CS7015: Deep Learning

Divya K Raman, EE15B085

9th December 2018

1 Introduction

Transferring the style from one image onto another can be considered a problem of texture transfer. In the texture transfer problem, the goal is to synthesize a texture from a source image while constraining the texture synthesis in order to preserve the semantic content of a target image. Recent approaches use deep convolutional neural networks and GANs. Two types of problems are explored - texture transfer using paired data and texture transfer using unpaired data. For texture transfer using unpaired data, cross domain GANs are used where a cyclic loss component is applied to the total loss. Using cycleGANs, photo to Vincent Van Gogh or Monet style painting; horse to zebra; Yosemite summer to winter transformations can be made.

Firstly, we present a literature survey of 3 important papers related to the topic. We then conduct experiments and analyze the results of cycleGAN in detail. We then take up the object transfiguration(horse2zebra) failure cases and look for methods to improvise them. A paper presenting one method to improve the results SemGAN is discussed. We finally propose two methods to improve the results - object detection and attention mechanism. The object detection method has been implemented and analyzed in detail.

2 Literature Survey

2.1 Image Style Transfer Using Convolutional Neural Networks; LA Gatys et al

In this paper, the authors demonstrate how convolutional neural networks can be used to independently process and manipulate the content and the style of natural images. They introduce an algorithm 'A Neural Algorithm of Artistic Style' to perform image style transfer.

Content representation: The input image is encoded in each layer of the Convolutional Neural Network by the lter responses to that image. The feature representation at each layer l for both the real image and the generated image is

first found. The content loss at layer l is then defined as the squared-error loss between the two feature representations. Higher layers in the network capture the high-level content in terms of objects and their arrangement in the input image but do not constrain the exact pixel values of the reconstruction very much. In contrast, reconstructions from the lower layers simply reproduce the exact pixel values of the original image. The feature responses in higher layers of the network is therefore referred to as the content representation.

Style Representation: To obtain a representation of the style of an input image, a feature space built on top of the lter responses in any layer of the network is designed to capture texture information. It consists of the correlations between the different lter responses, where the expectation is taken over the spatial extent of the feature maps. These feature correlations are given by the Gram matrix G_l belonging to R^{NlXNl}, where G_lij is the inner product between the vectorised feature maps i and j in layer l. We then obtain the the information captured by these style feature spaces built on different layers of the network by constructing an image that matches the style representation of a given input image. Style loss is the scaled euclidean difference between the gram matrices of the real image and the generated image summed over all layers.

A weighted sum of the content loss and style loss is minimised. This paper deals with artistic style transfer. Photorealism is not fully preserved.

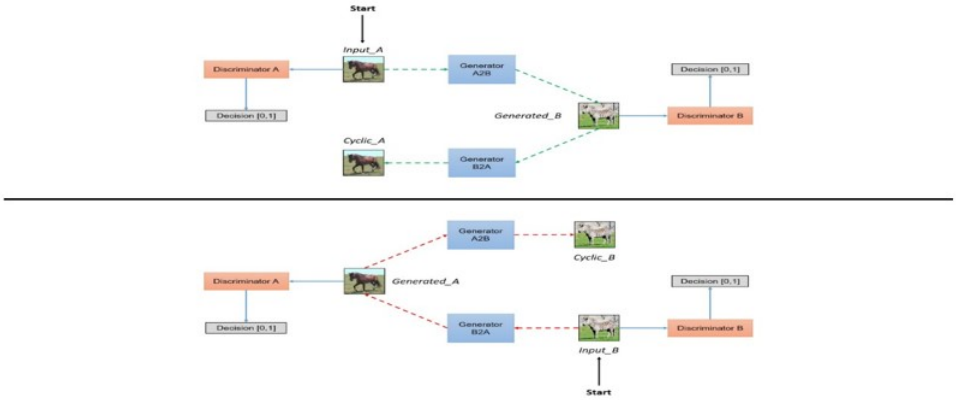
2.2 Deep Photo Style Transfer; Fujun Luan et al

The previous paper dealt with artistic style transfer. Photorealism was not preserved. This paper yields photorealistic style transfer results in a broad variety of scenarios, including transfer of the time of day, weather, season, and artistic edits. For this, the authors add a photorealism regularization term in the objective function during the optimization, constraining the reconstructed image to be represented by locally afne color transformations of the input to prevent distortions. The authors also incorporate a semantic labeling of the input and style images into the transfer procedure so that the transfer happens between semantically equivalent subregions and within each of them, the mapping is close to uniform (This prevents tree from getting mapped to sky, etc). The loss function has three components: content loss, augmented style loss and photorealism regularisation.

2.3 Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks; Jun-Yan Zhu et al

This is the paper we explore in detail in our project.

This approach is extremely useful when paired training data is not available. This paper proposes a method to translate an image from a source domain X to a target domain Y in the absence of paired examples. The goal is to learn a mapping G : X -¿ Y such that the distribution of images from G(X) is indistinguishable from the distribution Y using an adversarial loss. Since this constrained is insufficient, an inverse mapping F : Y -¿ X is also applied. The authors further introduce a cycle consistency loss to enforce F(G(X)) = X (and vice versa). There are two associated adversarial discriminators D_Y and D_X. D_Y encourages G to translate X into outputs indistinguishable from domain Y , and vice versa for D_X and F. Forward cycle consistancy and backward cycle consistancy losses are applied. The total loss is a weighted sum of the two adversarial losses and the two cycle consistancy losses.
Convolutional neural networks are used to model the generator and discriminator networks.



3 CycleGAN code, Experimentation and Results

The base code that we improvise and work upon is from https://github.com/golbin/CycleGAN. Datasets are got from https://people.eecs.berkeley.edu/ taesung park/CycleGAN/datasets/.

One major training observation:
The generator and discriminator losses vary erratically. There is no convergence trend. However, visually, the results get better as training proceeds. Testing the model is done after epoch and results are saved. The results of the model at different training stages for the vangogh2photo is given below:

Vangogh2photo dataset:

Training takes 1 day and 6 hours.

After epoch 1:
Vangogh2photo: The photos in order are generated, real, reconstructed.

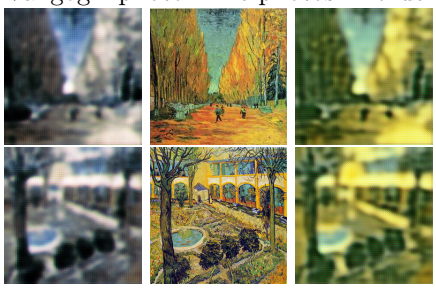
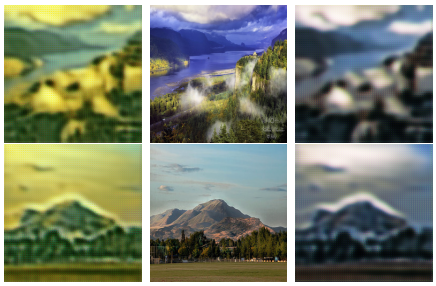
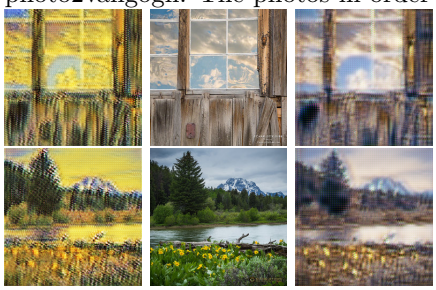


photo2vangogh: The photos in order are generated, real, reconstructed.



After epoch 10:
Vangogh2photo: The photos in order are generated, real, reconstructed.

photo2vangogh: The photos in order are generated, real, reconstructed.



After epoch 20:
Vangogh2photo: The photos in order are generated, real, reconstructed.

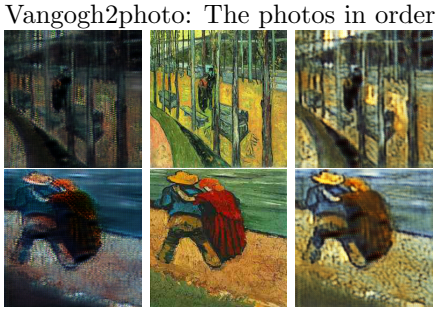


photo2vangogh: The photos in order are generated, real, reconstructed.

After epoch 40:
Vangogh2photo: The photos in order are generated, real, reconstructed.

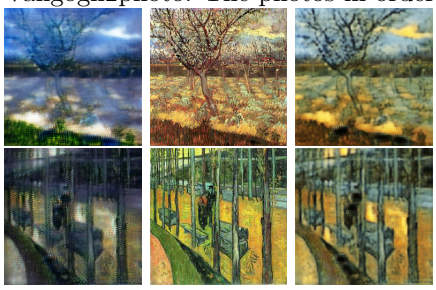


photo2vangogh: The photos in order are generated, real, reconstructed.

After epoch 44:
Vangogh2photo: The photos in order are generated, real, reconstructed.

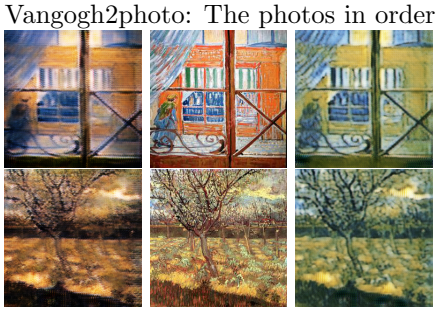


photo2vangogh: The photos in order are generated, real, reconstructed.

After epoch 55:
Vangogh2photo: The photos in order are generated, real, reconstructed.

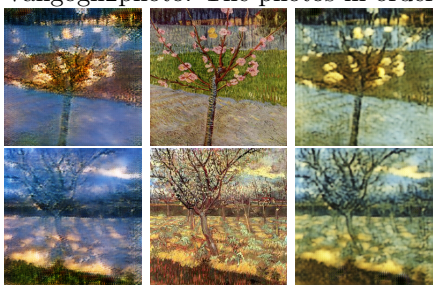
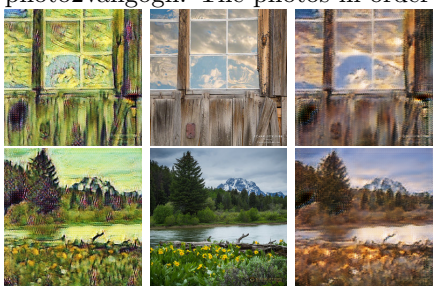


photo2vangogh: The photos in order are generated, real, reconstructed.



After epoch 60: Final results
Vangogh2photo: The photos in order are generated, real, reconstructed.

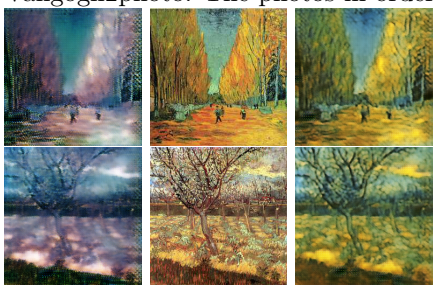


photo2vangogh: The photos in order are generated, real, reconstructed.

GAN generator and discriminator training losses vary very erratically. There doesn't seem to be any trend. Hence, visual interpretation seems to be the best way to see how well the method is performing. We clearly see an improvement in the results as training proceeds. Here, we've tarined for 60 epochs and the above results demonstrate how the results improve as training proceeds. The method works pretty well in both directions -converting vangogh paintings to photos and vice versa.

Few more final results:

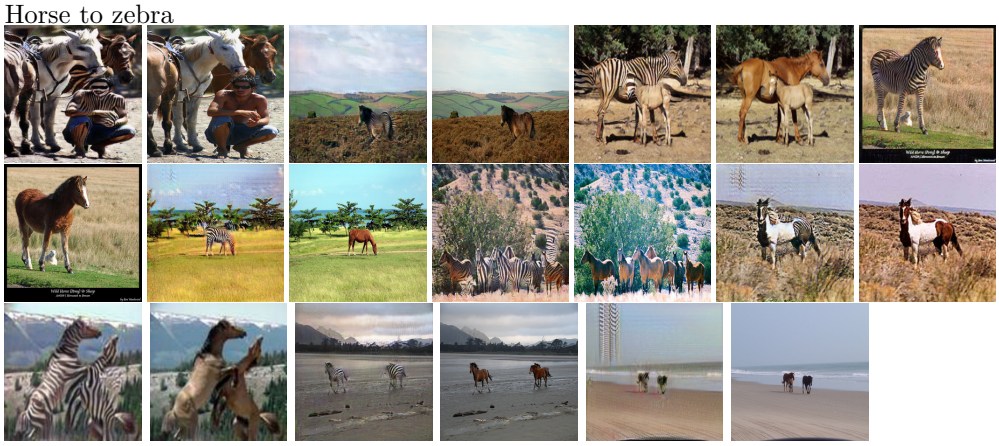
Vangogh2photo: The photos in order are generated, real, reconstructed.

photo2vangogh: The photos in order are generated, real, reconstructed.

Horse2zebra dataset:

We now show a few results on the horse2zebra dataset.

Horse to zebra



Zebra to horse

4 Object transfiguration failure cases: Horse2zebra dataset

The CycleGAN method produces wrong/bad results in a lot of cases. One major dataset where results are bad in a lot of cases is the horse2zebra dataset(object transfiguration case in general).

Also, the stripes of the zebra can appear all over while translating from horse to zebra. The background gets dull(due to the colour of the stripes). Similarly, in the zebra to horse conversion case, the background has shades of brown(colour of the horse). Thus, the entire picture gets distorted. Our aim is to only transform the texture of horse(or zebra) to zebra(or horse). We don't want the background to get affected.

5 Solving the problem

We now look into methods to solve the object transfiguration failure cases.

5.1 SemGAN

The paper 'Sem-GAN:Semantically-ConsistentImage-to-ImageTranslation, Anoop Cherian et al' handles this. The cycleGAN paper demonstrated good results when the domains are unimodal however the performance is bad in multi modal scenarios such as the image segmentation task. This is because invertibility does not necessarily enforce semantic correctness. For this, the authors present sem-GAN(semantically consistent GAN) in which the semantics are dened by the class identities of image segments in the source domain as produced by a semantic segmentation algorithm. Total loss has adversarial loss(for two GANs), cycle consistency component and semantically consistent GAN loss. Semantic segmentation architectures tested upon are PSP Net and FCN. Failure cases are addressed better than cycleGAN but results of other object transfiguration cases(unimodal cases) are not as good as cycleGAN.

5.2 Object detection

We propose an object detection + cycle GAN method to address this issue. The first step is to compute the bounding boxes of the object in question. This can be done by fasterRCNN or YOLO. A simple occlusion based method can also do this wherein we occlude different regions of the image using differently sized occlusion masks and finally take that result which yields the highest classification score for the object in question. The object can then be cropped off the image and a new dataset containing the cropped images can be made. These can be used for training. While testing, again, the object can be cropped, passed through the model and the result can be resized and stitched back with the background to get the entire transformed image. There will however be an issue if the bounding box of the object contains overlapping parts of the other objects in the image(for eg. the leg of the horse rider will appear in the bounding box of the horse if we consider an image of a person riding a horse).

For our experimentation purposes, we crop the test images, pass it to the horse2zebra pretrained model(not trained on the cropped dataset but on the original dataset itself), and stitch back the transformed cropped image to the the background to obtain the complete transformed image. The results are as follows:

Original image:

Transformed original image:

We observe that the rider has also become a zebra and not just the horse!
Cropped image

Transformed cropped image

We now stitch back the transformed horse to the rider and the background.

As we can see, only the horse has been transformed to the zebra and not the rider or the environment.

5.3 Attention mechanism

We propose a method where while training, we can give attention to only the object in question. This might improve results.

6 Further work

We next intend to improvise on the object detection cycle GAN method that we have proposed and train on a more diverse data set to obtain further improvement in the results.

7 Conclusion

We first did a literature survey of various style transfer methods. We analysed the cycleGAN paper in detail and experimented on the horse2zebra and vangogh2photo datasets and its object transfiguration failure cases. We then looked

into methods to solve this problem and implemented and tested a short version of our proposed solution 'object detection cycleGAN'. We intend to extend this work on a more diverse dataset to obtain better results.

8 References

1. Gatys, L. A., Ecker, A. S., Bethge, M. (2016). Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2414-2423).

2. Luan, F., Paris, S., Shechtman, E., Bala, K. (2017). Deep photo style transfer. CoRR, abs/1703.07511, 2.

3. Zhu, J. Y., Park, T., Isola, P., Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint.

4. Cherian, A., Sullivan, A. (2018). Sem-GAN: Semantically-Consistent Image-to-Image Translation. arXiv preprint arXiv:1807.04409.

5. https://hardikbansal.github.io/CycleGANBlog/

6. https://www.digitalocean.com/community/tutorials/how-to-perform-neural-style-transfer-with-python-3-and-pytorch

7. https://github.com/golbin/CycleGAN

8. https://medium.com/@jonathan$_hui/gan-some-cool-applications-of-gans-4c9ecca35900$

9.$https://github.com/junyanz/CycleGAN$

10.$http://hindupuravinash.com/$

11.$http://efrosgans.eecs.berkeley.edu/cyclegan$