Cartoonification Using CycleGAN

1st Suprava Sahoo
Computer Science and IM
Asian Institute of Technology
Bangkok, Thailand
st120984@ait.asia

2nd Shrabin Tuladhar
Computer Science and IM
Asian Institute of Technology
Bangkok, Thailand
st121718@ait.asia

3rd Urusha Rajkarnikar
Computer Science and IM
Asian Institute of Technology
Bangkok, Thailand
st121263@ait.asia

***Abstract*—Image-to-image translation involves generating a new synthetic version of a given image with a specific modification, such as translating a summer landscape to winter. Training a model for image-to-image translation typically requires a large dataset of paired examples. These datasets can be difficult and expensive to prepare, and in some cases impossible, such as photographs of paintings by long dead artists. Hence, in this report we are trying to 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. Our goal is to investigate generative adversarial networks (GANs) as a general-purpose solution to image-to-image translation problems. We try to achieve this on the unpaired dataset (i.e. human faces dataset and simpsons dataset) in contrast to traditional paired one via the use of CycleGANs.**

In CycleGAN as it involves the automatic training of image-to-image translation models without paired examples. The models are trained in an unsupervised manner using a collection of images from the source and target domain that do not need to be related in any way. This simple technique is powerful, achieving visually impressive results on a range of application domains, most notably translating photographs of real people to cartoon characters, and the reverse. Others might include object transfiguration, season transfer, photo enhancement, etc.

***Index Terms*—CycleGAN, Pix2pix, Image Translation, Unsupervised Learning**

I. Introduction

Many problems in image processing, computer graphics, and computer vision can be posed as "translating" an input image into a corresponding output image. Just as a concept may be expressed in either English or French, a scene may be rendered as an RGB image, a gradient field, an edge map, a semantic label map, etc. In analogy to automatic language translation, we define automatic image-to-image translation as the task of translating one possible representation of a scene into another, given sufficient training data. Cartoon faces appear in animations, comics and games. They are widely used as profile pictures in social media platforms, such as Facebook and Instagram. Drawing a cartoon face is labor intensive. Not only does it require professional skills, but also it is difficult to resemble the unique appearance of each person.

Image-to-image translation was first introduced by Isola et al , which utilizes the generative adversarial network (GAN) to translate an image from a source domain to a target domain such that the translated images are close to the ground truth measured by a discriminator network. This method and the follow-up works require paired data for training. However, it is not always easy to obtain a large amount of paired data. Thus, CycleGAN was introduced. It uses the cycle consistency loss to train two pairs of generators and discriminators in order to regularize the solution of trained networks. In this paper, we aim at generating alike cartoon faces for any person automatically. We cast this problem as an image-to-image translation task. However, we consider unpaired training data between cartoon and real faces.

Identify applicable funding agency here. If none, delete this.

II. Related Work

In the computer vision literature, image generation problem is tackled using autoregressive models [21, 29], restricted Boltzmann machines [26], and autoencoders [10]. Recently, generative techniques are proposed for image translation tasks. Models such as GANs [7, 34] and VAEs [23, 15] achieve impressive results in image generation. They are also utilized in conditional settings [12, 38] to address the image-to-image translation problem. However, in the prior research, relatively less attention is given to the unsupervised setting [20, 37, 4].

Many state-of-the-art unsupervised image-to-image translation frameworks are developed based on the cycle-consistency constraint [37]. Liu et al. [20] showed that learning a shared-latent space between the images in source and target domains implies the cycle-consistency. The cycle-consistency constraint assumes that the source image can be reconstructed from the generated image, in the target domain, without any extra domain-specific information [20, 37]. From our experience, this assumption severely constrains the network and degrades the performance and stability of the training process, in the case of learning the translation between different modalities. In addition, this assumption limits the diversity of generated images by the framework, i.e., the network associates a single target image with each source image. To tackle this problem, some prior research attempts to map a single image into multiple images in the target domain in a supervised setting [5, 3]. This problem is also addressed in [2] an unsupervised setting. However, they have not considered any mechanisms to force their auxiliary latent variables to represent only the domain-specific information.

Unpaired image-to-image translation frameworks (Zhu et al., 2017a;b; Liu et al., 2017; Shrivastava et al., 2017; Kim et al., 2017) such as CycleGAN remove the requirement of

having detailed pixel level supervision. In CycleGAN this is achieved by enforcing a bi-directional prediction from source to target and target back to source, with an adversarial penalty in the translated images in the target domain. Similar unsupervised circularity-based approaches (Kim et al., 2017; Yi et al., 2017) have also been developed. The CycleGAN family models (Zhu et al., 2017a;b) point to an exciting direction of unsupervised approaches but they also create artifacts in many applications. One reason for this is that the circularity constraint in CycleGAN lacks the straightforward description of the target domain, so it may change the inherent properties of the original dataset and generate unexpected results which are inconsistent.

In this work, in contrast, we aim to learn distinct domain-specific and domain-invariant latent spaces in an unsupervised setting. The learned domain-specific code is supposed to represent the properties of the source image which have no representation in the target domain. To this end, we train our network by maximization of a domain-specific variational information to learn a domain-specific space.

III. Objective

It is an approach to train the deep convolutional networks for Image-to-Image translation tasks. Unlike other GANs models for image translation tasks, CycleGAN learns a mapping between one image domain and another using an unsupervised approach. To transform one's face into cartoon-like character. For example, if we are interested in translating an image of a real person into an image of a cartoon character. We do not require the training dataset of a real person physically converted into a cartoon character. The way CycleGAN does it is by training Generator Networks to learn a mapping from domain X into an image that looks like it came from domain Y(and vice-versa).

IV. CycleGAN

A. Overview

CycleGAN is an approach to training image-to-image translation models using the generative adversarial network, or GAN model architecture. The CycleGAN is an extension of the GAN architecture that involves the simultaneous training of two generator models and two discriminator models.

In CycleGAN we treat the problem as an image reconstruction problem. We first take an image input (x) and use the generator G to convert into the reconstructed image. Then we reverse this process from reconstructed image to original image using a generator F. Then we calculate the mean squared error loss between real and reconstructed image. The most important feature of this cycleGAN is that it can do this image translation on an unpaired image where there is no relation exists between the input image and output image.

B. Architecture

In Pix2pix GAN model, Model G was trained to translate images from domain X to domain Y but it only works for paired images which are not always available . Whereas, Cycle

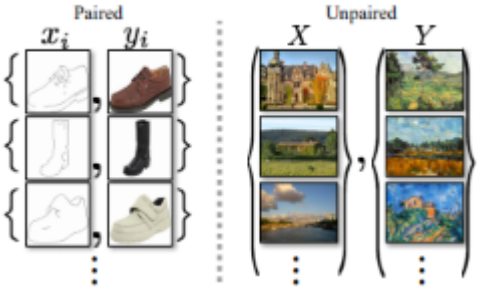

Fig. 1. Paired Vs Unpaired Image Translation

GAN does the same, but additionally it also trains a model F that translates images in the opposite direction - from domain Y to domain X. This introduces a cycle, hence the name, Cycle GAN.

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

where X is the input image distribution and Y is the desired output distribution.

Back-translation is a concept in which after translating from A to B, there is another translation process from B back to A in order to check how close is the original content compared to the one that went through the translation process. There is also an interesting story about Mark Twain and his work The Celebrated Jumping Frog of Calaveras County.

After discovering the French translation of his text and noticing how much of his signature humor and style were lost, Twain re-translated the French version word for word with intentional incoherency back into English with a new title The Jumping Frog: In English, Then in French, and Then Clawed Back Into A Civilized Language Once More by Patient, Unremunerated Toil to illustrate the problem of losing deep and subtle semantics during the translation process.

Like all the adversarial network CycleGAN also has two parts Generator and Discriminator, the job of generator to produce the samples from the desired distribution and the job of discriminator is to figure out the sample is from actual distribution (real) or from the one that are generated by generator (fake).

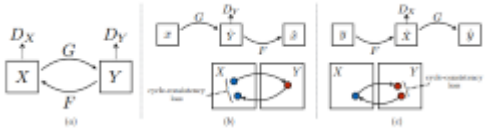

Fig. 2. CycleGAN Architecture

The CycleGAN architecture is different from other GANs in a way that it contains 2 mapping function (G and F) that acts as generators and their corresponding Discriminators (Dx and Dy): The generator mapping functions are as follows:

C. Defining Models

CycleGAN comprises two Discriminators(D_x and D_y) and two Generators(G_xtoy and G_ytox).

D_x — Identifies training images from domain X as real and translated images from domain Y to domain X as fake.

D_y — Identifies training images from domain X as real and translated images from domain Y to domain X as fake.

G_xtoy — Translates images from domain X to domain Y.

G_ytox — Translates images from domain Y to domain X.

To further regularize the mappings we used two more loss function in addition to adversarial loss. The forward cycle consistency loss and the backward cycle consistency loss .

The forward cycle consistency loss refines the cycle :

$$x - - > G(x) - - > F(G(x)) \approx x$$

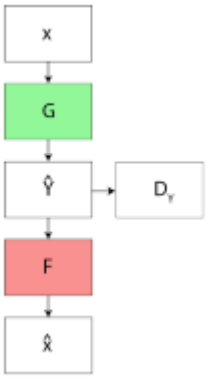

Fig. 3. Forward Cycle Consistency Loss

The backward cycle consistency loss refines the cycle:

$$y - - > F(y) - - > G(F(y)) \approx y$$

The objective consists of :

Adversarial loss: Adversarial loss is applied to both G and F. For generator G and discriminator DY, it can be formulated as:

LGAN (G, DY, X, Y) = $Eypdata(y)$ [log DY(y)] + $Expdata(x)$ [log (1 DY (G(x)))]

G tries to minimize this loss while discriminator DY tries to maximize it. This can be expressed as:

It is similar for F and DX and can be formulated as:

Cycle Consistency loss: The cycle consistency loss tries to capture how different is the reconstructed sample from the original sample. For example, how different is the original

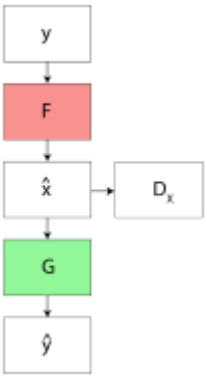

Fig. 4. Backward Cycle Consistency Loss

$$\min_{G} \max_{D_Y} \mathcal{L}_{GAN}(G, D_Y, X, Y)$$

story from Twain compared to the version that was translated from English to French, and then back to English?

The cycle consistency loss consists of:

Forward cycle consistency:

x $\rightarrow$ G(x) $\rightarrow$ F(G(x)) x

Backward cycle consistency:

y $\rightarrow$ F(y) $\rightarrow$ G(F(y)) y

The loss is then formulated as:

Lcyc(G,F) = Expdata(x) [F(G(x)) x1] + Eypdata(y) [G(F(y)) y1]

D. Generator Architecture

Each CycleGAN generator has three sections: 1. Encoder 2. Transformer 3. Decoder The input image is passed into the encoder. The encoder extracts features from the input image by using Convolutions and compressed the representation of image but increase the number of channels. The encoder consists of 3 convolution that reduces the representation by 1/4 th of actual image size. Consider an image of size (256, 256, 3) which we input into the encoder, the output of encoder will be (64, 64, 256).

Then the output of encoder after activation function is applied is passed into the transformer. The transformer contains 6 or 9 residual blocks based on the size of input. The output of transformer is then passed into the decoder which uses 2-deconvolution block of fraction strides to increase the size of representation to original size.

$$\min_{F} \max_{D_X} \mathcal{L}_{GAN}(F, D_X, Y, X)$$

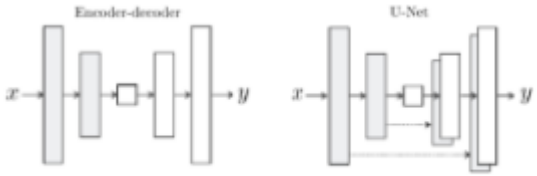

Fig. 5. Generator Architecture

E. Discriminator Architecture

The discriminator would take an image as an input and try to predict if it is an original or the output from the generator. Generator can be visualized in the following image.

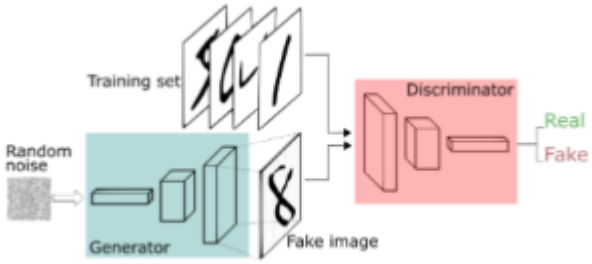

Fig. 6. Paired Vs Unpaired Image Translation

V. Implementation

We adopt the architecture for our common latent encoder, generator, and discriminator networks from Zhu and Park et al. . The domain-invariant encoders includes two stride-2 convolutions, and three residual blocks. The generators consist of three residual blocks and two transposed convolutions with stride-2. The domain-specific encoders share the first two convolution layers with their corresponding domain-invariant encoders, followed by five stride-2 convolutions. Since the spatial size of the domain-specific codes do not match with their corresponding domain-invariant codes, we told them to be the same size as the domain-invariant codes, and then concatenate them to create the generators' inputs. . For the discriminator networks we use 200 × 200 CycleGAN networks, which classifies whether 200 × 200 overlapping image patches are real or fake. We use Adam optimizer for online optimization with the learning rate of 0.0002.

VI. Experimentation Results

Our experiments aim to show that an interpretable representation can be learned by the domain specific variational information bound maximization. Visual results on translation tasks show how domain-specific code can alter the style of generated images in a new domain. We compare our method against baselines both qualitatively and quantitatively.

A. Qualitative analysis

We had a hard time deciding which GAN to use as there are many versions of DCGANS that could yield the same result of turning a person into something like a simpson character. After researching and getting the opinion of Prof. Matthew Dailey, we decided to go with Cycle GAN as it fit our requirements quite well. Looking at a few results that our Model had produced during training it is evident that it requires quite a lot of training.

We also had a hard time getting good quality human faces. Since it was very impractical to align each and every human faces photos that we had, we had to rely on CelebA dataset. This is why the quality of some of the images turned out to be a little underwhelming. However by the end, we ran 10 epochs and dropped our learning rate to 0.0000667 and we were able to get a somewhat satisfactory result. The model might have been able to perform much better had we found a better human faces dataset.

B. Quantitative analysis

We also had to rely on the CelebA dataset and Kaggle's simpson dataset as we needed a dataset that had thousands of photos of human faces as well as thousands of simpson faces. We also were not actually sure if the results would turnout the way they did, so we also tried to do the reverse of what we had initially set out to do, as cycle GAN does support it, and tried to make simpson characters' faces more human. We did have some luck in that department at first but it didn't really make any progress at the end. Since we found that our model did much better in regards to the converting human faces to simpson, we decided to go with what we had initially planned to do.

Fig. 7. This is how the first few epochs turned out.

Then we tested the model with a photo of Dr Chaklam from his facebook just to see how he would look if he was "simpsonized". Based on the photo above, one can say the model is still quite racist and might need to be trained more.

Since the results were quite satisfactory we felt that it might do better if we trained it even more so we tried training the model for 4 more epochs.

However, it turns out that Mark Twain might have actually been correct. The images turned out green by the end of epoch 12 which was a huge disappointment as we thought it might learn to make human characters look more like simpsons. It

Fig. 8. Epoch 10 showed some promise so we stopped here

Fig. 9.

turns out translating an image to another and then translating it back leads to very significant loss when it is repeated multiple times.

VII. Research Challenges and Future Work

While doing the research, and implementing the codes in the project we did face some challenges. Firstly, we had a bit of difficulty in deciding which type of GANs is best for our project between the Cycle GAN and Pix2pix CGAN. Secondly, due to the geometric inconsistency issue between the two domains, there was some extent of distortion seen in the final generated image. Lastly, we also had trouble managing the dataset for Simpson faces as well for the human faces. However, we were able to get both the required dataset in the end. But preparing paired sets of data is time-intensive and difficult. For-eg By paired set, what I mean is that we need to have an image in the same position or with the same background to be able to learn to map. For future works we plan to resolve the issue of distortions in the final generated image thereby, producing even high quality images. We also plan on using this technique and creating other different types of cartoons and characters.

VIII. Conclusion

CycleGAN has the ability to generate very good results for unpaired and paired data.In this paper, We presented a general framework for unsupervised image-to-image translation. We showed it learned to translate an image from one domain to another without any corresponding images in two domains in the training dataset. In particular, we observe that the improvement in performance increases as the dataset size decreases. As image-to-image translations models require a large amount of data, learning from unlabelled examples is of paramount importance. It is also interesting how learning the inverse mapping can be used to allow the model to learn on unpaired datasets. However, there are some limitations in this such as not getting very pronounced output images as hoped. We hope that the proposed approach gives rise to further research work.

IX. Citations and References

References

[1] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros. – Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks – ICCV 2017,https://arxiv.org/pdf/1703.10593

[2] Unsupervised Image-to-Image Translation Using Domain-Specific Variational Information Bound,https://proceedings.neurips.cc/paper/2018/file/c7c46d4baf816bfb07c7f3bf96d885 Paper.pdf

[3] Harmonic Unpaired Image-to-image Translation with Rui Zhang,Tomas Pfister and Jia Li,http://arxiv.org/abs/1902.09727

[4] Y. Li and S. Tang and R. Zhang and Y. Zhang and J. Li and S. Yan with Asymmetric GAN for Unpaired Image-to-Image Translation

[5] Ming-Yu Liu, Thomas Breuel, Jan Kautz – Unsupervised Image-to-Image Translation Networks,https://arxiv.org/pdf/1703.00848.pdf

[6] Generative Adversarial Networks, Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio

[7] Image-to-Image Translation with Conditional Adversarial Networks, Phillip Isola and Jun-Yan Zhu and Tinghui Zhou and Alexei A. Efros

[8] Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros

[9] Conditional Generative Adversarial Nets, Mehdi Mirza, Simon Osindero

[10] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. 2017.