Deep Vison
Project Report

Sketch2face

by

Christopher Lüken-Winkels and Richard Häcker

12.08.2020

Github: https://github.com/R-Haecker/sketch2face

Abstract

In our project the goal was to crate a translation between image domains. As concrete application we present a model to generate images of real faces from simple sketches using a modified Version of a Cycle GAN being based on a Variational Autoencoder.

Contents

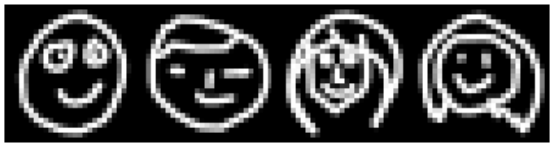


Figure 1: Examples of the Quickdraw dataset

1 Data Set

To train a network to create realistic images of faces from an input of sketches we first needed datasets to represent the different domains. To represent the sketch domain we chose the Quickdraw dataset [1] by Google and for the face domain we used the Celeb A dataset [2].

1.1 Sketches

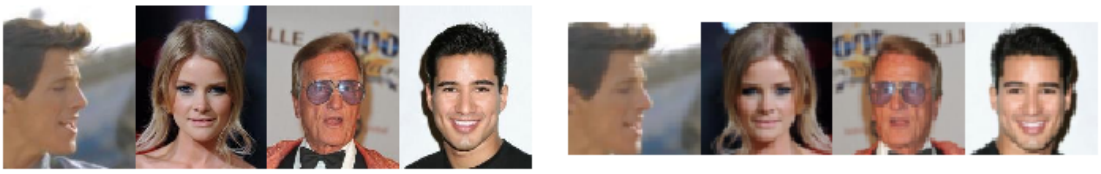
As sketch images for the training we wanted to use a dataset that contains only simple images such that everybody can produce input images that are a suitable input for the network. This lead us to the Quick Draw dataset which contains 50 million images displaying drawings over 345 categories which are created by 15 million users, each drawn in a few seconds. The category "face" which we used contains about $150,000$ grayscale images, each having a resolution of 28×28. As one can see in the examples in Figure 1 this results in a large variety of very simple face sketches.
For the letter Network architecture we prefer images with a size of powers of 2 so the images are rescaled to be of the size of 32×32.

1.2 Faces

The images of real faces are drawn form the Celeb A dataset. It contains around $200,000$ images of the faces from about $10,000$ celebrities in different orientations, settings and exposure in RGB and a resolution of 178×218. Additionally landmarks and 40 binary attributes are assigned to each image which did not use. The images are available in an aligned way such that a certain point of the faces is always in the same position. We used these images as the preprocessing implies less variation within the images simplifying the task for the network. Again this results in a large variety of appearances of faces.
For easier processing in the network we cropped the images around the center to get a quadratic shape and additionally rescaled the crop to either 32×32 or 64×64 as seen in Figure 2. The low resolution is used for first testing the approach before using a higher one.

1.3 Preprocessing

Besides cropping we additionally applied a normalization on the images resulting in values between -1 and 1 for each pixel in each channel. As augmentation we



(a) Raw data

(b) Preprocessed

Figure 2: Exapmles of the Celeb A dataset before and after preprocessing including a centered crop, rescaling (to 64×64)and random horizontal flipping

introduced random horizontal mirroring for the images effectively doubling the size of the datasets.

2 Approach

The final goal was to train a model which can generate real looking faces from simple sketch images. Thus we need a network to learn a connection between the two domains form unpaired data, a task that is approached in a so called Cycle GAN fist presented by Jun-Yan Zhu et al. [3].

2.1 Cycle GAN

Here, not only a translation from one to the other domain but simultaneously the opposite direction is learned. In our case this means that we train two generators: one transforming sketch to face images and one producing sketch from face images. This is then used to enforce the so called cycle consistency of the connection between the domains which means, that a transformation into the other domain and back using the two generators should result in the same image. Thus the information of the input is to be conserved in the translation resulting in a more solid connection between the domains. For the generated images to fit to their domain an additional adversarial loss is used which is produced by two discriminators. Examples from the original paper can be seen in Figure 3. One can see that the generated images fit well into their domain and in this case especially the shape of the objects is well preserved by the translation.

These are attributes of the translation we also try to achieve in our work. We first of all want the generated faces to look real and additionally want to keep some attributes of the input within the translation. For example if the input sketch contains a drawing with long hair we would like this to be contained in the generated face. Yet we do not want a spatial consistency as in the original paper as for example the shape of real faces differs from the form of the outline seen in the sketches. To tackle this problem we used a different generator model. In the original paper either a U-Net [4] or residual blocks are used which easily preserve the spatial information of the input. We replace this by a Variational Autoencoder (VAE) [5] which breaks down the spatial information to then reconstruct an image from the latent representation. With the cycle consistency loss included we expect the output to contain attributes of the input.
Our first apporach is to train such a model from scratch. Yet the task of generating and reconstructing realistic images of faces is already hard, thus our second approach is to pretrain the different parts of the network to achieve better results. This approach also yielded the possibility to first familiarize ourselves with the task of generating and reconstructing images of real faces.

Figure 3: Cycle GAN examples

2.2 GAN's and Pretraining

DCGAN To start of we tried to generate real faces and real sketches separately form noise using the baseline of a Deep Convolutional Generative Adversarial Networks (DCGAN) [6]. We sampled from a random latent representations to generate images which afterwards classified by the discriminator. The discriminator is trained to differentiate between images generated by our model and input images from the data set. The generator has to fool the discriminator to classify the generated images as input images. During training both networks improve over time and the generator will at last create similar images alike ones from the data set.
Eventhough the model was able to mimic certain features of images of human faces, the training took a long time and there was still room for improvement. This lead to the more sophisticated Wasserstein GAN (WGAN) [7].

WGAN The general idea of the WGAN is the same as before. A generating network is trying to fool the disriminator and thus learns to produce realistic images. Yet the WGAN has three important differences compared to the DCGAN.
First of all we use a gradient penalty for the discriminator to only slightly change the weights at every step. This way the generator can fulfill a more consistent task as the calculation of the adversarial loss does not change as quick, which should result in a more robust training.
The second difference is that the generator only gets updated every 5th step during training. This together with the overall smaller steps leads to a more robust disriminator, that is more specialized for the current state of the generator. The resulting more precise classification leads to a better training of the generator.
The third difference is that no activation is applied to the output of the discriminator. It is trained to output a value as high as possible for real and as low as possible for fake images. As this model showed improvement to the DCGAN we chose it as basline for our decoder.

Adding an encoder For our final goal we need to encode images and extract information from sketches to at last generate images of real faces. We used the WGAN baseline for the decoder but introduced a VAE as the generative network. We choose a VAE instead of an Autoencoder for future possibilities to sample from the latent space and to generalize better to draw our own sketches.
To pretrain all parts of the final Cycle GAN model we need to train encoder, decoder and discriminator for each domain. To achieve this we train networks to reconstruct

faces from faces and sketches form sketches, to later stitch the parts together. This method of pretraining allows an easier control of the training as the expected outcome is paired with the input. For a Cycle GAN model the transformation from one to the other domain has to be learned in an unpaired fashion which is harder to evaluate. Additionally it is easier to include a new dataset as the pretraining does not require a connection between the domains and only the new encoder and decoder have to be trained from scratch.

3 Architectures and training

3.1 GAN

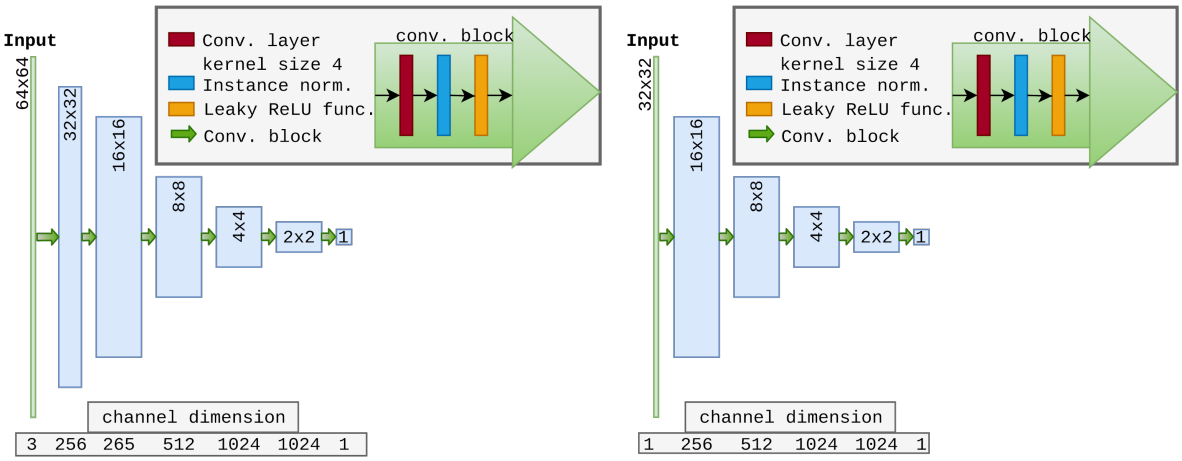
We first started to pretrain our pure reconstruction models. The loss of the generator is composed of the reconstruction loss being the L2- or L1-loss between the input and output image, the adversarial loss being the negative output of the disciminator and KLD-loss [8] in the latent representation. Each contribution is weighted to control the influence of the respective loss.

$$\mathcal{L}_G(x, z, y) = \lambda_{rec}\mathrm{L}_1(x, y) - \lambda_{adv}\mathrm{D}(y) + \lambda_{KLD}\mathrm{KLD}(z) \tag{1}$$

where x denotes the input, z the latent space representation, y the prediction of the model and D is the discriminator. The discriminator loss is the same as for the WGAN with one contribution from the real, one from the generated image and one from the gradient penalty.

$$\mathcal{L}_D(x, y) = -\lambda_{real}\mathrm{D}(x) + \lambda_{fake}\mathrm{D}(y) + \lambda_{gp}\mathcal{L}_{gp} \tag{2}$$

Both networks are trained using the ADAM-optimizer [9]. For the Generators ReLU is used in the intermediate layers with a final hyperbolic tangent as activation to produce outputs within the range of our normalization. For the Discriminator leaky ReLU is used. In both networks Batch-Normalization is applied. The architectures can be seen in Figures 4, 5 and 6.



(a) face (b) sketch

Figure 4: Architecture of Discriminators

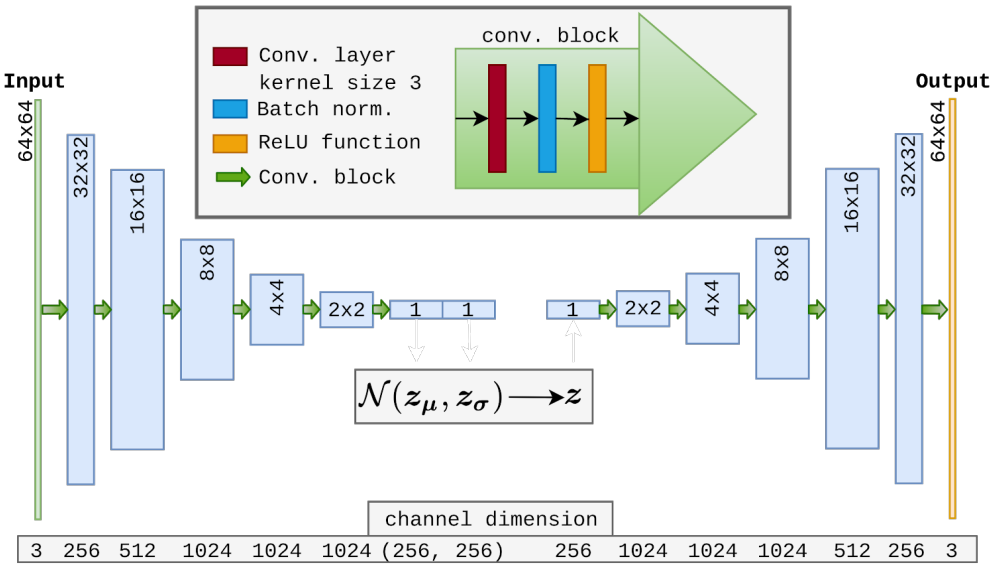


Figure 5: VAE Face Generator

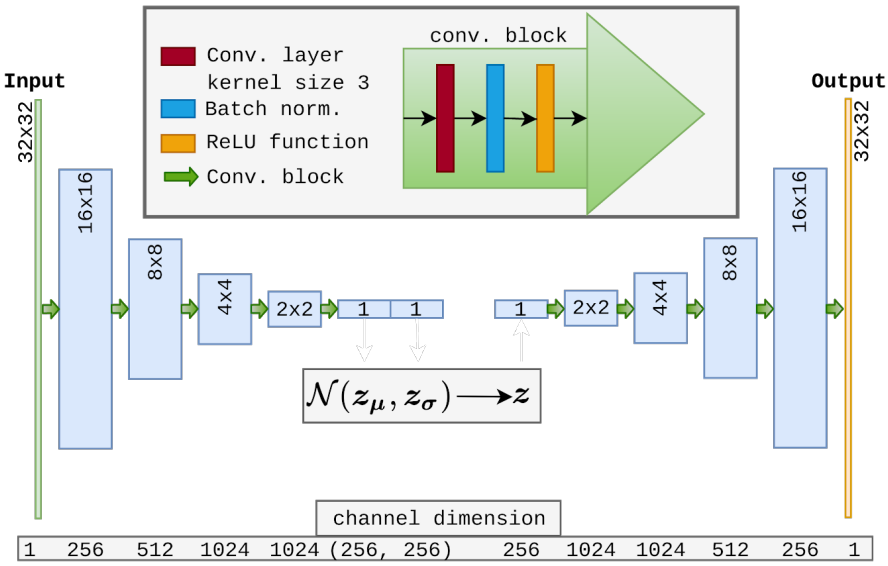


Figure 6: VAE Sketch Generator

3.2 Cycle GAN

Each step of the training of the Cycle GAN consists of the propagation of one face and one sketch image through the full network. The pipeline for a sketch image (sketch cycle) is shown in Figure 7. The face cycle works analogously using another discriminator. First sketch image a is encoded to z_a and decoded in the first VAE model to produce a fake face image b_{fake}. The outcome is then the input to the discriminator yielding the adversarial loss for the fake face. The generated image is then propagated though the second VAE producing an image of the sketch domain a_{rec}. Here we apply an L1-loss between the input image and the now reconstructed

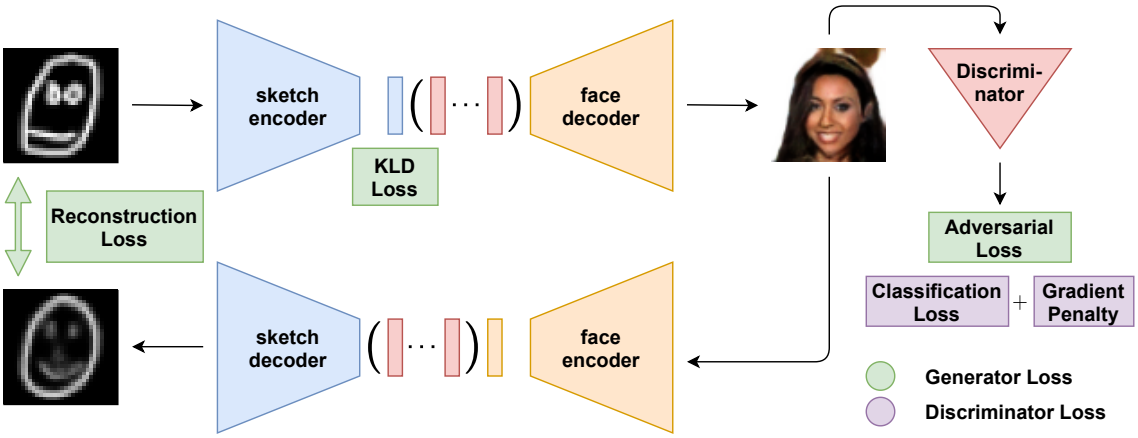


Figure 7: Pipeline of a sketch image through our Cycle GAN model either containing linear layers in the bottleneck (red boxes) or not. Face cycle works analogously starting in the face domain and using another discriminator.

image. Within the latent space the KLD-loss is calculated. The same happens for the forward pass of the face image b. It is transformed to a sketch a_{fake} where the adversarial loss is calculated and back to a face b_{rec} which is used for the reconstruction loss. The combined loss is then given by:

$$\begin{aligned}\mathcal{L}_G = \lambda_{rec}\mathrm{L}_1(a, a_{rec}) - \lambda_{adv}\mathrm{D}_b(b_{fake}) + \lambda_{KLD}\mathrm{KLD}(z_a) \\ + \lambda_{rec}\mathrm{L}_1(b, b_{rec}) - \lambda_{adv}\mathrm{D}_a(a_{fake}) + \lambda_{KLD}\mathrm{KLD}(z_b)\end{aligned} \tag{3}$$

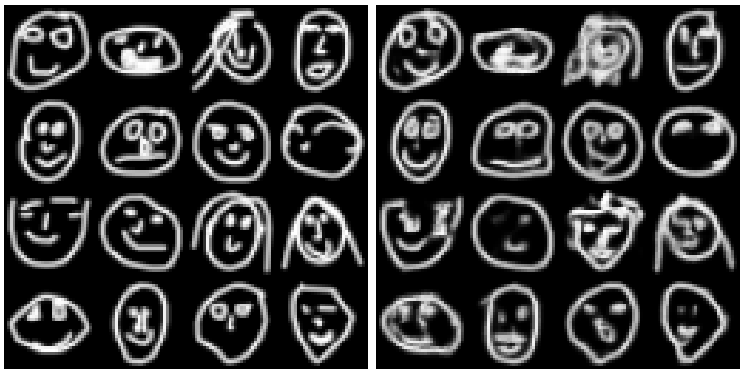
with D_a and D_b denoting the Discriminator in the sketch and face domain respectively.

The discriminators are updated separately as for the WGAN:

$$\mathcal{L}_{D,a}(a, a_{fake}) = -\lambda_{real}\mathrm{D}_a(a) + \lambda_{fake}\mathrm{D}_a(a_{fake}) + \lambda_{gp}\mathcal{L}_{gp}) \tag{4}$$

$$\mathcal{L}_{D,b}(b, b_{fake}) = -\lambda_{real}\mathrm{D}_b(b) + \lambda_{fake}\mathrm{D}_b(b_{fake}) + \lambda_{gp}\mathcal{L}_{gp}) \tag{5}$$

Again we used the ADAM optimizer and the same activations as for the GAN.
For the model trained from scratch we used the encoder and decoder of the VAE models shown in Figure 6 and 5 and the discriminators for the domains are the same as in Figure 4.
For our networks using the pretrained VAE's we added linear layers in the latent dimension as indicated by the red layers in Figure 7. These are introduced for the network to learn a translation between the latent space representations rather than through the full network. This should help to not discard the progress of the pretraining due to a mismatch in the latent representation of the different domains.



(a) Results of the GAN for sketch reconstruction.

(b) Results of the GAN for face reconstruction.

Figure 8: Reconstruction of face and sketch images by the GAN's. Left images show the ground truth right images show the reconstruction

4 Results

4.1 GAN

We started off by training the GAN architectures as pretraining for the Cycle GAN for $100,000$ steps each. As one can see in the results shown in Figure 8 the generation of realistically looking face worked out quite well. Yet the reconstruction task is only barely fulfilled. Also varying the weights of the losses did not improve the reconstruction. The cause seemed to lie within the unbounded classification of the WGAN Discriminator. As the margin of the output of the discriminator steadily grew, the L1-loss of the generator remained at the same level. Sooner or later, depending on the loss weights, the adversarial loss went beyond the reconstruction loss focusing the training on the adversarial part. Yet all measures to reduce this effect or using another discriminator with a bounded output did not improve the result. Yet the generated images often resembled real instances and thus we used the pretrained networks in our Cycle GAN model.

4.2 Cycle GAN

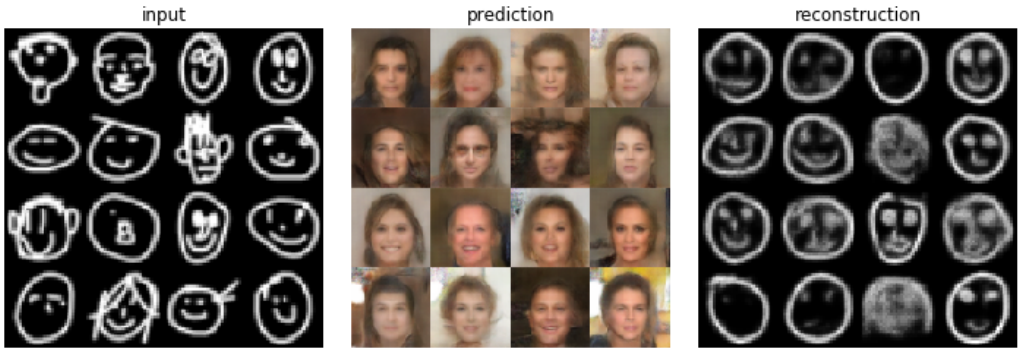
For our Cycle GAN model we followed three different approaches. Once training the full Cycle GAN architecture without pretraining and without linear layers in the latent representation, once using the pretrained models training the full model and lastly training only the linear layers.
The results of the three approaches can be seen in Figure 9. One can see that the quality of the generated fake faces is roughly the same for the latter two models. The quality is comparable to the pretrainied models, thus reasonably realistic images of faces are produced by the models. The model trained from scratch does not produce sharp images with an output resembling an average of the faces rather then a specific one. Yet this was not the only task we want to fulfill. The reconstruction task seemed to be way more difficult. For all models roughly the general shape of the sketches could be reconstructed. In the output of the models where the full architecture was learned, the lines are blurry showing an averaging character of the generation. For the model with a frozen pretrained part of the network the sharpness of the lines is preserved better resulting in a slightly better reconstruction. To quantitatively compare the models we introduce the frechet inception distance (FID score) [10] which measures the difference between the input and its reconstructon. The evaluation shown in Table 1 supports the above observations leaving the model that only trained the latent layers with the best score.

	trained from scratch	whole model	only latent layers
sketch cycle	83.32	60.22	33.21
face cycle	81.87	57.61	46.41
mean	82.59	58.91	39.81

Table 1: FID scores of the three approaches. The model trained from scratch, the pretrained model where the whole model kept on trainin and the model where only the layers in the latent representation are trained after the pretraining.

Another interesting part of the evaluation is the consistency of the output. As sampling is involved in the translation using a VAE, we want to examine the influence of different input images in contrast to different samplings. To do so we classified some images multiple times but with the same random seed for the different input images. In Figure 10 we see that the variety caused by the sampling is almost the same as the one produced by different input images. This certainly is one big reason for the reconstruction to be so hard to learn since the information being encoded is broadly distributed by the sampling making it harder to preserve. This is a major flaw in the architecture.

However the approach of pretraining the model and to only learn the connection in the latent representation seems to be a promising way to learn a translation between two image domains.



(a) Output of sketch cycle of model trained from scratch without additional latent layers

(b) Output of sketch cycle of model using pretraining and being trained completely

(c) Output of sktech cycle of model using pretraining and only training in latent layers

Figure 9: Comparison of the results of the three different approaches each showing the input, fake face prediction and the reconstruction from the generated face.

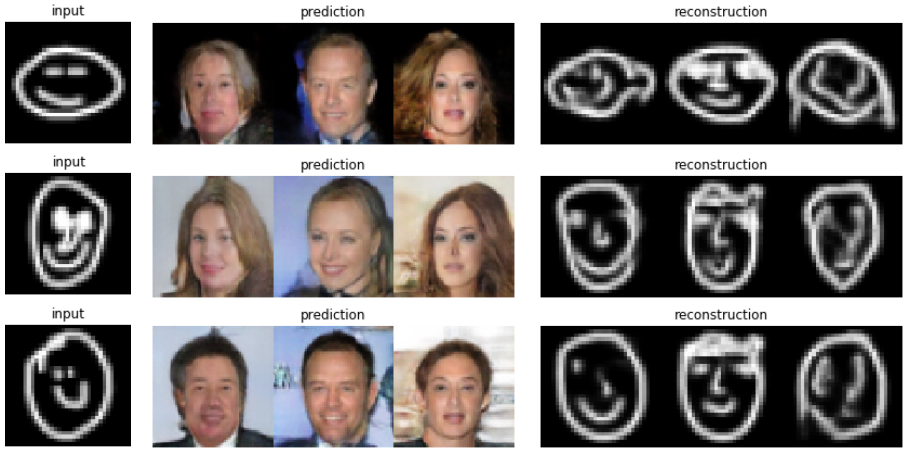


Figure 10: Comparison of influence of different samples of the encoding of the same image (from left to right) and the influence of different input images (top to bottom).

5 Conclusion

Written by Christopher Lüken-Winkels

In our project we trained a neural network to learn a connection between simple drawn sketches of faces to in the end generate new faces from sketch images. The general approach of modifying the original Cycle GAN architecture by using a VAE as generator does not seem to be the optimal choice. There are difficulties in enforcing a good reconstruction leading to rather inconsistent output. However we achieved a reasonably well generation of faces from the sketch images with some cycle consistency left with our approach of pretraining the and focusing the training on linear layers in the latent dimension. With a more suitable baseline for the generating model, this could be a good approach to learn translations between different image domains that are not spatially consistent in the transition form one to the other domain.

Future work To continue from here the first problem to focus on would be the improvement of the reconstruction task. A broad hyperparameter search could still yield a better performance in this manner. As there are still some problems concerning the adversarial loss always surpassing the reconstruction loss a different choice for discriminator might also help to improve this. One possibility would be a Markovian Discriminator as an alternative to the standard discriminator which classifies regions of the input rather than the full image and produces a more precise evaluation of the generated images.

References

[1] *The Quick, Draw! Dataset*. Available at `https://github.com/googlecreativelab/quickdraw-dataset` (cit. on p. 3).

[2] Ziwei Liu et al. "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. 2015 (cit. on p. 3).

[3] Jun-Yan Zhu et al. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". In: *Computer Vision (ICCV), 2017 IEEE International Conference on*. 2017 (cit. on p. 5).

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597 (2015). arXiv: `1505.04597` (cit. on p. 5).

[5] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2013. eprint: `arXiv:1312.6114` (cit. on p. 5).

[6] Alec Radford, Luke Metz, and Soumith Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2015. eprint: `arXiv:1511.06434` (cit. on p. 6).

[7] Martin Arjovsky, Soumith Chintala, and Léon Bottou. *Wasserstein GAN*. 2017. eprint: `arXiv:1701.07875` (cit. on p. 6).

[8] S. Kullback and R. A. Leibler. "On Information and Sufficiency". In: *Ann. Math. Statist.* 22.1 (1951), pp. 79–86 (cit. on p. 7).

[9] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. eprint: `arXiv:1412.6980` (cit. on p. 7).

[10] Martin Heusel et al. "GANs Trained by a Two Time-Scale Update Rule Converge to a Nash Equilibrium". In: *CoRR* abs/1706.08500 (2017). arXiv: `1706.08500` (cit. on p. 11).