7th Semester Project Report

Unpaired Visual Domain Translation using Advanced GAN

Indian Institute of Information Technology,Allahabad

Submitted by :

Roll No	Name
IIT2015032	Rohan MR
IIT2015042	Raghav Saboo
IIT2015044	Jayesh Chaudhari

Supervised by
Dr. K.P.Singh

Candidate's Declaration

This is to certify that the project report entitled

Unpaired Visual Domain Translation using Advanced GAN submitted to the Dept. of Information Technology, Indian Institute of Information Technology, Allahabad in partial fulfillment of the 7th Semester Mini-Project work, is a record bonafide work carried out by:

IIT2015032	Rohan MR
IIT2015042	Raghav Saboo
IIT2015044	Jayesh Chaudhari

This project is our original work, and it has not been presented anywhere else for any purpose.

Supervisor's Certficate

THIS IS TO CERTIFY THAT THE PROJECT REPORT ENTITLED

Visual Domain Translation using Advanced GAN

submitted to the Dept. of Information Technology, Indian Institute of Information Technology, Allahabad in partial fulfillment of the 7th Semester Mini-Project work, is a record bonafide work carried out by:

IIT2015032	Rohan MR
IIT2015042	Raghav Saboo
IIT2015044	Jayesh Chaudhari

under my supervision and guidance.

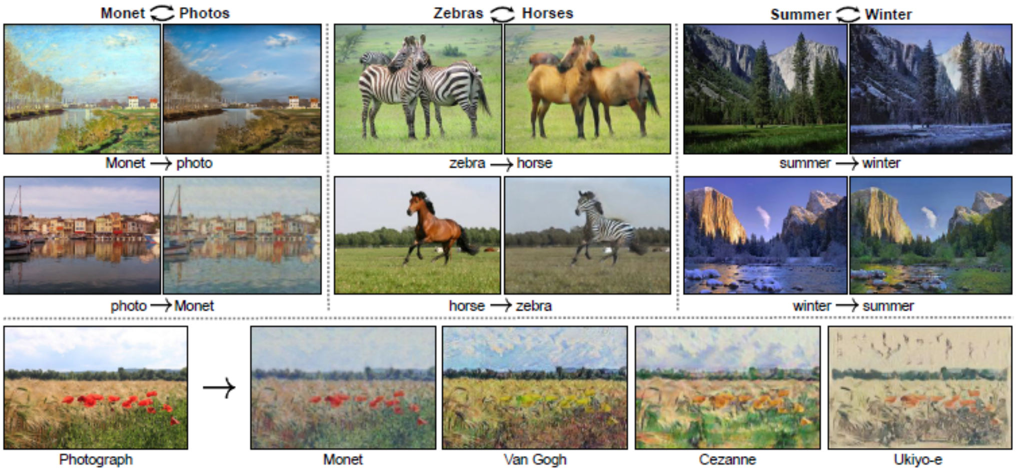
No part of this project has been submitted elsewhere for any purpose.

Dr. K.P. Singh
Dept.of I.T.
IIIT Allahabad

Contents

1. Introduction

Visual Domain translation is a class of vision and graphics problems where the goal is to learn the mapping between an input image and an output image using a training set of aligned image pairs. However, for many tasks, paired training data will not be available. 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. When we work with unpaired images for image translation problem, normal GAN cannot be used as the loss function in a GAN works by minimizing loss w.r.t paired (defined) output image.So to use the power of GANs for achieving our objective we use a modified version of GAN which minimizes a more subjective and obvious loss function Thereby we extend this synthesized model for paired and unpaired video-to-video translation thus covering the whole visual domain .



2. Motivation

The capability to model and recreate the dynamics of our visual world is essential to building intelligent agents. Apart from purely scientific interests, learning to synthesize continuous visual experiences has a wide range of applications in computer vision, robotics, and computer graphics. For example, in model-based reinforcement learning , a image synthesis model finds use in approximating visual dynamics of the world for training the agent with less amount of real experience data. Using a learned image synthesis model, one can generate realistic images without explicitly specifying scene geometry, materials, lighting, and their dynamics, which would be cumbersome but necessary when using standard graphics rendering techniques Field of Artificial intelligence and specifically machine learning needs ample amount of data to increase efficiency of model while training. Taking an example to relate, let us say we are training model for self driving cars. To properly train the model you need equal amount of training data in day and night conditions. But problem comes when we are not able to collect data for night conditions due to any unspecified reason. To overcome above mentioned problem and other related problems we need an image to image translator for related domains with very high accuracy for unpaired images.

3. Problem Statement

Our goal is to learn a mapping $\mathrm{G} : \mathrm{X} \longrightarrow \mathrm{Y}$ such that the distribution of images from G(X) is indistinguishable from the distribution Y using an adversarial loss. Because this mapping is highly under-constrained, we couple it with an inverse mapping $\mathrm{F} : \mathrm{Y} \longrightarrow \mathrm{X}$ and introduce a cycle consistency loss to enforce $\mathrm{F}(\mathrm{G}(\mathrm{X})) \approx \mathrm{X}$ (and vice versa). Qualitative results will be presented on several tasks where paired training data does not exist, including collection style transfer, object transfiguration, season transfer, photo enhancement, etc. Quantitative comparisons against several prior methods demonstrate the superiority of our approach. We extend this synthesized model to video for video to video translation.

4. Literature Review

Generative Adversarial Networks [13]- We have used GANs to build our model. GANs are deep neural network architectures comprised of two neural networks contesting with each other like two players in a minimax game. It can learn to mimic any distribution of data and thus it has a huge potential. It has two components namely the generator and discriminator. The generator generates new instances of data and the discriminator evaluates them and predicts whether the sample is real or fake. The generator is analogous to a team of counterfeiters that are trying to generate fake currency and the discriminator is analogous to police as they are trying to detect fake currency. Competition in the game drives both the teams to improve their methods until the fake currency is indistinguishable from the real ones. GANs have shown extremely good results in generating images that are almost indistinguishable from the real ones and thus they can be used for many applications like converting images to higher resolution , generating datasets in scenarios where less data is available , image to image translation etc.

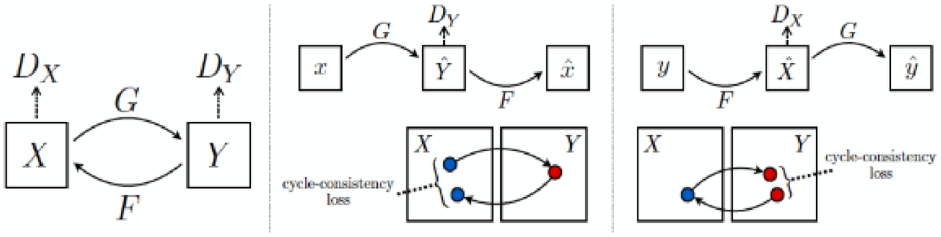
Improved techniques for training GANs [14] - Some useful tricks for faster training and convergence are - Normalize the inputs - Normalize the images between -1 and 1. Modified loss function - The loss function to optimize is min log(1-D(G(z)) but in practice it is better to use max log (D(G(z)) as the first one leads to the problem of vanishing gradients. Distribution for z - It is better to use gaussian distribution rather than normal distribution. Label smoothing - If you have two target labels ie Real=1 and Fake=0, then for each incoming sample, if it is real, then replace the label with a random number between 0.7 and 0.9, and if it is a fake sample, replace it with 0.0 and 0.3 (for example). Avoid Sparse Gradients - The stability of the GAN suffers if you have sparse gradients thus it is better to use Leaky ReLU activation function for both generator and discriminator and for Downsampling use Average Pooling, Conv2d + stride and for Upsampling use PixelShuffle, ConvTranspose2d + stride. Use adam optimizer for gradient descent and use dropout layer for avoiding overfitting.

Image to Image translation using Cycle GAN [**12**] - Image to Image translation is a class of computer vision and graphics problem where the goal is to learn the mapping between input image and output image. Earlier approaches required paired training dataset which is not available for all the tasks. So Cycle GANs can solve this problem ie. they can translate an image from a source domain to a target domain without paired training examples. The model was tested on several datasets like horse to zebra, summer to winter, apple to oranges, aerial photos to maps, semantic labels to photos, monets to photos,sketches to photos and many more. The results were really great and there are many applications of this like collection style transfer, object transfiguration , season transfer and photo generation from paintings.

Pix2Pix [**1**] uses paired set of images to train a model to translate images from one domain to the other. It uses what it calls a conditional GAN which learns loss values by measuring the difference between the generated and existing pair of the image. **Vid2Vid**[**15**] is an extension to the research done in [1]. When [1] is applied to video data, the video generated in unstable as the consecutive images generated are uncorrelated. This happens because adjacent frames predicted are not correlated. [15] solves this problem by adding an extra discriminator which determines if the predicted image corresponds to the sequence of previously predicted images.

5. Methodology

Unpaired video to video translation requires fusion of unpaired image to image translation and paired video translation concepts. Firstly image to image translation needs to be done. We need to convert an image in domain X to an image in domain Y. We have two generators G and F that will convert an image from domain X to Y and from domain Y to X respectively. We have two discriminators DX and DY that distinguish between images in domain X and images generated by generator G and between images in domain Y and images generated by generator F respectively.



$G : X \rightarrow Y$
$F : Y \rightarrow X$

G and F map images from domain X to Y and vice versa.In a nutshell the model works by taking an image say A from domain X and giving it to the generator G which converts it to an image in domain Y and then this image is fed to another generator F which converts it to an image in domain X say $\mathrm{Cyc_A}$ and thus $\mathrm{Cyc_A}$ should be close to A to define a meaningful mapping that is absent in unpaired data set.
The full objective function is -

$$\mathcal{L}\left(G, X, D_{\mathrm{X}}, D_{\mathrm{Y}}\right) = \mathcal{L}_{\mathrm{GAN}}\left(G, D_{\mathrm{Y}}, X, Y\right) + \mathcal{L}_{\mathrm{GAN}}\left(F, D_{\mathrm{X}}, Y, X\right) + \lambda \mathcal{L}_{\mathrm{cyc}}\left(G, F\right) \tag{5.1}$$

The objective function has 2 adversarial losses and one cyclic loss.Lambda controls the relative importance of two objectives. If it is more than 1 then it means that we give more importance to cyclic loss.

$$\mathcal{L}_{\text{cyc}}(G, F) = E_{x\backsim p_{data}(X)}[\| F(G(x)) - x \|_1] + E_{y\backsim p_{data}(Y)}[\| G(F(y)) - y \|_1] \tag{5.2}$$

This is the cyclic consistency loss. F(G(x)) is the regenerated image that we want should be similar to x and same goes for G(F(y)).

This is the standard adversarial loss used in GAN.

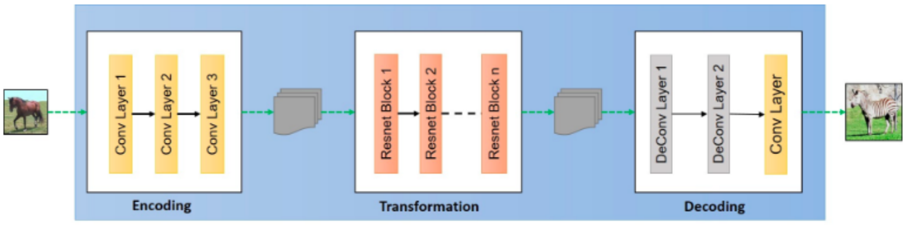
$$\mathcal{L}_{GAN}(G, D_{\text{Y}}, X, Y) = E_{y\backsim p_{data}(Y)}[log D_Y(y)] + E_{x\backsim p_{data}(x)}[log(1 - D_Y G(x))] \tag{5.3}$$

This is the equation we need to solve.

$$G^{*}F^{*} = \arg\min_{G,F}\max_{D_X, D_Y} \mathcal{L}(G, F, D_X, D_Y) \tag{5.4}$$

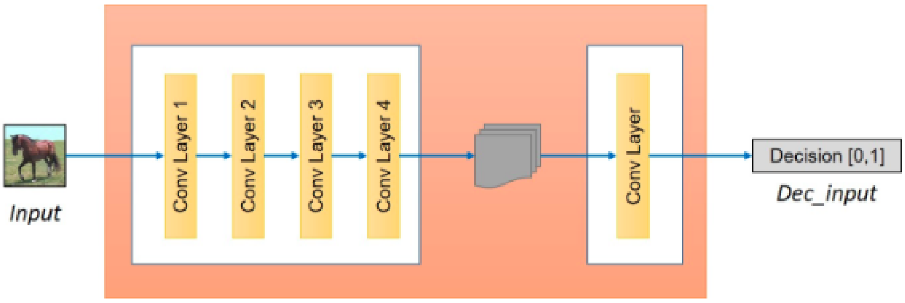
Building the generator -

The structure of generator has three components encoder, transformer, and decoder. In encoding high level features of input image are extracted and then a feature encoding representing the input image is formed. It consists of convolution and pooling layers. The transformer transforms the feature vector of an image in Domain X to the feature vector of an image in Domain Y. Then the decoder converts this feature vector into an image using deconvolution layers.



Building the Discriminator -

Discriminator is just a CNN which classifies the image fed to it as real or fake.



The loss function for the discriminator will have two parts. First the Discriminator must approve all the original images ie. it must predict output 1 for real images and secondly it must output 0 for fake images generated by the generator. Thus the Discriminator D_X will minimize $(D_X(x) - 1)^2$ where x is a real image and minimize $(D_X(F(y)))^2$ as F(y) is the fake image generated and similarly for discriminator D_Y.

The loss function for generator also has two parts. Firstly the Generator should eventually be able to fool the discriminator about the authenticity of it's generated images. This can done if the recommendation by discriminator for the generated images is as close to 1 as possible. So generator would like to minimize $(D_Y(G(x)) - 1)^2$ and also the cyclic loss
$\| F(G(x)) - x \| + \| G(F(y)) - y \|$.

Thus with the loss function defined we will train the model to minimize the loss using gradient descent with alternate gradient descent on the discriminator and generator.

We have tested the model on these data sets - horse to zebras, summer to winter, apple to oranges, aerial photos to maps, semantic labels to photos, monets to photos and sketches to photos.

After having implemented the model for unpaired image to image translation we will now use the concept of unpaired image to image translation and paired video translation for unpaired video to video translation.

The goal is to learn a mapping function to convert an input video to an output video. Let $s_1^T = \{s_1, s_2, ...s_T\}$ be a sequence of source images for video synthesis. Let $x_1^T = \{x_1, x_2, ...x_T\}$ be a sequence of corresponding output images.

There are two generators G and F and four Discriminators $D_{FI}, D_{FV}, D_{GI}, D_{GV}$.
G : S $\longrightarrow X$
$F : X \longrightarrow S$

G is a generator for mapping input video to output video and F does the same mapping in the reverse direction.
$\tilde{x}_1^T = G(s_1^T)$
$\tilde{s}_1^T = F(x_1^T)$

We use two types of discriminators one is an image discriminator and the other one is a video discriminator.

Purpose of the Image discriminator is to ensure that each output frame resembles a real image given the same source image and the purpose of the Video discriminator is to ensure that consecutive output frames resemble the temporal dynamics of a real video given the same optical flow.D_{FI} and D_{GI} conditions on the image and D_{FV} and D_{GV} conditions on the flow.

x_t is t^{th} frame in the output video
$\tilde{x}_t$ is the t_{th} frame generated by G
s_t is the t_{th} frame in the input video
$\tilde{s}_t$ is the t_{th} frame generated by F
w_{t-1} is the optical flow used to convert x_{t-1} to x_t
o_{t-1} is the optical flow used to convert s_{t-1} to s_t
w_{t-k}^{t-2} is the k-1 optical flow for k consecutive real images x_{t-k}^{t-1}
o_{t-k}^{t-2} is the k-1 optical flow for k consecutive input images s_{t-k}^{t-1}

D_{GI} is the image discriminator for discriminating between images generated by generator G and images or frames in the output video.

D_{FI} is the image discriminator for discriminating between images generated by generator F and images or frames in the input video.

D_{GI} will try to make it's output 1 for x_t and 0 for $\tilde{x}_t$ and D_{FI} will try to make it's output 1 for s_t and 0 for $\tilde{s}_t$.

D_{GV} is the video discriminator for discriminating between k consecutive images generated by generator G given the optical flow and k consecutive images or frames in the output video given the same optical flow.

D_{FV} is the video discriminator for discriminating between k consecutive images generated by generator F given the optical flow and k consecutive images or frames in the input video given the same optical flow.

D_{GV} will try to make it's output 1 for (x_{t-k}^{t-1} , w_{t-k}^{t-2}) and 0 for ($\tilde{x}_{t-k}^{t-1}$, w_{t-k}^{t-2}) and D_{FV} will try to make it's output 1 for (s_{t-k}^{t-1} ,o_{t-k}^{t-2}) and 0 for ($\tilde{s}_{t-k}^{t-1}$,o_{t-k}^{t-2}).

Thus we have two types of losses adversarial loss which is the one used in Standard GAN and cycle loss used in Cycle GAN.

There are 4 adversarial losses -

$$\mathcal{L}(G, D_{GI}, S, X) = E_{\phi_I(x_1^T)}[log D_{GI}(x_i)] + E_{\phi_I(\tilde{x}_1^T)}[log(1 - D_{GI}(\tilde{x}_i))] \tag{5.5}$$

$$\mathcal{L}(G, D_{GV}, S, X) = E_{\phi_V(x_1^T, w_1^{T-1})}[log D_{GV}(x_{i-k}^{i-1}, w_{i-k}^{i-2})] + E_{\phi_V(\tilde{x}_1^T, w_1^{T-1})}[log(1 - D_{GV}(\tilde{x}_{i-k}^{i-1}, w_{i-k}^{i-2}))] \tag{5.6}$$

$$\mathcal{L}(F, D_{FI}, X, S) = E_{\phi_I(s_1^T)}[log D_{FI}(S_i)] + E_{\phi_I(\tilde{s}_1^T)}[log(1 - D_{FI}(\tilde{s}_i))] \tag{5.7}$$

$$\mathcal{L}(F, D_{FV}, X, S) = E_{\phi_V(s_1^T, o_1^{T-1})}[log D_{FV}(s_{i-k}^{i-1}, o_{i-k}^{i-2})] + E_{\phi_V(\tilde{s}_1^T, o_1^{T-1})}[log(1 - D_{FV}(\tilde{s}_{i-k}^{i-1}, o_{i-k}^{i-2}))] \tag{5.8}$$

Cycle loss -

$$\mathcal{L}_{cyc}(G, F) = \frac{1}{T}\sum_{t=1}^{T}[\parallel x_t - F(G(\tilde{x}_{t-l}^{t-1}, s_{t-l}^{t})) \parallel_1 + \parallel s_t - G(F(\tilde{s}_{t-l}^{t-1}, x_{t-l}^{t})) \parallel_1] \tag{5.9}$$

Total loss function -

$$\begin{aligned}\mathcal{L}(G, F, D_{GI}, D_{GV}, D_{FI}, D_{FV}) = \mathcal{L}(G, D_{GI}, S, X) + \mathcal{L}(G, D_{GV}, S, X) \\ +\mathcal{L}(F, D_{FV}, X, S) + \mathcal{L}(F, D_{FV}, X, S) + \mathcal{L}_{cyc}(G, F)\end{aligned} \tag{5.10}$$

Final objective function -

$$G^*, F^* = \arg\min_{G,F} \max_{D_{GI}, D_{GV}, D_{FI}, D_{FV}} \mathcal{L}(G, F, D_{GI}, D_{GV}, D_{FI}, D_{FV}) \tag{5.11}$$

6. Implementation Details

Discriminator is a CNN which classifies the image fed to it as real or fake. In our discriminator architecture there are 5 convolutional layers with filter sizes of 4 and no of filters as 64,128,256,512 and stride of 1,2. We have used leaky relu activation function as it solves the problem of vanishing gradient which happens with sigmoid and tanh and also the output is not zero for input value less than zero. Input to the discriminator is a (128,128,3) RGB image and output is 1 for real image and 0 for fake image.

Generator is a Neural Network with convolutional layers for extracting the higher level features from the input image and thereby converting input image to an encoding vector and then we use resnet blocks for converting the encoded vector of an image in input domain to an encoded vector of an image in output domain and then use deconvolutional layers for upsampling the encoded vector back to an image. We have used 6 resnet blocks along with batch normalization and the activation function used is leaky relu. In our generator there are 7 convolutional layers and 7 deconvolutional layers with filter sizes of 4 and no of filters as 64,128,256,512 and stride of 1,2. Dropout is also used in some layers to prevent overfitting. Input to the generator is a (128,128,3) RGB image in the input domain and output is a (128,128,3) RGB image in the output domain.

We have used a learning rate of 0.0002 for both generator and discriminator. Adam optimizer is used for gradient descent and we have trained our model on 6 datasets namely day to night , horse to zebra, apple to orange , Earth view to Map view , segmented scapes to city scapes and Summer to Winter. We ran our model for 500 epochs on all the datasets with a batch size of 1 thus effectively applying a stochastic gradient descent.

7. Software Requirements

Software Used

- Python 3.5.2
- Pycharm Community Edition 2017

Libraries Used

- Tensor Flow
- Keras
- NumPy
- SciPy
- Sklearn

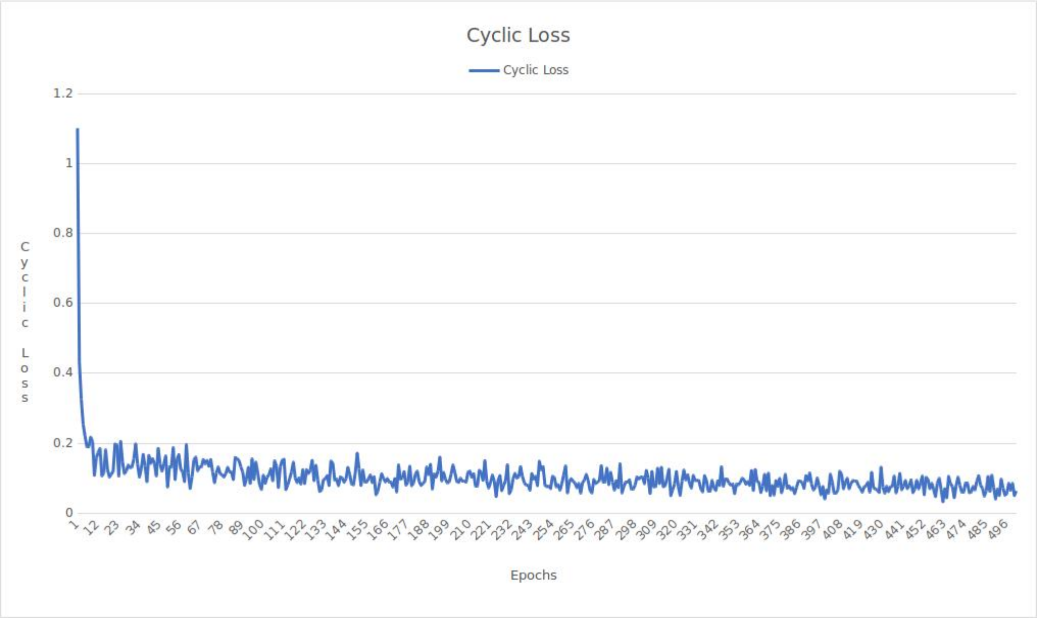
System Specification

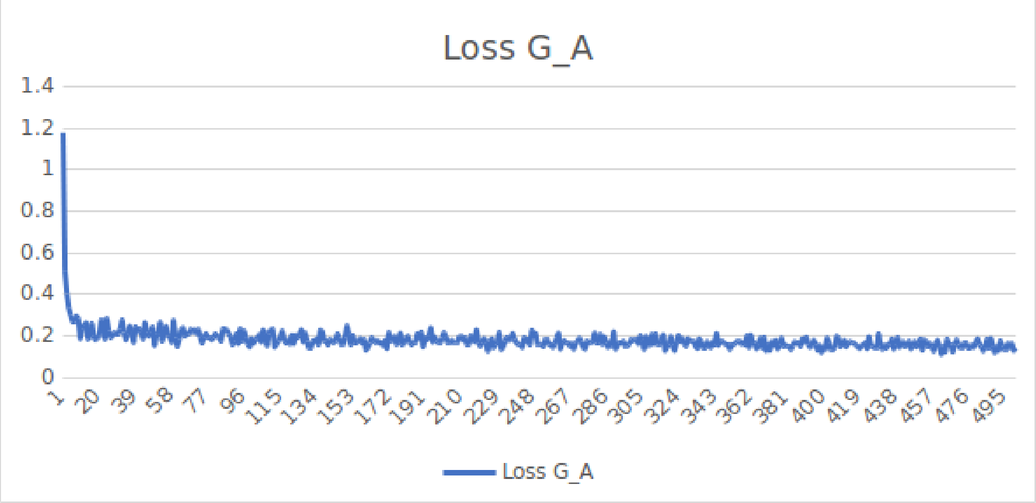
- CPU Intel Xeon E5-2640
- GPU Nvidia Geforce 1080i
- RAM 32 GB

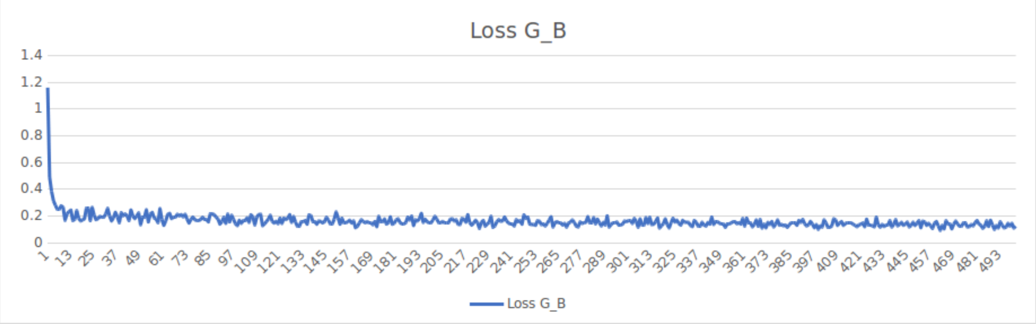
8. Results

The results below are shown for first 500 epochs of training night to day data. Similar results were obtained during training of Horse to Zebra, Summer to Winter, Earth view to Map view, Apple to oranges and Cityscapes to segmented scapes Dataset.

To guarantee success of such type of model while training, we take into account the decrement of losses namely cyclic loss all over the model. For this particular case there are in total five losses namely , Loss DA, Loss DB, Loss GA, Loss GB and Cyclic loss.


Cyclic Loss
Cyclic Loss
Cyclic Loss
1.2
1
0.8
0.6
0.4
0.2
0
1 12 23 34 45 56 67 78 89 100 111 122 133 144 155 166 177 188 199 210 221 232 243 254 265 276 287 298 309 320 331 342 353 364 375 386 397 408 419 430 441 452 463 474 485 496
Epochs


Loss G_A
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1 20 39 58 77 96 115 134 153 172 191 210 229 248 267 286 305 324 343 362 381 400 419 438 457 476 495
Loss G_A


Loss G_B
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1 13 25 37 49 61 73 85 97 109 121 133 145 157 169 181 193 205 217 229 241 253 265 277 289 301 313 325 337 349 361 373 385 397 409 421 433 445 457 469 481 493
Loss G_B

The following images show results obtained by training our model on various datasets. The results format shows Source image in first row, generated image in second row and reconstructed image in third row.

Horse to Zebra

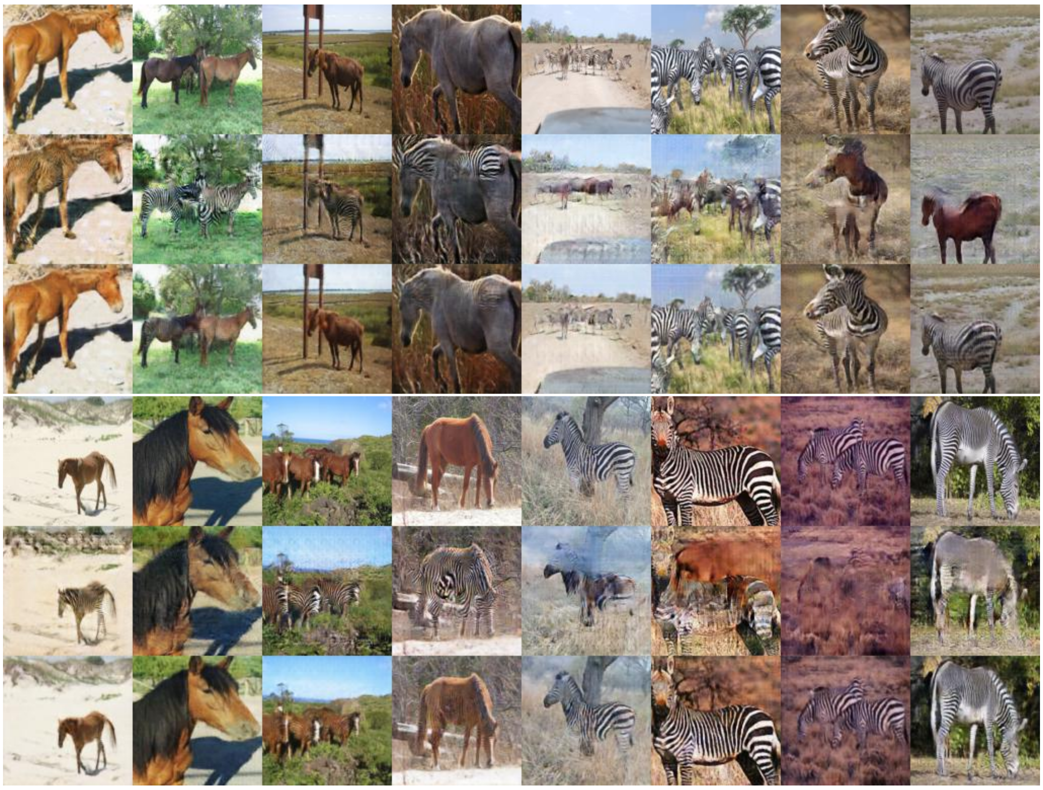


Fig 2.

Apples to Oranges

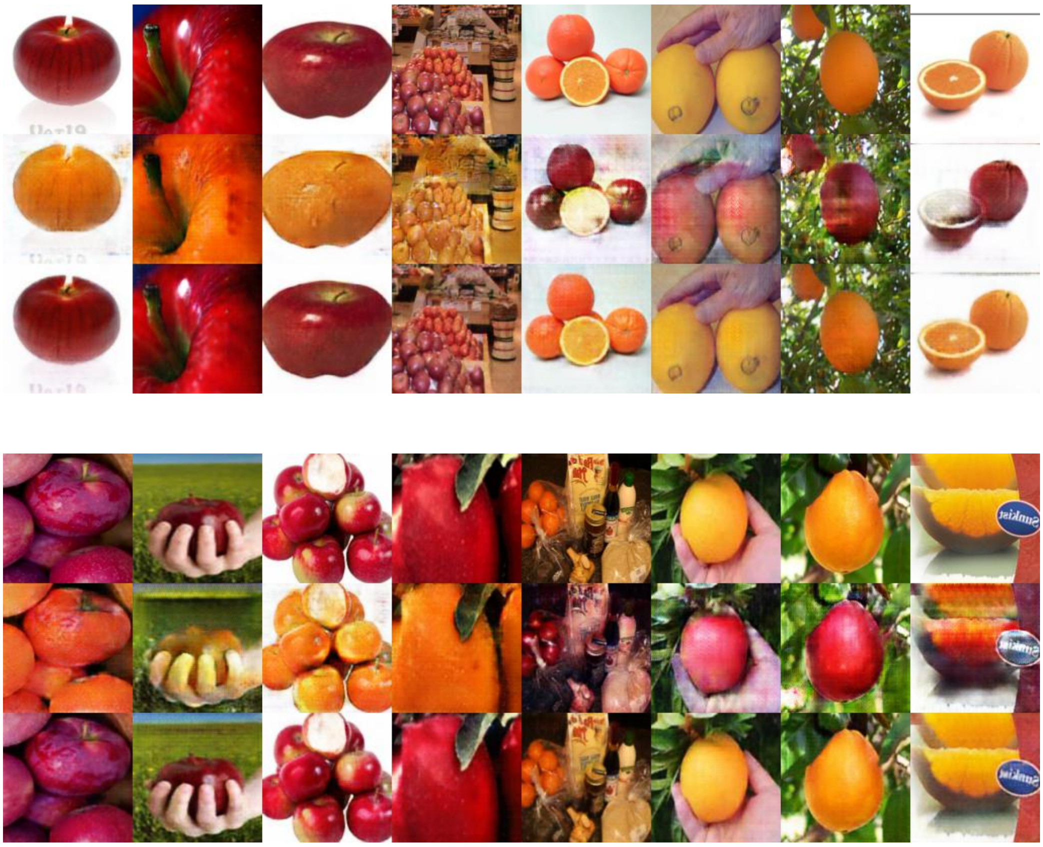


Fig 3.

Summer to Winter

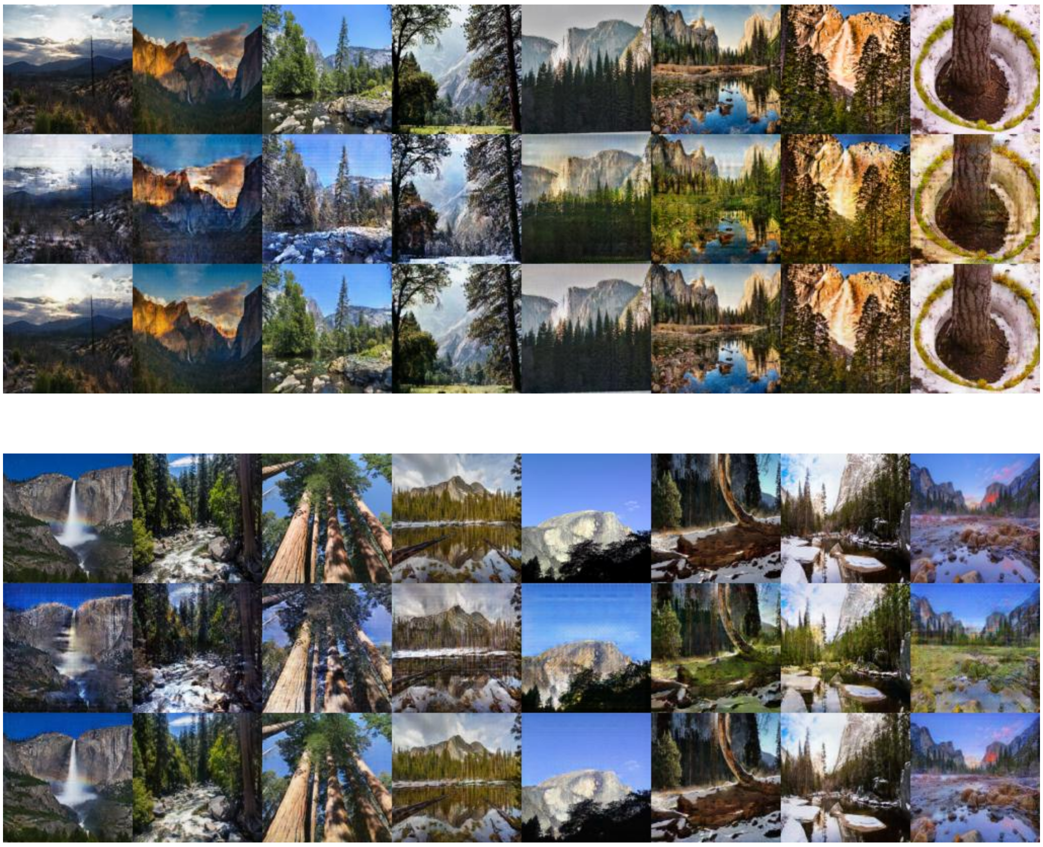


Fig 4.

Earth view to Map view

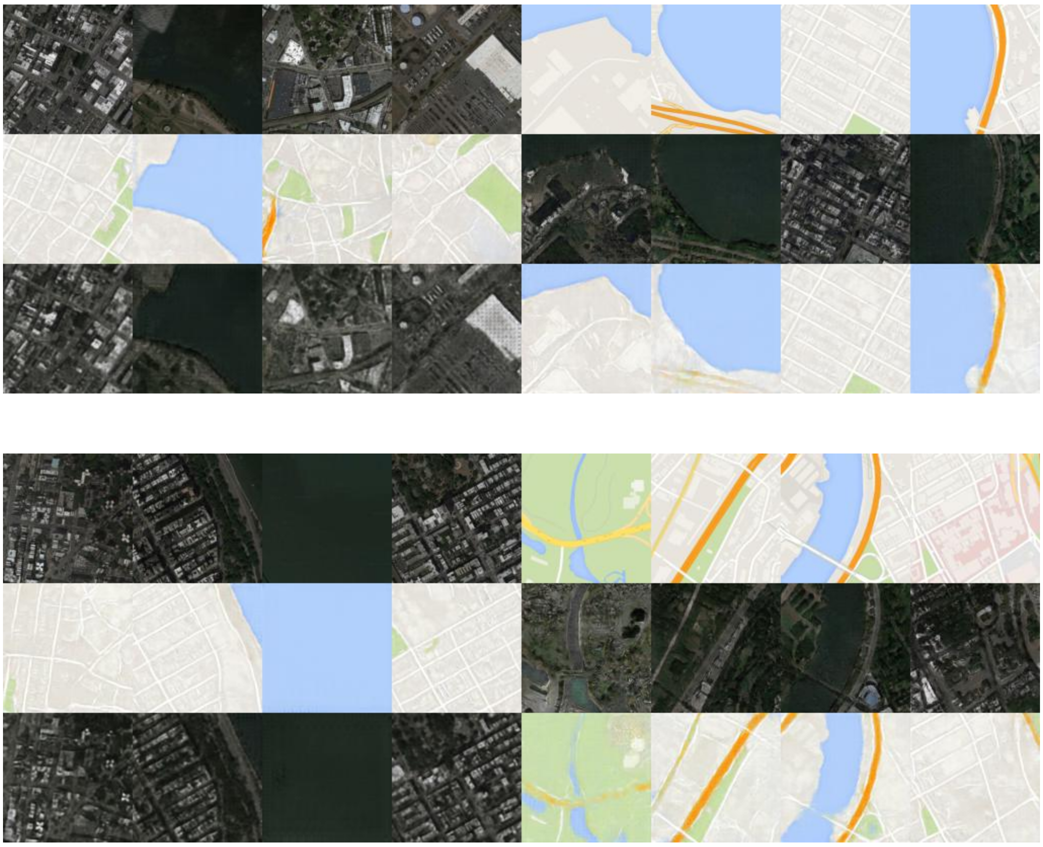


Fig 5.

Day to Night

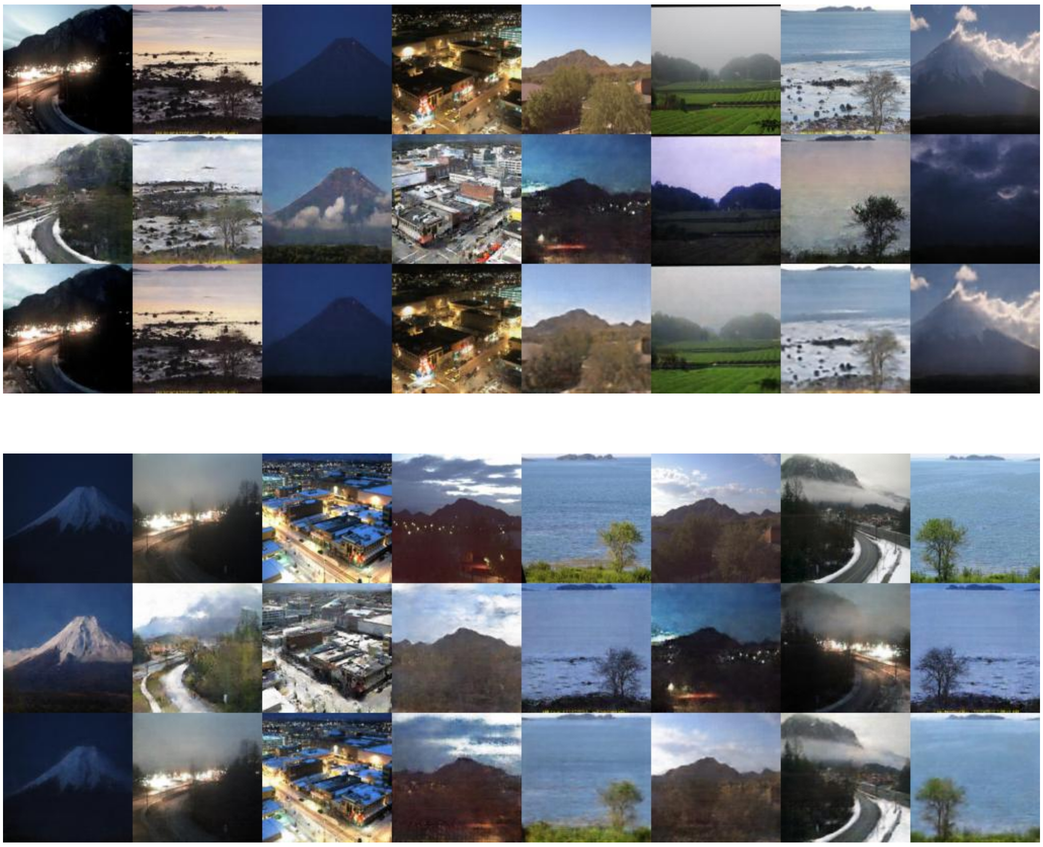


Fig 6.

.

Cityscapes to Segmented scapes

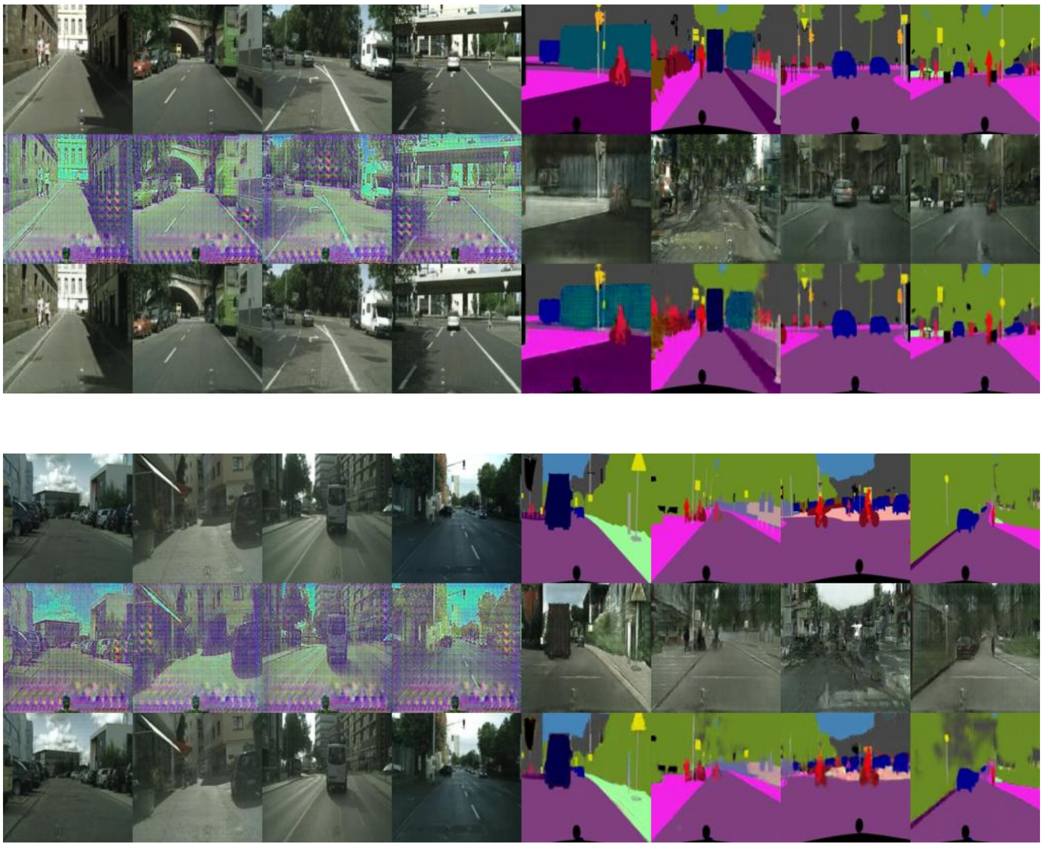


Fig 7.

9. Conclusion

We present a general video-to-video synthesis framework based on modified - GAN. Through carefully-designed generator and discriminator networks as well as a optical flow included objective,we can synthesize good-resolution, photorealistic, and temporally consistent videos. Extensive experiments demonstrate that our results on images are significantly better than the results by state-of-the-art methods. Its extension to the paired image to image (pix-to-pix) methodology and it also compares favorably against the competing approaches. Limitations and future work for our proposed image to image model - Although our approach outperforms previous methods, our model still fails in a couple of situations. For example, our model performs very bad for image size more than 256*256. We anticipate this problems cause to be lack of good dataset and computational power.

References

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

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

[3] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In SIGGRAPH, 2001.

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

[5] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In ECCV, 2016.

[6] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays. Transient attributes for high-level understanding and editing of outdoor scenes. ACM TOG, 33(4):149, 2014.

[7] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. In ICLR, 2016.

[8] Y. Shih, S. Paris, F. Durand, andW. T. Freeman. Data driven hallucination of different times of day from a single outdoor photo. ACM TOG, 32(6):200, 2013.

[9] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In ECCV, 2016.

[10] S. Xie and Z. Tu. Holistically-nested edge detection. In ICCV, 2015

[11] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In ECCV, 2016.

[12] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In IEEE International Conference on Computer Vision (ICCV), 2017.

[13] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozairy, Aaron Courville, Yoshua Bengioz. Generative Adversarial Nets. In arXiv:1406.2661 2014.

[14] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen. Improved Techniques for Training GANs. In arXiv:1606.03498 2016.

[15] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, Bryan Catanzaro. Video-to-Video Synthesis. In arXiv:1808.06601.