Automatic transformation from animated face image to real human face image

Ryota Eki

Stanford University

reki@stanford.edu

Abstract

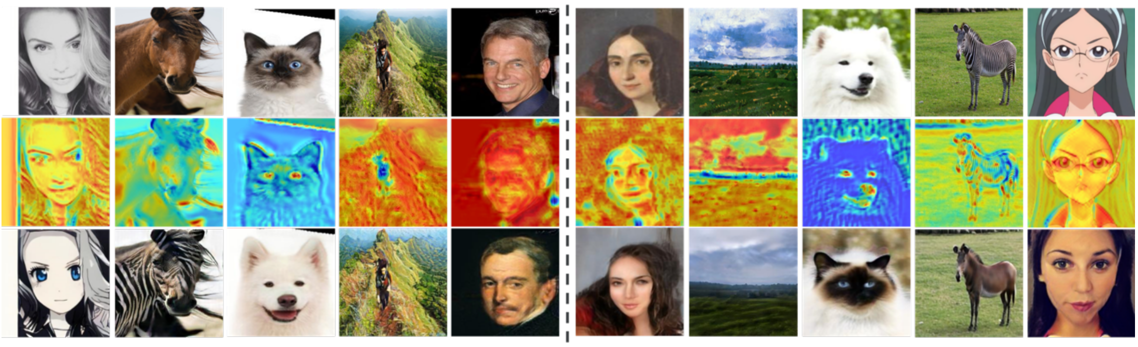
With the rapid development of artificial intelligence, the development of humanoid robots will become a reality in the future. And if this becomes a reality, it will be possible to reconstruct fictional characters from the world of anime and manga that do not exist in reality. One of the significant challenges in realizing this is the reproduction of appearance. It would be challenging to recreate the form of a virtual character in the real world. In this paper, we will try to solve this problem using various approaches.

Introduction

The goal of this paper is the automatic conversion of animated human images to real human images. To achieve this, we use three methods: 1. calculation of the difference vector of the image between the two domains 2. transformation using neural network 3. transformation using CycleGAN+α

Related Work

Most of the methods in this task are GAN-based methods. In particular, since it is challenging to create corresponding paired datasets between two domains, models that do not require such paired datasets are often used. The most successful strategy is probably the one called U-GAT-IT by Junho Kim et.al. (2020). This method combines heatmaps and GAN to achieve high-quality inter-domain transformations.



(U-GAT-IT, Junho Kim et.al. (2020) https://github.com/taki0112/UGATIT)

Dataset

We prepared two datasets: 15000 animated human face images and 15000 real human face images. The animation dataset is from Kaggle, and the real human dataset is from CelebA. Both datasets contain only faces. We also adjusted the image size of the dataset depending on the methods we are about to introduce.

Methods

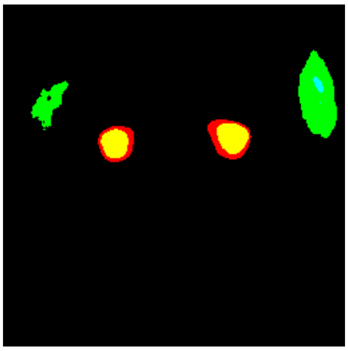
Method 1: Difference Vector (traintrans.py, testtrans.py)

Procedure

The first method attempts to compute the difference vector of images between two domains. We randomly extract 100 images from each of the two 15,000 dataset images each time and take their average. We compare the two average images of the two domains and multiply the difference vector of them by the learning rate to update the entire difference vector. With each iteration being one iteration, we trained 10000 iterations.

Result/Discussion

We recorded the difference vectors for each of the 1000 iterations. We used a total of 10 difference vectors to transform the test data for each of the animated human face images, all of which produced the following image.



Considering this result, it is expected that there is no difference vector that can uniformly represent the difference between the two domains. We can say that the vector representation of this method is too simple to represent the differences between the two domains. On the other hand, we can see that the two orange areas in the center of this image are probably related to the eyes. Therefore, the difference vector does capture some characteristics of the differences in the images between the two domains. Thus, although the difference vector is not perfect for transforming the images, it may be possible to construct a model using these difference vectors as an auxiliary feature.

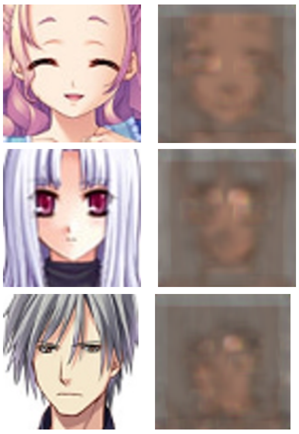
Method 2: Neural Network (trainnn.py, testnn.py)

Procedure

animated human face image) using a convolutional neural network. Then, we decode the encoded one using a convolutional neural network. The model is trained by comparing the output image with the destination image (real human face image).

Result/Discussion

The following are some of the input and output pair.



The output images show that the model successfully captured the characteristic outlines of the face and hairstyle. Therefore, although we learned only 15 epochs using a thin layer network due to the time required for learning, if we train more epochs using a thicker layer, the model will probably be able to capture more detailed features and convert the images better. In addition, we think that the output image can be effectively used as an auxiliary feature map for another model, as in the previous section.

Method 3: CycleGAN + α (create_anime_angle_data.py, create_human_angle_data.py)

Procedure

CycleGAN is an inter-domain image-to-image translation method. Compared with other image-to-image translation method such as pix2pixHD, CycleGAN does not require a paired dataset of the two domains. Therefore, we concluded that it is suitable for this task.

CycleGAN consists of generator and discriminator, where the generator generates images and the Discriminator classifies whether the generated images are true or false. By repeating this process, we improve the accuracy of both systems.

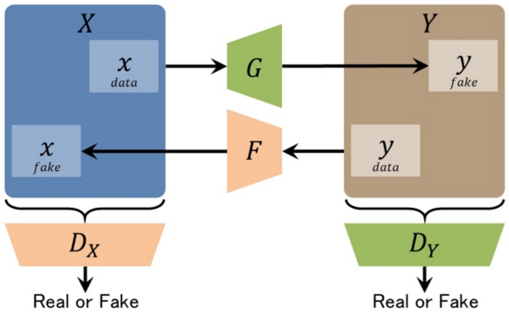
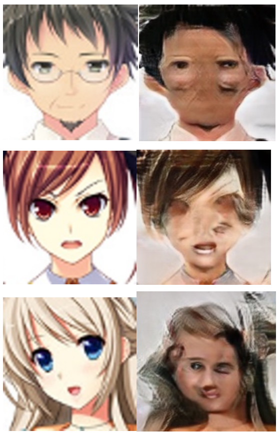


fig: the system of CycleGAN

We used this method, CycleGAN, to convert an animated human image to a live-action human image. Initially, we wanted to train for more than 50 epochs, but we were only able to train for 15 epochs due to resource limitations.

Result/Discussion

The following are some of the input and output pair.



As we can see from these results, the model seems to be learning correctly. If we train the model for a sufficient number of epochs, we can expect high-quality transformations. On the other hand, we can also see the challenges of this model. For example, in the third output, the facial parts are not arranged correctly compared with the first and second outputs. A possible reason for this is that in the first and second cases, the orientation of the face in the source image is frontal, while in the third case, the face is slightly tilted. The model may not have been able to correct this misalignment correctly. Based on this, we tried the following improvement plan.

Improvement

To deal with the different angles of the faces, we tried to pass the data including the angle information of the faces to the model. (Here, the face angle refers to the angle with respect to the linear axis perpendicular to the image plane.) To obtain the face angle information, we first detected the landmarks on the face. Then, we calculated the tilt of the face based on the positional information of the eye landmarks, scaled the angle, and incorporated the angle information into the data by incorporating the information as the fourth layer of the image data. As for landmark detection, we used existing detection algorithm. We attempted to train CycleGAN on these newly created datasets, but were unable to do so because we could not make change to CycleGAN so that the model adapts to these datasets.

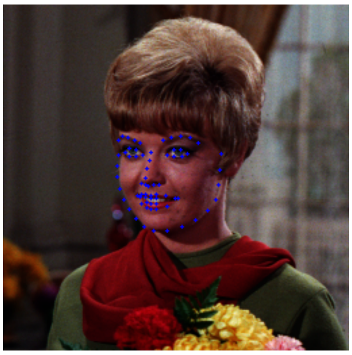


Fig: face landmark example

Conclusion & Future Work

In this experiment, we were able to achieve some positive results despite our limited resources. The output results of the neural network and difference vectors correctly captured the features of the original images and are expected to be even more helpful when used in conjunction with other models. We can expect these algorithms to be useful in other image transformation tasks as feature map generation. This experiment also helped us to understand the difficulty of this task. When we trained CycleGAN, we also trained in the reverse direction, i.e., from real human images to animated human images, and this seemed to produce better results in the same training time. We think the reason for this has to do with how we evaluate each result. We are naturally more familiar with real people than with animated ones. Therefore, when we assess the output, we overreact to even the slightest sense of discomfort when we see a real person's image outputs. In contrast, we are not so used to seeing an animated image, so we subconsciously do not care as much, even if there is some discomfort. Therefore, it seems that the evaluation of the output of real human images naturally becomes more severe, making it difficult to feel that we achieved some results. Therefore, in order to achieve significant results in this task, it is essential to have sufficient computing resources, and we would like to try again when we can secure them.

References:

[1] celebA dataset http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

[2] Kaggle dataset https://www.kaggle.com/splcher/animefacedataset

[3] CycleGAN figure https://blog.negativemind.com/2020/03/30/cyclegan-image-to-image-translation-by-learning-some-relationship-between-the-domains/

[4] neural network instruction https://child-programmer.com/ai/cnn-originaldataset-samplecode/

[5]anime face landmark https://github.com/kanosawa/anime_face_landmark_detection

[6]human face landmark https://qiita.com/mimitaro/items/bbc58051104eafc1eb38

[7] CycleGAN instruction https://www.slideshare.net/meownoisy/cyclegan-192304094

[8] CycleGAN https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix