Deep Learning and its Applications Project Presentation on Coloring Black And White movies Using Style Transfer Group-16

Ashutosh Jamadari(B16014), Naveen Kumar Chouhan(B16022),
Shubham Choudhary(B16035)

April 17, 2019

1. Problem Statement
2. Motivation and Challenges
3. Data sets
4. Proposed Methodology
5. Methodology explored
6. Results and their discussion
7. Conclusion and future work
8. Work Done

Problem Statement

Neural style transfer is one of the most fascinating and interesting application of Deep Learning. With the help of this neural style transfer we are going to color black and white videos.
Let's say we have a very old Bollywood movie in black and white and we have a latest Bollywood coloured movie. Now using this colored movie we are going to color our old black and white movie and obtain a coloured version of that old movie.

Motivation and Challenges

Nowadays many people want to enjoy old movies and games, but because of their old color contrast and shady black and white color people avoid watching them. This might be the reason behind all this remakes coming nowadays. Using neural style transfer we can provide new life to all this old movies. Also it can provide a new look to many modern games and videos and can boast today's entertainment industry.

Neural style transfer seems to be very attractive and fascinating but implementing this network can be bit challenging in various aspects.

1) Style transfer is used to redefine images and small clips with few objects involved at a time in each frame but the most challenging part of this project is to apply style transfer to a movie with lots of objects involved at a time in each frame.

2) Other problems involved are whether all the objects are getting appropriate color, our network is able to distinguish between variety of objects, etc.

3) Problems related to network may occur at the time of implementation.

Dealing with all this problems will be challenging part for us.

Data sets

Figure: A pic from Mughal-e-Azam

This black and white image from Mughal-e-Azam is transformed into a coloured image using style transfer.

Proposed Methodology

The steps or let's say methodology in which are thinking of proceeding are as follows:
1) Dividing the input video and the style video into numerous frames.
2) Learning neural network f:X $\rightarrow$ $Yi.e.Generator(X2Y)$,
which will transform the frames data sets from X domain (Black White) to Y domain (colored).
3) Learning another network f:Y $\rightarrow$ $Xi.e.Generator(Y2X)$,
which will transform the frames data sets from Y domain (colored) to X domain (Black White), which is the exact reverse of the above network.
4) Learning a network called Discriminator X, which will classify whether the frame belongs to domain X or it is transformed into domain X from another domain.
5) Learning another network called Discriminator Y having similar functioning like of Discriminator X, this network will classify whether the frame belongs to domain Y or it is transformed into domain Y from another domain.

6) Computing all the losses obtained from Discriminator X, Discriminator Y, Generator X2Y and Generator Y2X and training all the networks simultaneously and training the model with appropriate data set.

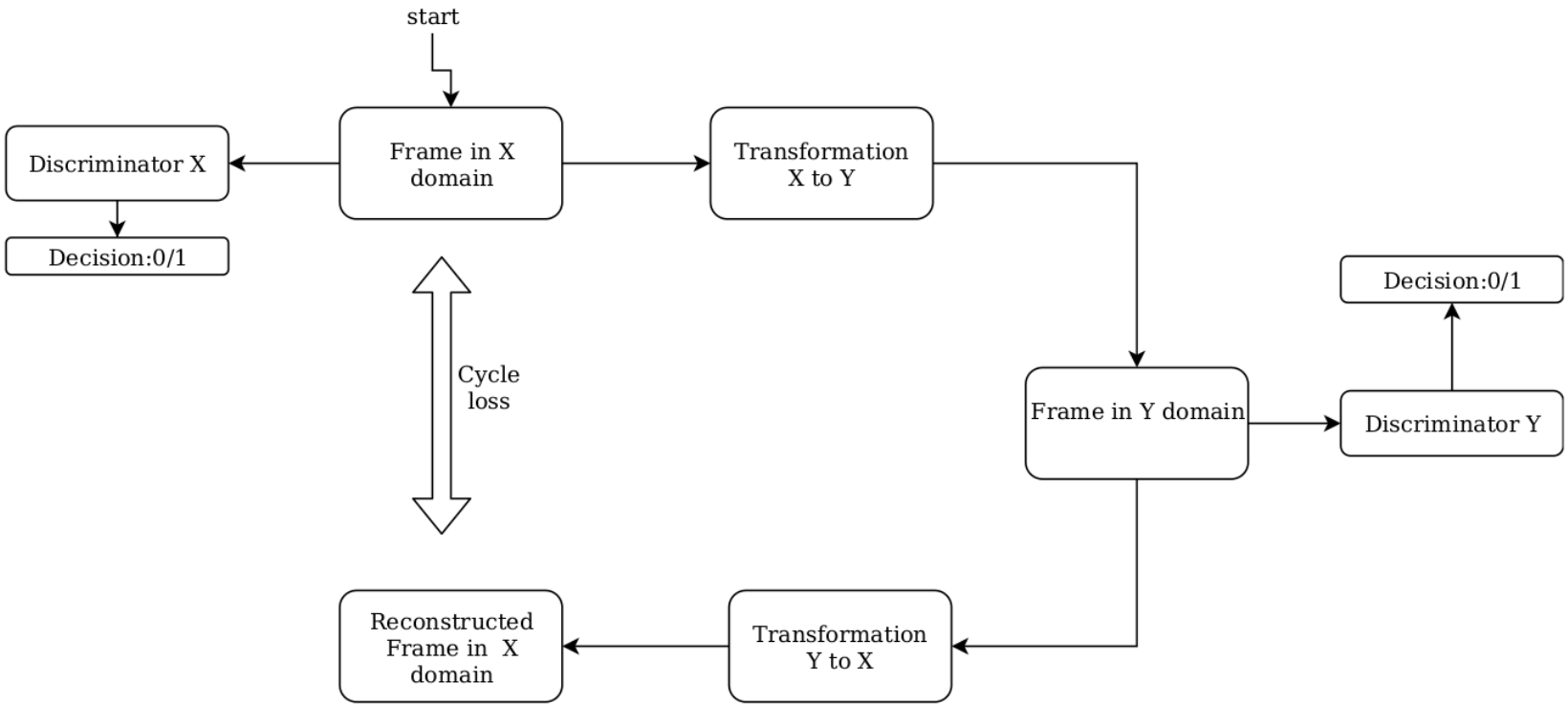

Figure: Block diagram of network when input is from Domain X

Methodology explored

Architecture of Generators:
The Generator have three components:
a) Encoder: It is a CNN network consisting 3 convolutional layers.
b) Transformer: It is a residual network consisting 6 resnet blocks.
c) Decoder: It is a deconvolution network consisting 3 deconv layers.
Architecture of Discriminators:
The Discriminator is a simple classifier consisting several convolutional layers.
Training all the networks needs all the losses to be computed. And the losses obtained from all the networks are as follows:

1) Loss of Discriminator X
D_X_loss = L2(dec_X, 1) + L2(dec_gen_X, 0)
dec_X = score of discriminator when frame is from X domain.
dec_gen_X = score of discriminator when frame is transformed into X domain.
2) Loss of Discriminator y
D_Y_loss = L2(dec_Y, 1) + L2(dec_gen_Y, 0)
dec_Y = score of discriminator when frame is from Y domain.
dec_gen_Y = score of discriminator when frame is transformed into Y domain.
3) Loss of Generator X2Y
g_loss_X2Y = L2(dec_gen_Y, 1) + Lambda * cyc_loss.
4) Loss of Generator Y2X
g_loss_Y2X = L2(dec_gen_X, 1) + Lambda * cyc_loss.
Where,
cyc_loss = L2(input_X, cyc_X) + L2(input_Y, cyc_Y)
cyc_X = Frame generated when input_X (frame of X domain) is transformed into Y domain and then again transformed into X domain.
cyc_Y = Reverse of cyc_X .

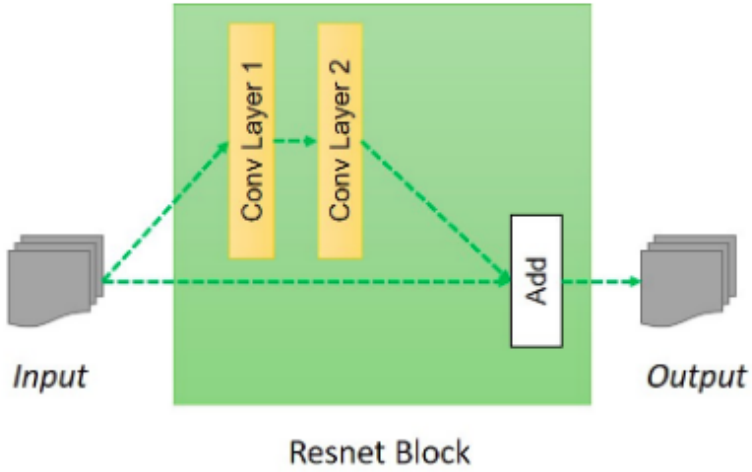

Figure: Resnet Architecture

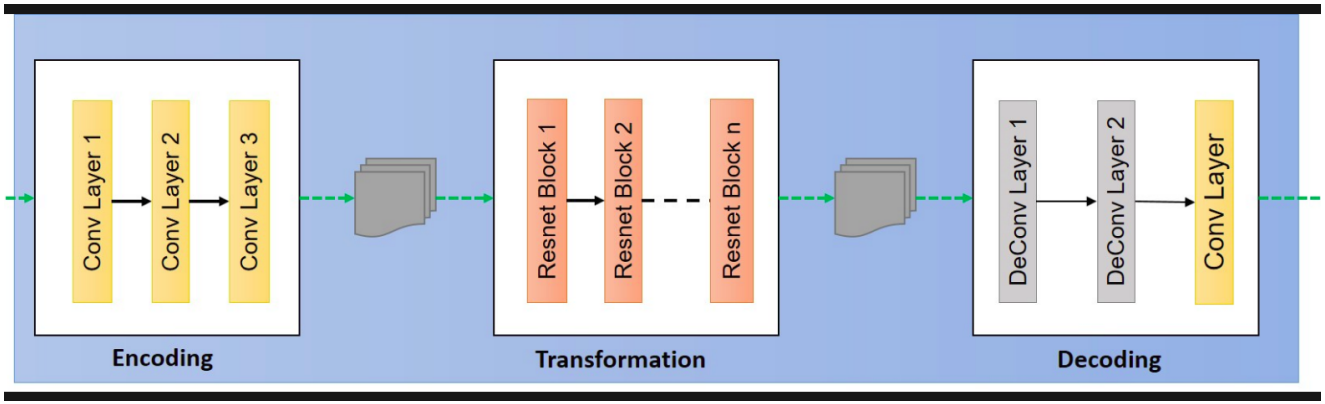

Figure: Generator Architecture

Results and their discussion

1) Generator X2Y tries to learn the model in such a way that the Discriminator Y is not able to differentiate between the generated frame and the real frame.
2) Similarly Generator Y2X tries to learn the model in such a way that the Discriminator X is not able to differentiate between the generated frame and the real frame.
3) After successfully training the model both the Generators would be able to make fool of the Discriminator.

Conclusion and Future Work

At this stage we can conclude that now we have basic understanding of CycleGAN, it's architecture and how to implement it. From now onward our major focus will be on the implementation part. We will try to achieve best model by varying the architecture of Generators and Disrciminator.

Work Done

The current scenario of the work done is as follows:
1) We are trying to collect and read as many papers related to style transfer as possible so that we can get essence of it.
2) Basic understanding of CycleGAN, TextureGAN and related stuff.
3) Trying to implement CycleGAN model.
4) Reading and understanding blogs like Turning Fortnite into PUBG with Deep Learning (using CycleGAN).
5) Also we are following prof.Andrew Ng tutorial on style transfer.