the model struggles to capture properties like hair-colour and facial tint. The reason is the weakly supervised nature of the training process based on unpaired image to image translation. The semantics are spatial features and the distance we are using to measure the difference between source and target image is the Euclidian norm, so weightage given to spatial features makes sense. The model can be trained with an additional objective to capture other properties such as facial colour and hair colours.

For future work, investigation of semantics based translation on other domains apart from facial data would be interesting. Apart from image translations, the task of frame translations of an entire video to produce a target video can be explored. Conversely, cartoonization of real photos using the model can be qualitatively analysed.

REFERENCES

[1] Carter, T., 2007. An introduction to information theory and entropy. *Complex systems summer school, Santa Fe* (2007).

[2] Russell, S. and Norvig, P., 2002. Artificial intelligence: a modern approach.

[3] S. Chang, T. Cohen, and B. Ostdiek, "What is the machine learning?," *Springer*, vol. 97, no. 5, Mar. 2018, doi: 10.1103/PhysRevD.97.056009.

[4] G. O'Regan, "The First Digital Computers," 2016, pp. 55–72.

[5] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. *Advances in neural information processing systems*, *27*.

[6] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4. Now Publishers Inc, pp. 197–387, 2013, doi: 10.1561/2000000039.

[7] J.-Y. Zhu, T. Park, P. Isola, A. A. Efros, and B. A. Research, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks Monet Photos."

[8] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, and B. A. Research, "Image-to-Image Translation with Conditional Adversarial Networks."

[9] K. Fukunaga, "Introduction to Statistical Pattern Recognition," 2013.

[10] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative learning for differing training and test distributions," in *ACM International Conference Proceeding Series*, 2007, vol. 227, pp. 81–88, doi: 10.1145/1273496.1273507.

[11] M. Holmqvist, L. Gustavsson, and A. Wernberg, "Generative learning: Learning beyond the learning situation," *Educ. Action Res.*, vol. 15, no. 2, pp. 181–208, Jun. 2007, doi: 10.1080/09650790701314684.

[12] W. H. Lopez Pinaya, S. Vieira, R. Garcia-Dias, and A. Mechelli, "Autoencoders," in *Machine Learning: Methods and Applications to Brain Disorders*, Elsevier, 2019, pp. 193–208.

[13] D. P. Kingma and M. Welling, “An introduction to variational autoencoders,” *Foundations and Trends in Machine Learning*, vol. 12, no. 4. Now Publishers Inc, pp. 307–392, 2019, doi: 10.1561/2200000056.

[14] D. Chicco, P. Sadowski, and P. Baldi, “Deep Autoencoder Neural Networks for Gene Ontology Annotation Predictions,” *dl.acm.org*, pp. 533–540, Sep. 2014, doi: 10.1145/2649387.2649442.

[15] T. Ghosh and M. Kirby, “Supervised Dimensionality Reduction and Visualization using Centroid-encoder,” Feb. 2020.

[16] P. Kim, “Convolutional Neural Network,” in *MATLAB Deep Learning*, Berkeley, CA: Apress, 2017, pp. 121–147.

[17] F. Gao, Y. Yang, J. Wang, J. Sun, E. Yang, and H. Zhou, “A Deep Convolutional Generative Adversarial Networks (DCGANs)-Based Semi-Supervised Method for Object Recognition in Synthetic Aperture Radar (SAR) Images,” *mdpi.com*, 2018, doi: 10.3390/rs10060846.

[18] D. Wu, Y. Wang, S.-T. Xia, J. Bailey, and X. Ma, “SKIP CONNECTIONS MATTER: ON THE TRANSFER-ABILITY OF ADVERSARIAL EXAMPLES GENERATED WITH RESNETS.”

[19] D. M. Hawkins, “The Problem of Overfitting,” *ACS Publ.*, vol. 44, no. 1, pp. 1–12, Jan. 2004, doi: 10.1021/ci0342472.

[20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition.”

[21] Y. Li, N. Wang, J. Liu, and X. Hou, “Demystifying neural style transfer,” in *IJCAI International Joint Conference on Artificial Intelligence*, 2017, vol. 0, pp. 2230–2236, doi: 10.24963/ijcai.2017/310.

[22] A. J. Champandard, “Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artworks,” Mar. 2016.

[23] Radford, A., Metz, L. and Chintala, S., 2015. Unsupervised representation

learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

[24] H. Nam and H. E. Kim, "Batch-instance normalization for adaptively style-invariant neural networks," in *Advances in Neural Information Processing Systems*, 2018, vol. 2018-December, pp. 2558–2567.

[25] R. Xu, Z. Zhou, W. Zhang, and Y. Yu, "Face Transfer with Generative Adversarial Network," Oct. 2017.

[26] "Cartoon Set: An Image Dataset of Random Cartoons." [Online]. Available: https://google.github.io/cartoonset/download.html. [Accessed: 02-Jul-2021].

[27] Y. Chen, Y.-K. Lai, and Y.-J. Liu, "CartoonGAN: Generative Adversarial Networks for Photo Cartoonization."

[28] J. Lehtinen and T. Aila NVIDIA, "Analyzing and Improving the Image Quality of StyleGAN Tero Karras NVIDIA Samuli Laine NVIDIA Miika Aittala NVIDIA Janne Hellsten NVIDIA."

[29] A. Royer *et al.*, "XGAN: Unsupervised Image-to-Image Translation for Many-to-Many Mappings."