Image-to-image translation: Progress and applications

Abstract

Image synthesis is uniquely position to give us insight into the working of deep learning by providing a visual representation for inner happenings. The culture of open-sourcing models in the learning community also allows non-academics access to these models. Therefore progress made and applications of these advancements is interesting to both technical and non-technical people.

Purpose – The purpose of this short paper is to reflect on the progress being made in the image-to-image translation from a less technical perspective. We compare three models pictorially (as appose to using metrics) from a layman perspective.

Design/methodology/approach – In order to reflect on the progress made we introduce the key components of popular techniques and the models used. After this we conduct some experiment in which we compare a select few examples on common data sets. We do so using an "easy to translate" data-set, followed by a more challenging one. We then comment on our findings, speculating as to why and when the model performs well and do not perform well.

In the appendix we perform further analysis constructing a data-set to strengthen evidence of our speculations.

Findings – We find that image-to-image translation techniques have made large improvements over the last 5 years. Previously supervised models could translate images across similar domains. Recent unsupervised models are almost able to translate across very different domain, some for which no current mapping exists. We found that current models can easily translate a human face into an anime face but the inverse translation proved to be more difficult.

Practical implications – Investigations like these can be performed with a mid-range desktop PC and open-source code. This allows individuals to take part in community driven research and/or use these models in application of their own. Due to the limited hardware we cannot reproduce the results of some other experiment.

Research limitations/implications – Image-to-image translation techniques have seen a large amount of growth in the last 5 years. We can anticipate more growth due to funding from private industries and interest from other related academic fields.

Keywords – Image-to-image translation, Generative Adversarial Networks, Deep-learning,

Paper type – Short paper, experiment.

steed@unist.ac.kr

1 Introduction

Deep image synthesis is uniquely positioned to pictorially communicate the challenges, progress, and possibilities of deep learning. It opens up the door for many gain a high level understanding of its inner workings with less technical experience/expertise.

Due to advances in hardware, software ecosystems, and research techniques these techniques have made a huge progress in the last decade, moving from largely linear/deterministic transformation to adaptive transformations with the ability to generate unobserved and sometimes convincing graphics.

In the last five years we have advanced very quickly from using Google's Deep Dream [2] as a toy example to industry ready application like NVIDIA's DSR and SKYLUM's Luminar.

1.1 Motivation and contribution

This paper aims to communicate the progress made in deep image synthesis over the last 5 years by comparing a few models.

In order for the reader to understand this we need to understand the core concepts of these models, the core components, challenges they face, and current applications. We address these questions in the remainder of this chapter. Next (Chapter 2) we select some well known models and highlight there concepts/contributions. We later (Chapter 3) use it to reason about experimental results.

1.2 Deep generative networks

Traditional generative models are based on machine learning techniques like GMM and HMM where deep generative models are rely on deep learning algorithms like DNNs, stochastic backprop, and Bayesian inference. Here we are specifically interested in GANs and VAEs.

VAEs learn the underlying probability distribution and generate new samples based on Bayesian inference (maximizing likelihood). GANs learns through adversarial training, where the generator tries to trick the discriminator. GANs offer a few advantages, most notably the ability to learn and model complex data distributions.

The idea of a GAN is based on a game of two competing agents. A generator G tries to generate realistic images and a discriminator D tries to distinguish real images from fakes. The generator and discriminator can be two algorithms (e.g. generator=CNN, discriminator=RNN) as long as the generator has the ability to learn the underlying distribution. This flexible definition lends itself to some interesting implementations.

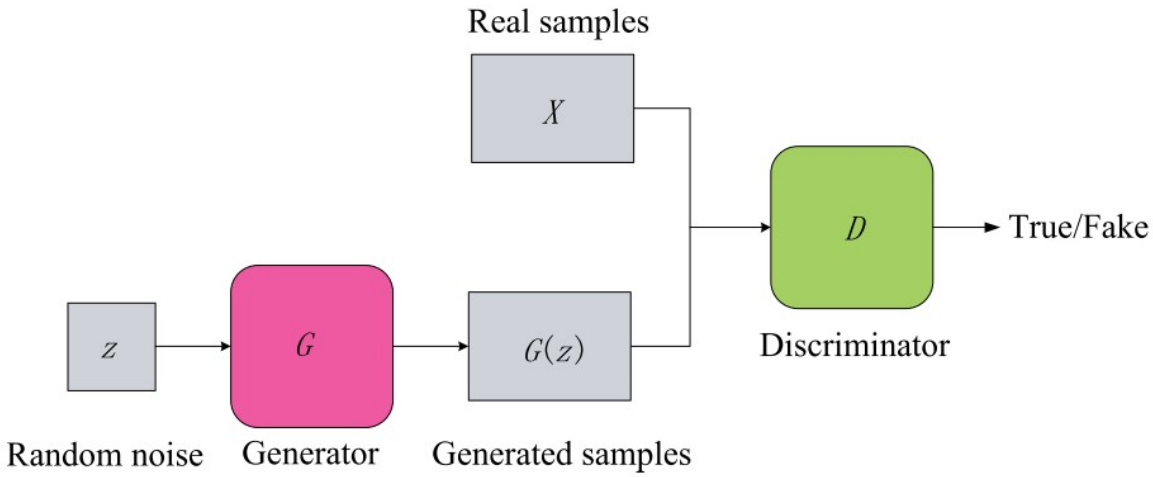

Figure 1: The general structure of Generative Adversarial Networks.[7]

1.3 Core components

Objective functions serve as the measure of distance between the real sample distribution and the generated sample distribution. This is minimized to achieve a realistic results. Selecting an appropriate objective function can alleviate the issues of mode collapse and vanishing gradient.

In GAN's we refer to this as adversarial loss. The discriminator and the generator have separate loss functions that are trained simultaneously to achieve Nash equilibrium. The discriminator aims to maximize probability of assigning a correct label, and the generator aims to deceive D e.g. minimize $(1 - D(G(z)))$. The original GAN objective function (shown below) used a cross entropy function for D and used a min max loss for G. In WGAN, the D is used to estimate the Wassertien distance from P_{data} to P_G in order to stabilize training. This can be further regularized using gradient penalty regression. Least squares can be applied to G in order to overcome the vanishing gradient problem. This translates to higher-quality images. The D has also been modeled as an energy function, assigning a higher to generated examples. Sometimes adding traditional L2 and L1 distances to the adversarial loss have also been reported to improve the image sharpness [3].

$$L_{GAN}(G, D) = E_y[logD(y)] + E_{x,z}[log(1 - D(G(x, z))]$$

The structure of GANs can also vary. GAN is based on multi-layer perceptions but here we are mostly concerned with Deep learning components for D and G. Some interesting developments include the use of a self-attention CNN to both D and G in SAGAN. Progressively growing (PG)GAN where G and D are progressively increased in size. VAE-GAN which uses an auto-encoder for the discriminator.

Evaluation metrics are used to quantitatively evaluate the performance of image-translations. The kernel inception distance provides a simple estimation of the discrepancy between the reference and estimated distribution. The Frechet inception distance represents the distance between the generated and real image distributions and is calculated using the inception network. The amazon mechanical turk uses human workers to evaluate images. The learned perceptual image patch similarity measures the diversity among generated images.

1.4 Challenges

1.4.1 Learning problem

Mapping images from one domain to another can have an infinite number of solutions and therefore is a highly ill posed problem. This is worsened by using unpaired image (unsupervised) translation. To address this problem we typically need additional assumptions [4]. This leads to some problems when training GANs, most notably training instability and mode collapse[1].

Although computer hardware has made substantial strides, especially due to the recent GPUs, training times can still be prohibitive [5][2], and much work needs to be done to reduce training times.

1.4.2 Supervised vs Unsupervised

Paired image sets refer to a set where an image in the X domain corresponds to a specific image in the Y domain. These typically perform very well [3] but the task of gathering these image pairs is often tedious and sometimes impossible.

Unpaired image-to-image translation drops this assumption making data collecting much simpler and allowing for more flexible use of the techniques. Now random images of domain X and Y can be gathered and translated.

1.5 Applications

Image-to-image translation learns the mapping from the input domain to the output domain. This can manifest itself in a number of applications. A good example of this is translating and image from summer to winter [6] . This is usually extending to a one-to-many translation (multi-modal) e.g. [4] →[1] (summer→ winter, spring, fall).We list some example applications here.

Super-resolution refers to the process of translating a low-resolution image to a high-resolution target image. This is the technique behind NVIDIA's DSR. GANs have been used in an end-to-end manner where the generator generates a high resolution image with a discrete discriminator.

Style transfer preserves the content but transfer to a specific style domain. Examples of these including converting photos to paintings [6].

---

1 all input images map to the same output image and the optimization fails to make progress.

2 StylesGan training (256x256) requires 3 days 8 hours with an NVIDIA DGX1 with 8 V100 GPUs. See training time table on GitHub.

Object transfiguring involves changing only a specific object in the scene, such as replacing a zebra with a horse [6]. This involves automatically detecting the object of interest and distinguishing it from other regions, like the background.

Medical imaging has also seen an increase in interest as of lately. Both in domain specific algorithms and in recent reviews on the topic [7], [8]. Some applications include (1) generating images in cases of limited quantity of positive cases or privacy issues and (2) detecting abnormal images.

2 Experiment models

Here we describe the models used in our experiment.

2.1 Historic overview of models

2.1.1 Pix2pix

Pix2pix [3] is a supervised image-to-image translation technique that is popular among academics, hobbyists, and artists. The reason behind its popularity is that it can be used as a general purpose solution, learns the loss function, and this technique yields in high quality results. This makes it easy to use for a variety of applications.

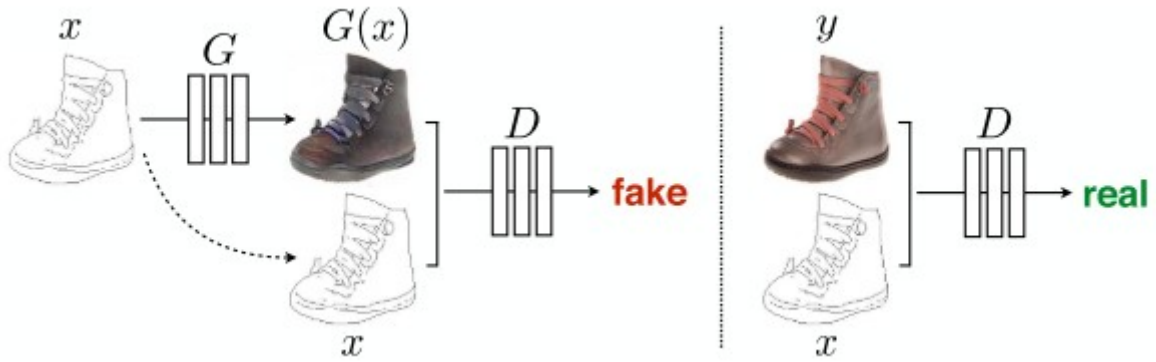

Figure 2: Traditional conditional GAN. D learns to classify between fake and real. G learns to fool D. Unlike unconditional GAN, both D and G observe input. [3]

They use a conditional GAN (CoGan) where the discriminator also observes the input. They also add the L1 distance between the real and fake to the objective function resulting in lower blurring[3].

$$G^* = arg\ min_G\ max_D\ L_{CoGAN}(G, D) + \lambda L_{L1}(G)$$

, where

3 The L1 term forces low-frequency correctness

$L_{CoGAN}(G, D) = E[logD(x, y)] + E[log * 1 - d(X, G(x, z))]$, and
$L_{L1}(G) = E[||y - G(x, z)||_1]$

This architecture diverges slightly from the traditional GAN as there are skip connections between multiple layers.

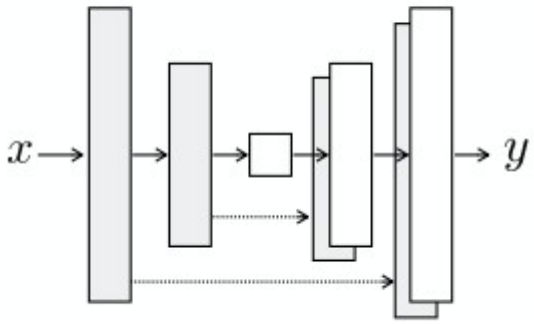

Figure 3: The architecture used in pix2pix, skipped connections between mirrored layers. [2]

Pix2pix yields high quality images but requires paired images. Gathering image pairs can often be time consuming, costly, and sometimes impossible. The unsupervised image-to-image translation techniques relax this assumption allowing the use of unpaired image sets.

2.1.2 CycleGAN

Having relaxed the constraints of paired images, unsupervised image-to-image translation is a highly ill-posed problem. CycleGAN [6] addresses constraining the translation to an inverse translation, called enforcing Cycle consistency.

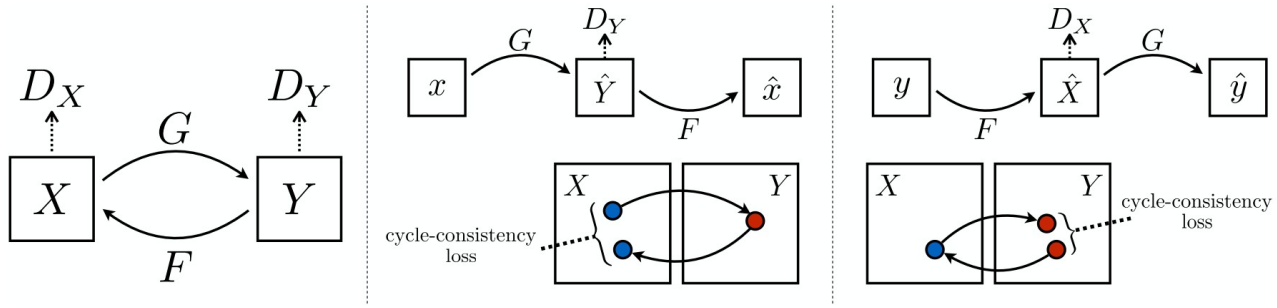

Figure 4: CycleGAN schematic showing mapping between domains and cyclic-consistency losses. [5]

In traditional GANs, the G translates from the input domain X to the output domain Y. We now include a F an inverse translation $Y \rightarrow X$. We can now compare the distance between the $G(X)\ and\ Y$ and $F(Y)\ and\ X$, incentivising a high quality reconstruction (or inverse mapping).

We now modify our objective function to include adversarial losses for mapping from $G(X \rightarrow Y)$, $F(Y \rightarrow X)$, and the cyclic consistency loss.

$$L(G, F, D_x, D_y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda L_{cyc}(G, F)$$

, where $L_{cyc}(G, F) = E[||F(G(x)) - x||_1] + E[||G(F(y)) - y||_1]$

This technique yields good results for simple datasets but is not as accurate as pix2pix. Also it assumes that a reconstruction will look the same, which is not the case in some applications. Therefore it does not perform well on some image-translations tasks.

2.1.3 UNIT

UNIT [4] uses a different approach by assuming a shared latent space between two GANs. Two auto-encoders encode different images to the same latent space, thereafter it is used an input to the two GANs. They show that the shared latent space assumption implies cycle-consistency but not vice versa. They also show that the latent variable z can be semi-constructed to a high level variables h after which you can apply specific low level features (e.g. sunrise, rainy, etc.)[4].

4 Disentanglement of high and low frequency features. This appears to be the basis of the multi-modal versions [1]

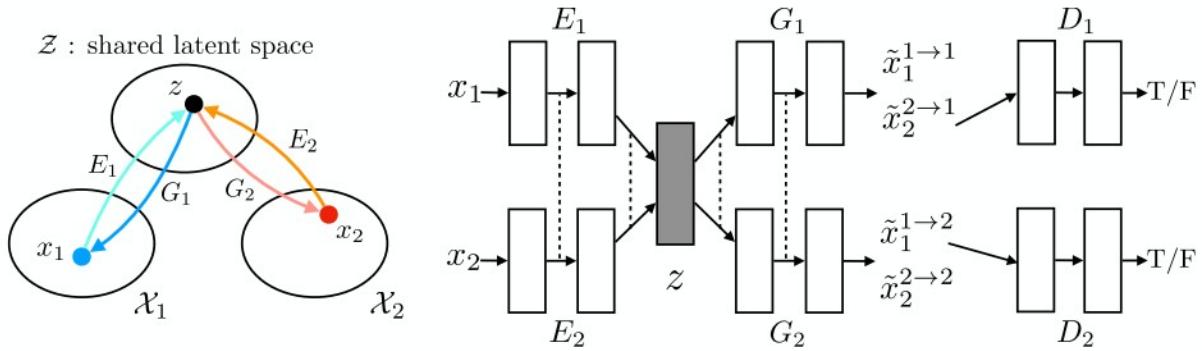

Figure 5: UNIT architecture has (1) 2 seperate GANs, (2) a shared latent space, and (3) and 2 auto-encoders. Note how the shared latent space resembles cyclic consistency. Also note the links between final layers of E_1 and E_2.[3]

The weights of the last few layers of the VAEs are shared to constrain the few layer responsible for extracting high-level representation of input images.

The objective function now takes the form summing the loss functions of the VAEs, the adversarial losses, and the cyclic loss[5].

$$L_{VAE1} + L_{GAN1} + L_{CC1} + L_{VAE2} + L_{GAN2} + L_{CC2}$$

This technique yield a slightly better result than Cycle-GAN.

2.1.4 CUT

More recently Contrastive Learning for Unpaired image-to-image Translation (CUT) was proposed [9]. Here the same group from Cycle-GAN have dropped the cycle assumption and instead compare patches of the image and output image. This assumes that the input and output should look (patch-wise) feature-similar[6], maximizing mutual information.

The architecture encodes both the input and and output images, using these as separate inputs to MLPs. Then a “query” and its corresponding “positive” is contrasted with other points called “negatives” using a cross-entropy loss.

5 Note the cyclic loss takes a dissimilar form from that shown in Cycle-GAN. Refer to the source for details.

6 If a leg appears on one patch in the input image, a leg should appear on the same patch of the output image.

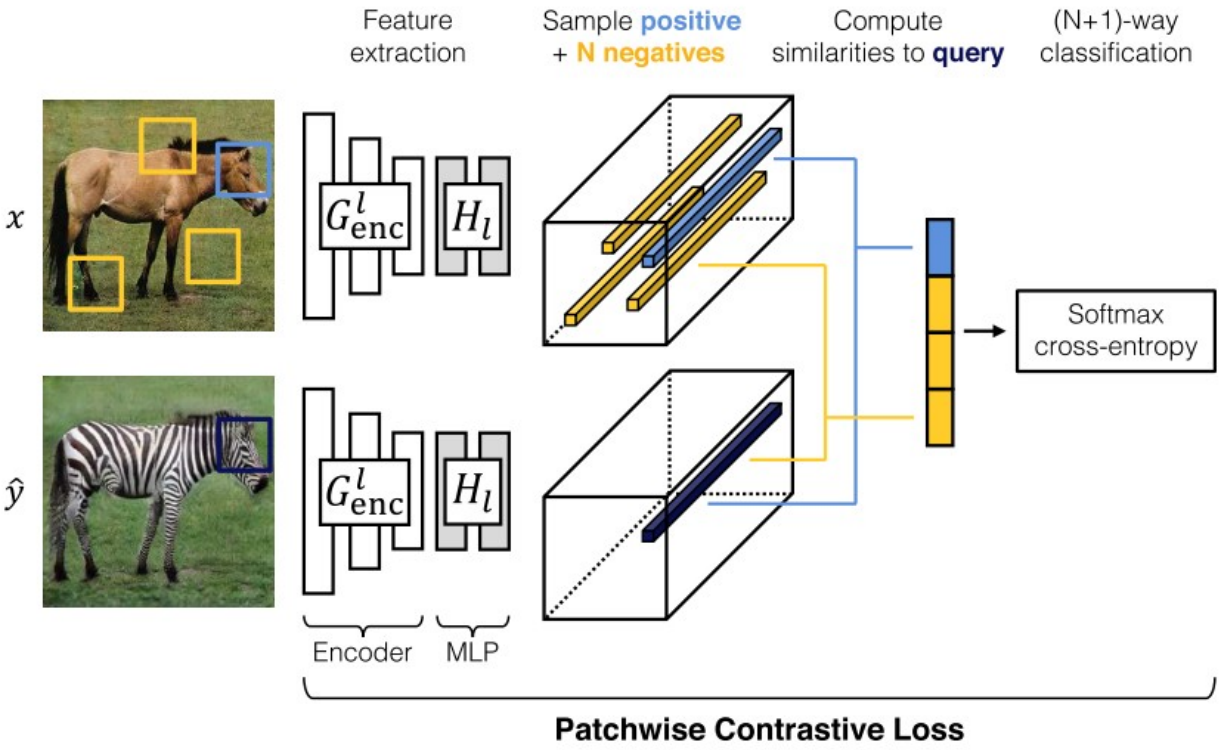

Figure 6: Contrastive Learning patch-wise loss. Both images are encoded into features. Patches are then queried and compared.[8].

The objective function now sums the usual adversarial loss with the patch-wise cross-entropy loss of input and out images.

$$L_{GAN}(G, D, X, Y) + \lambda_X L_{PatchNCE}(G, H, X) + \lambda_Y L_{PatchNCE}(G, H, Y)$$

Here a faster variant called FastCUT is also available by setting the $\lambda_Y = 0$ [7].

CUT reports training is 40% faster and 30% more memory efficient than CycleGAN. Furthermore FastCUT is 63% faster and 53% lighter while achieving superior metric to CycleGAN.

7 In Appendix B we investigate this model, but it was not compatible with out data-set and yields poor results.

3 Experiments

3.1 Data sets

We conducted our experiments on two well known training sets. Maps to photos, and Pub2OSR.

Maps to photos translates from an aerial photograph to a GPS map and can be regarded a benchmark data-set. It was commonly used for a long time as the input and output domains have similar local structures. It is a good data-set for comparing pixel wise translation and picture clarity, but does not involve much use of high level features.

Our second data-set involves translating from celebrity faces to landscape pictures. This is substantially more challenging as there is no obvious mapping between the two domains and features (e.g. nose, mouth → bridge, grass). Moreover this set is exclusively unpaired as there exists no obvious (at the time of writing this) map between landscape pictures and the human face.

3.2 MAPS comparison

We note that pix2pix performed much better than other techniques on this set, but that is largely due to it being a supervised training technique.

The fakes tend to be believable to the human eye when the true photo is hidden. This data-set is tested in all of the reference papers, finding a pixel accuracy of 0.85 pix2pix, 0.57 Cycle-GAN, 0.6 UNIT.

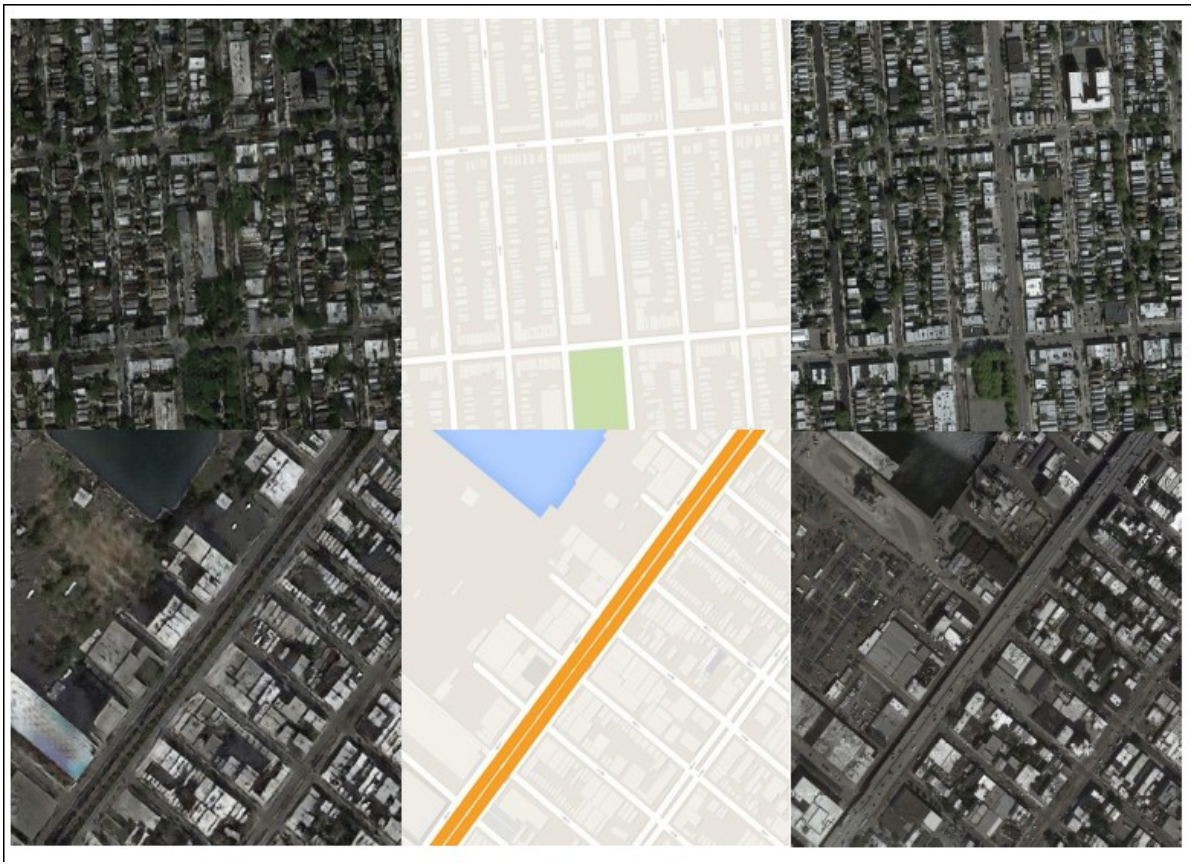

Figure 7: Cycle-GAN map-to-photo translation. The fake is shown on the left. Note how it looks believable to the human eye. When comparing it to the real photo however the difference can be seen.

We did notice however that in areas with high noise, where the street is not clear, the translator made some errors.

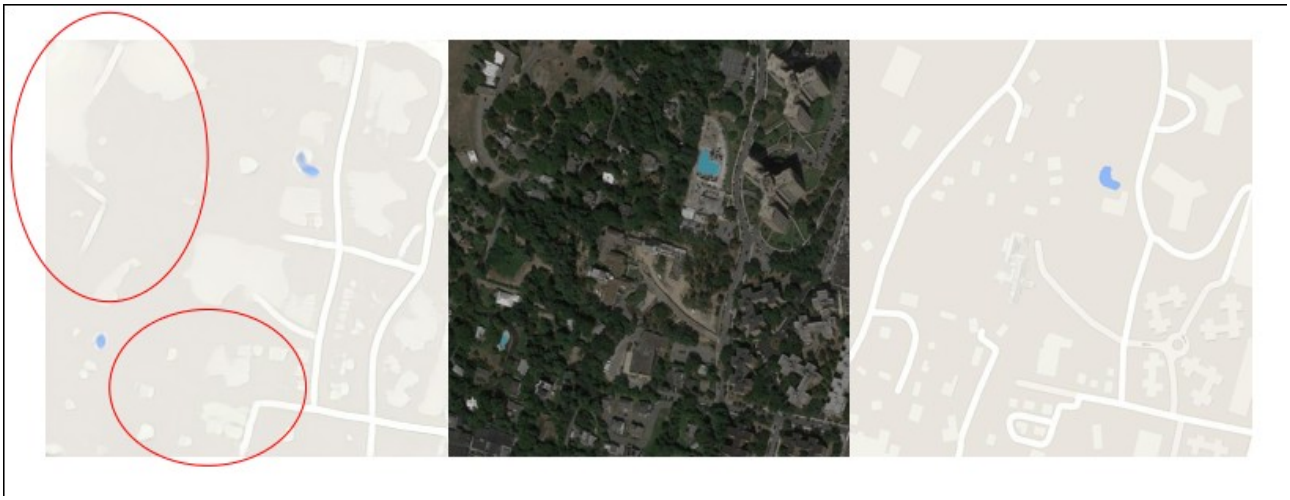

Figure 8:Cycle-GAN photo-to-map translation shows some errors in the places where the street can not be easily seen.

3.3 OSR2pub

This is an unpaired data-set, therefore pix2pix could not be used. Moreover because this set was so challenging it does not make sense to perform analysis using conventional metrics, instead we look at some of the produced fakes and try to reason about what is happening and where the techniques are failing.

3.3.1 Cycle-GAN

The first thing we notice is that Cycle-GAN does not do a good job with either fakes, that is these pictures are not convincing to the human eye. We can see that the reconstruction is good, only slightly blurry, indicating a small loss of information.

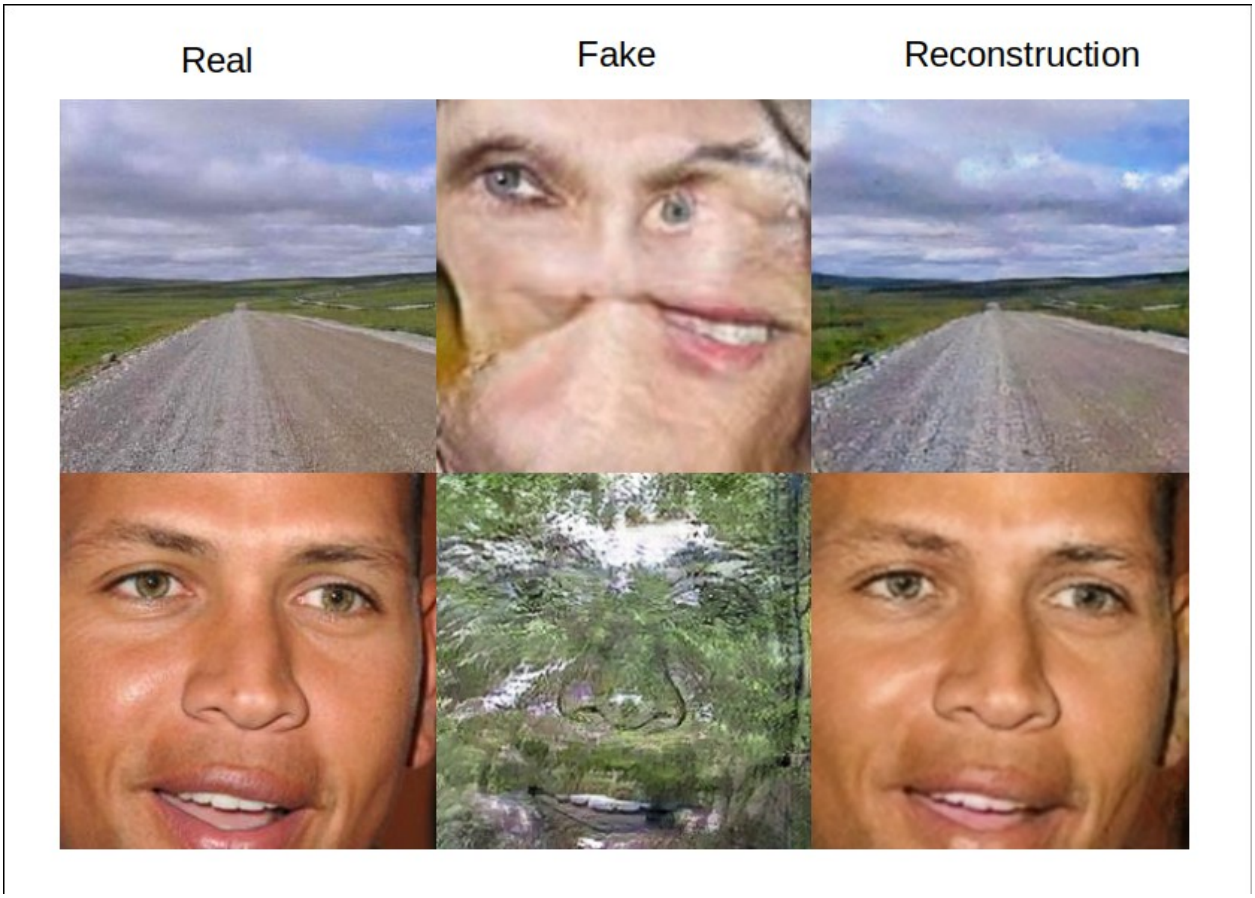

Figure 9: Cycle-GAN osr2pub translation is not good. Although a the reconstructed images are of high quality, the translated ones are not. Note the fake face has blue eyes and the real face has brown.

When looking at the fake face we can see that some of the features have been detected and translated, for example we can see that the eyes, skin, and mouth of the fake does not match that of the input[8]. The fake has two eyes, one mouth, and a nose. These are all signs that these features are learned by the model. The position of these features appears to be mixed, the eyes, nose, and eyebrows are well placed, where the mouth is not. The distortion of these features leads to a poor resulting fake face that fails to deceive anyone. From this we can hypothesize that the spacial location of features may or may not be well learned, and distortion of features is a problem.

Looking at the faked scene leads gives the impression that the textures (high-frequency) have been translated while preserving the features. This hints that maybe the high level features (nose, mouth) are being translated in one direction, and the low-level features (textures) are being translated in the opposite direct.

Overall this **cannot** be considered a useful/convincing translation.

3.3.2 UNIT

When comparing the fake faces, UNIT suffers from the same issues as Cycle-GAN. Again we see the facial features are being translate (eye, nose, mouse, etc.) but the spatial layout is not believable.

8 Input has hazel eyes, output has blue. Different skin tone and mouth position is also visible.

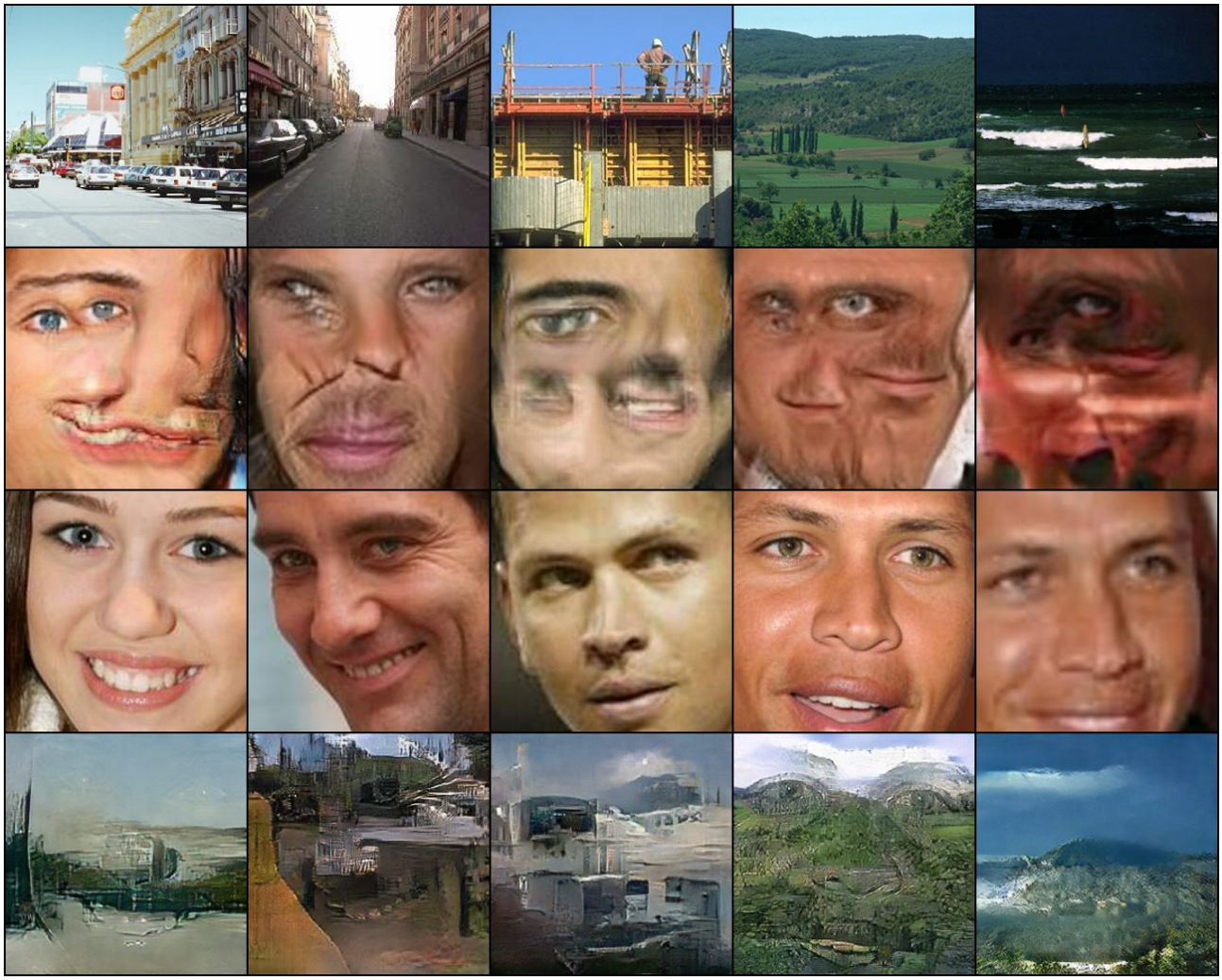

Figure 10: UNIT osr2pub tranlsation projects some features from the landscape onto the generated faces. Note how the landscape generated in the bottom right corner looks believable.

The scenery images however seem to be more believable. They appear to have translated the input face in most cases. The scenery looks somewhat believable and not highly related the inputs.

Note how the example on the far right (bottom) does did not have the face resonated on the output picture. This is likely due to the blurry quality of the input picture. We can hypothesize that the constrained weights of the encoder (final layers) are preserving he facial outline as high level feature, this is not exactly something we want in this translation. The blurry fake looks more believable than the others. Another figure show a similar result.

steed@unist.ac.kr

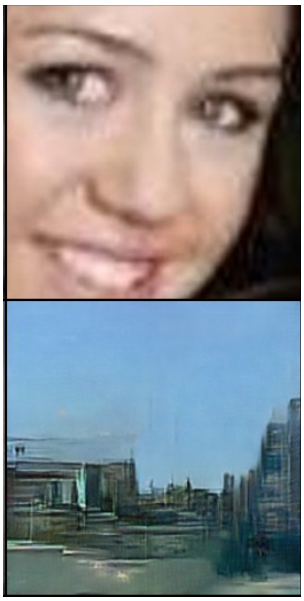

Figure 11: Another example of how a blurry face results in a believable (albeit blurry) landscape picture.

We are again left wondering if the high-level features are being translated in one direction (eyes, mouth, etc.) and the low-level features in the other direction (textures, etc.)

steed@unist.ac.kr

3.3.3 CUT

CUT produce the most realistic results yet. That said the algorithm is a one-way mapping and therefore had to be trained twice. Also the epoch was increased from 200 to 400, this negates its advantages of faster training time.

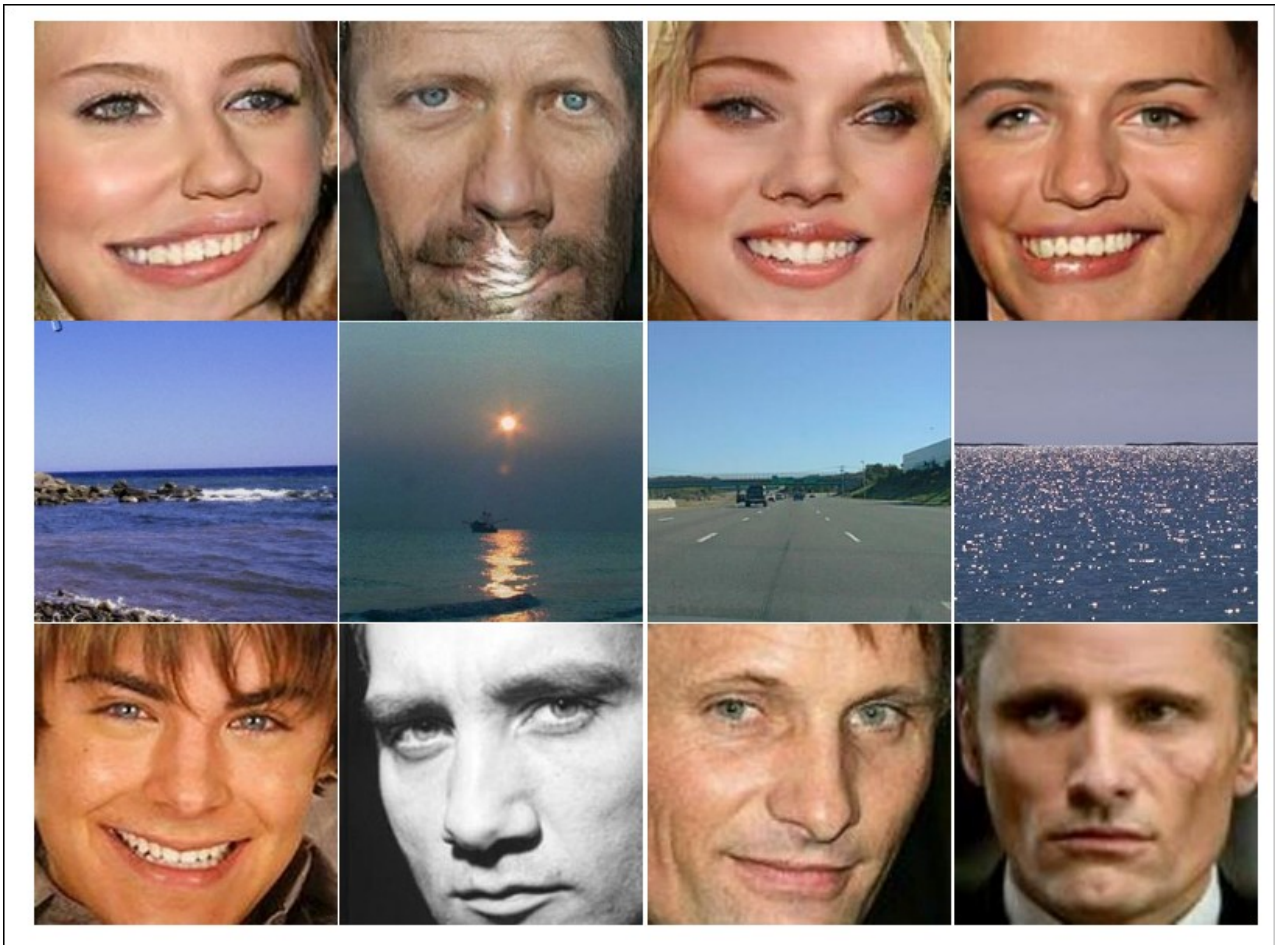

Figure 12: CUT does a good job of generating faces from landscape photos (osr2pub). We see some deformation, but the results are much better than previous models.

We can see that CUT does a much better job of translating the landscapes to faces. This is likely due to ensuring patch-wise information similarity, but this is also the reason why it fails to convince us.

We find that the generated facial features (mouth, nose, and eyes) suffer from warping and distortion. This is likely due to the limitations of patch-wise information similarity, not knowing the 3D representation of the feature. Therefore it tries to place a mouth in the respective patches, but does so by placing an elongated feature that is inconsistent with the current face orientation. This issue has been investigated in DiscoGAN [10].

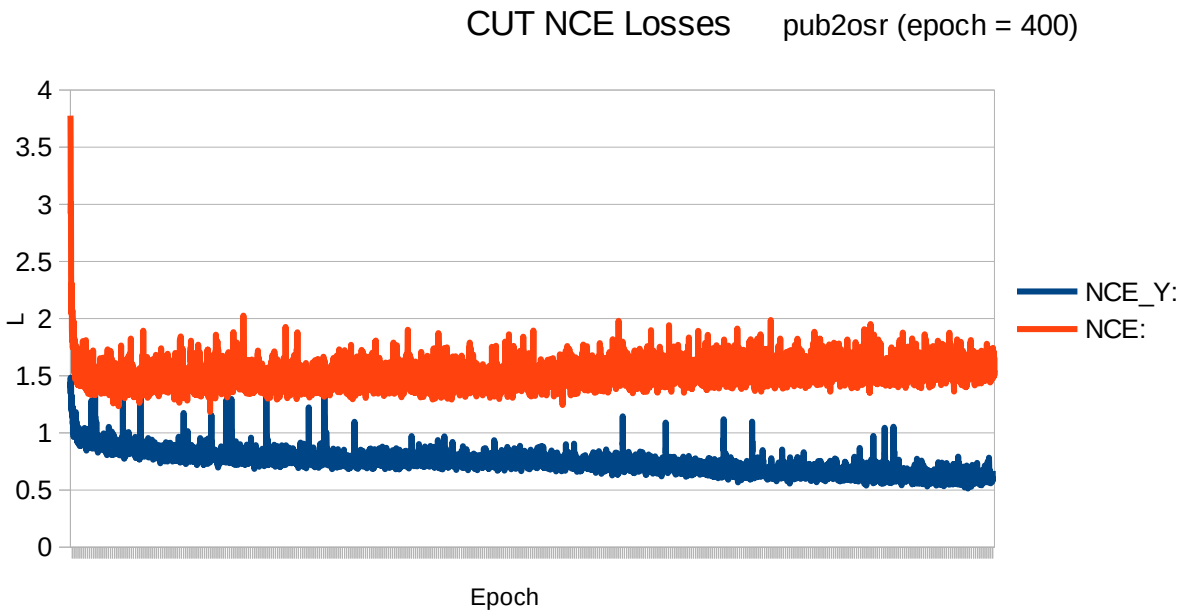

Figure 13: We see the value of the NCE_Y loss continues to drop linearly. We may have benefited from additional training time. Moving average smoothed.

Upon inspecting the objective function it appears that the translator would have benefited from additional training time. It seems the NCE_Y loss is still decreasing linearly. This would be unfair to the other algorithms being compared. Still we should remember this when comparing with other recent algorithms.

steed@unist.ac.kr

4 Discussion

4.1 Experiment

4.1.1 Maps

All the models tested here performed well on the maps data-set. So much so that the findings were not very interesting. The pix2pix performed substantially better than the unsupervised models according to the per-pixel accuracy, but this is a very limiting metric to use. It is important to remember that unsupervised models are trying to solve a different problem. Put simply they have the potential to produced believable models, by using unpaired sets of data, allowing them to interpolate where no data/map exists. Therefore metrics that encourage similarity (like per-pixed accuracy) are not well suited. Amazon Turks is a much better suit and here Cycle-gan is noted to perform well [6].

It would not be a stretch to say that a product could use this kind of image translation to generate photos from a GPS system (GPS→ photos), particularly in areas of little interest (e.g. highways). The opposite translation (photos→ GPS) is not practical at this time as small error can lead to frustrated drivers.

4.1.2 OSR2Pub

The OSR2PUB data-set provided an interesting experiment. Firstly because there exists no obvious mapping between landscape photographs and human faces. Therefore this is a difficult translation to make and small a mounts of progress can be visually interpreted.

We saw that none of the models were able to fool a human, but in the case of CUTs it may be that there was not sufficient training time. Both Cycle-GAN and UNIT was able to learn the facial features (eye, nose, mouth, etc.) and place them in positions that resembled the landscape. There is little connection between a landscape layout and the human face, for this reason these results did not fool a human inspector/discriminator.

Landscapes were created more believably. Still there was a high resemblance to the face in the landscape. We note that when given blurry pictures of a face, UNIT was able to produce believable (albeit blurry) picture of the landscape. This is interesting because image sharpness is sought after in literature[9], but in this case it comes at the cost of believably.

CUT performed better than its predecessors on this data-set. We speculate this is due to enforcing patch-wise information similarity, it was able to keep features close together. Therefore having a “high correctness of feature position”. This was more convincing that previous results. It still suffered from “a poor correctness of deformation”. We suspect this may be because features are learned in the pixel space and deform unnaturally. If we could somehow map these features into a

9 It is mentioned that often L1 and L2 distances are added to the objective function to increase image sharpness.

3D space and restrict it to a 3D surface, there deformation would be more believable. We speculate from the losses that this model could have benefited from additional training and would be a good future exercise. We also perform some additional experiments on facing translation in the appendix, please see those for additional explanation[10].

Comparing the results from the previous models to CUT, we see that lots of progress has been made and the generated pictures resemble a human face.

4.2 Limitations

4.2.1 Metrics

We notice that the limited amount of metrics and the migration of supervised metrics limit our perception of progress. Similarity and pixel accuracy are only relevant in existing mapped domains.

Unsupervised image translation has the potential to answer very different questions and therefore requires metrics, such as how believable if the translation. Or more abstractly, how pleasurable is it too look at the generated images. These metrics may encourage the creating of applications not possible before.

The deconstruction of these metrics can also help, for example we saw a “high degree of positional correctness of features” in CUT made the resulting images much better to look at. We also saw that it suffered from a “low degree of deformation correctness”. These ideas should eventually find there way into evaluation metrics and objective functions.

4.2.2 Training time

We used a Nvidia RTX2080Ti for our experiments and most models took upward of 2 days to train (osr2pub). We notice that StylesGAN and StylesGAN2 produce high quality human faces but these results are not reproducible due to hardware limitations. These models run for weeks on hardware that is out of reach of most academic institutions. It would not be fair to compare the results generate here, with that of the aforementioned models.

4.3 Conclusion

We showed how we have been able to generate believable human faces using image-to-image translation over the last half decade. In this specific case we showed we can generate human faces from seemingly unrelated objects. We also highlight that these techniques can be easily communicated with little prior mathematical knowledge. All of this work has been done with freely available code and a mid-range home PC. This shows that many people can access these tools and build viable products using this technology.

10 In summary we conclude that the human face has lots of information (a high level of detail), and having to infer this causes problems. However if we translate from the human face, to a simpler domain, (faces→ anime) these issues are mitigated.

We see evaluation metrics and application of this technology is still somewhat limited, but anticipate these will be quickly addressed due to increasing hardware performance and the interest from multiple industries/backgrounds.

We can attribute this progress to a large amount of private funding[11], a wide variety of applications, open code, and community driven research. Moreover image based deep learning application are uniquely positioned to: provide insight as to how these machines work and interest the public in scientific developments.

11 Particularly NVLABS (NVIDIA)

Bibliography

[1] X. Huang, M. Y. Liu, S. Belongie, and J. Kautz, “Multimodal Unsupervised Image-to-Image Translation,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11207 LNCS, pp. 179–196, 2018, doi: 10.1007/978-3-030-01219-9_11.

[2] A. Mordvintsev and Mike Tyka, “DeepDream - a code example for visualizing Neural Networks.” https://web.archive.org/web/20150708233542/http://googleresearch.blogspot.co.uk/2015/07/deepdream-code-example-for-visualizing.html (accessed Dec. 17, 2020).

[3] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 5967–5976, 2017, doi: 10.1109/CVPR.2017.632.

[4] M. Y. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 701–709, 2017.

[5] T. Karras NVIDIA and S. Laine NVIDIA, “#StyleGAN - A Style-Based Generator Architecture for Generative Adversarial Networks Timo Aila NVIDIA,” *Cvpr 2019*, 2019, [Online]. Available: https://github.com/NVlabs/stylegan.

[6] J. Zhu, T. Park, A. A. Efros, B. Ai, and U. C. Berkeley, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks.”

[7] L. Wang, W. Chen, W. Yang, F. Bi, and F. R. Yu, “A State-of-the-Art Review on Image Synthesis with Generative Adversarial Networks,” *IEEE Access*, vol. 8, pp. 63514–63537, 2020, doi: 10.1109/ACCESS.2020.2982224.

[8] S. Kaji and S. Kida, “Overview of image-to-image translation by use of deep neural networks: denoising, super-resolution, modality conversion, and reconstruction in medical imaging,” *Radiol. Phys. Technol.*, vol. 12, no. 3, pp. 235–248, 2019, doi: 10.1007/s12194-019-00520-y.

[9] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, “Contrastive Learning for Unpaired Image-to-Image Translation,” pp. 319–345, 2020, doi: 10.1007/978-3-030-58545-7_19.

[10] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, “Learning to discover cross-domain relations with generative adversarial networks,” *34th Int. Conf. Mach. Learn. ICML 2017*, vol. 4, pp. 2941–2949, 2017.

Appendix A: additional data-set investigation.

For interest we construct our own data-set, anime faces to public figures (anim2pub). We hoped that the similarity between human faces and animation faces would result in more “believable” generated figures.

Figure 14: : CUT translates anime faces to public figures (anim2pub) better than osr2pub. Still the distortions mentioned before occurs.

We found slightly better performance that than the osr2pub data-set. We find that some these faces (anim2oub) look more believable, but suffer from the same distortions as osr2pub.

On the other hand translating from the human face to anime face performed much better. Even from a low epoch we see believable translations.

Figure 15: CUT translates human faces to anime faces really well. At epoch 200 we are starting to find believable translations.

The believably of the fakes increase with training time. Yet the test results show little variation, with all the fakes having purple hair. This could be a sign of over-training or that mode collapse is approaching. The data-set did not contain an excessively large fraction of purple haired anime, nor was the last training epoch mostly purple haired.

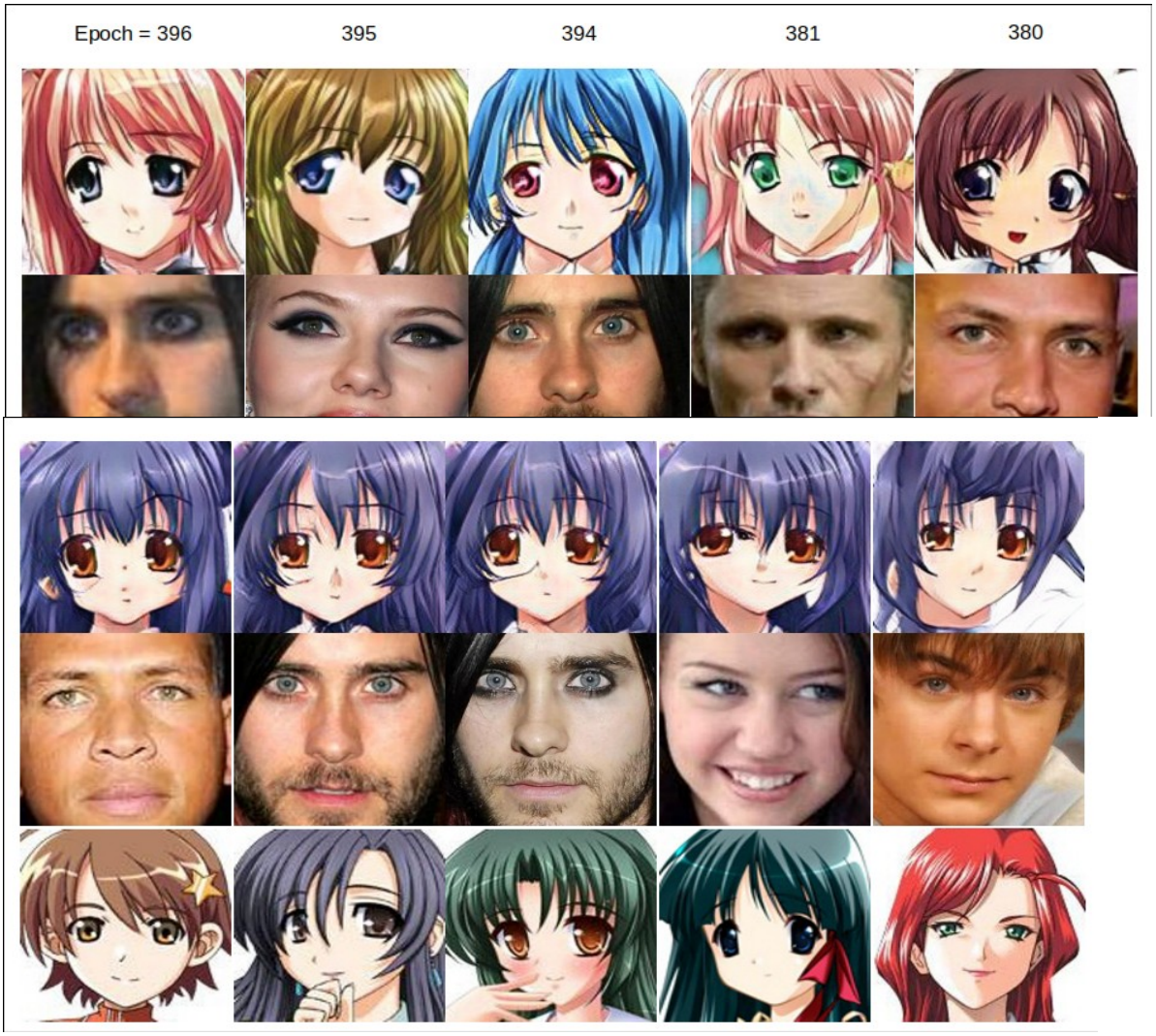

Figure 17: CUT produces convincing fakes during tests. Note how all similar the results are. All fakes have similar faces and hair color. This limited amount of variation is undesirable and could be a result of mode-collapse.

We also use the grumpy-cat data-set suggested by the author. This set results in a much more believable result. This could be because grumpy cats only have one facial expression “grumpy”, where human faces have many e.g. a smile distorts the mouth in a specific way. This makes a human face translation more difficult as it may have generate a number of feature states (e.g. smiling mouth, sad mouth, happy eyes, sad eyes, etc.)

Figure 18: CUT translates grumpy cats really well. In this data-set the domains resemble each other.

From these results it appears that generating human face has much more detail/information than cats or anime. Hence the generator has to infer more unknown information when generating a human face. It when this infered information is not constrained we find results like those in UNIT and Cycle-GAN. Although CUT performs better it does not do so convincingly.

When translating from a domain of high detail/information to low (pub2anim/cat2grumpycat) the problem of high detail/information inference is mitigated.

Appendix B: FastCUT results

Attempting to use FastCUT resulted in no translation at all. Occasionally it did flip the picture along the vertical access, but the results were of little worth and therefore was not explored further.

Figure 19: Results of training FastCUT.

steed@unist.ac.kr