Automated Landmark-Guided CartoonGAN

Xin Liu, Jingru Wu, Hung-yu Chang, Xuefan Zha

January 7, 2020

Abstract

Style transfer is popular topic and have wide applications in daily life. In this final project, we implement an automated cartoon face generator based on CycleGAN, which solve the hair color and face landmark inconsistency between different domain. Our goal is to generate high-quality cartoon faces from real faces picture. To achieve this, we proposed hair color constrained method using conditional generator and discriminator and facial landmark constraint method,which consists of pre-trained landmark regressor local discriminator approach and landmark-aligned transformation approach.

1 Introduction

Generating cartoonize faces from people's faces has lots of applications, especially on social media. Unfortunately, most of these woks do not keep important features from real face images. For example, one of the most popular style transfer algorithms, Cycle-Consistent Adversarial Networks (CycleGAN), can get weird results in this kind of tasks. The main reason is that the geometric appearance of real face and cartoon face may differ significantly from each other. Previous works by Wu *et.al.* [1] showed that using hand-labeled landmarks data can improve the CycleGAN's outputs. However, it is not convenient to end-users. In order to solve this problem, we propose an automated cartoon face generator, which automatically detect landmarks on human faces and guiding our local discriminator in CycleGAN. We also tried data augmentation to see if we can generate side view cartoon faces. The main contributions of our work are as follow:

- We adopted Multi-task Cascaded Convolutional Networks as an automation landmark detector (MTCNN) in Cartoon Face Generator to guide our local discriminator in CycleGAN. Comparing to previous research, we do not need hand labeled data which will definitely improve user experience.
- Our second contribution is implemented a local discriminator around eyes. Comparing to only using a global discriminator, local discriminator helps restore appearances from different camera angles.

- We also tried an alternative approach, which is applying a transformation to the cartoon image to align facial features in cartoon images to the real images. The discriminator works as a global conditional discriminator as key features are supposed to be well aligned.
- Unlike the previous paper, we also included hair color control to match the hair color of input and output images.

2 Related Work

2.1 Style Transfer Between Cartoon and Real

Image-to-image translation is a class of vision and graphics problems where the goal is to learn the mapping between an input image and an output image, which is applied widely in style translation. Style transfer generally based on image-optimisation-based and model-optimisation-based[2].With development of CNN, more generation models are proposed to deal with this problem, in which Generative Adversarial Network (GAN) is firstly and most commonly used as unsupervised learning method.[3] However, in cartoon face generation, the data from real domain to fake domain is always unpaired, which is very different from lots of traditional style transfer tasks since they always require paired data[4].
Several methods have been proposed to address this problem.DualGAN revises close looping allowing images from either domain to be translated and reconstructed[5] DiscoGAN use reconstructed loss to map original domain to target domain[6].CycleGAN was introduced to use the cycle consistency loss to train two pairs of generators and discriminators in order to regularize the solution of trained networks.[7] However, CycleGAN fail to link geometric structures of the two domains which are so different from each other. To deal with this problem, landmark assisted method is also proposed in [1]. However, this paper still need label landmark by hand and make it difficult to combine this with total automatic system.

2.2 Face Landmark Detection

Face landmark detection is a key step in the field of face recognition and analysis. It is an important steps of other face-related problems such as automatic face recognition, expression analysis, 3D face reconstruction and 3D animation. In our work, we plan to use face landmark detector as a tool to guide our CycleGAN generator. Methods in face landmark detection can be categorized into traditional image processing method and deep learning method. Deep learning methods has become the mainstream methods in facial landmark detection since 2013.

Traditional Facial Landmarks Detection Method:

- **Active Shape Model and Active Appearance Model** In 1995 Tim Cootes and Chris Taylor proposed Active shape models(ASMs) [8]. ASMs are statistical models of the shape of objects which iteratively deform to fit to an example of the object in a

new image. In 1998, Cootes *et. al.* improved ASM [9] and proposed Active Appearance Model by using shape constraints and adding texture features of the entire face region.

- **Cascaded pose regression** In 2010, Dollar [10] proposed CPR (Cascaded Pose Regression), which progressively refines a specified initial prediction value through a series of regressions. Each regression instrument relies on the output of the previous regression to perform simple Image manipulation, the entire system can automatically learn from the training samples.

Deep Learning Facial Landmarks Detection Method:

- **Deep Convolutional Network Cascade for Facial Point Detection** In 2013, Sun et al. applied CNN to face keypoint detection for the first time [11]. By carefully designing a cascaded convolutional neural network with three levels, the author not only improves the problem of initial misplacement leading to local optimization, but also uses CNN's powerful feature extraction capability to obtain more accurate keypoint detection.
- **Extensive Facial Landmark Localization with Coarse-to-fine Convolutional Network Cascade** Face++ improved the DCNN model, and proposed a face-to-face detection algorithm from coarse to fine, which achieved high-precision positioning of 68 human face key points [12]. The algorithm divides the face key points into internal key points and contour key points. The internal key points include 51 key points of eyebrows, eyes, nose and mouth, and the key points of the outline contain 17 key points.
- **Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks** In 2016, Zhang et al. proposed a multi-task Cascaded Convolutional Networks (MTCNN) to simultaneously handle face detection and face keypoint localization [13]. The author believes that there are often potential links between face detection and face key point detection. However, most methods do not effectively combine the two tasks. In order to make full use of the potential links between the two tasks, A multi-task cascade face detection framework is proposed to perform face detection and face key point detection simultaneously.

3 Data

3.1 Dataset

3.1.1 CartoonSet

The Cartoon set varies in 10 artwork categories, 4 color categories, and 4 proportion categories. Different components in face can be assigned with different artwork, color, and proportion.[14] So there are about trillions of combinations. They already provide two datasets with one containing 100 thousand images and another containing 10 thousand images. The disadvantages as described before, is the fixed position of key components. We will use various methods to solve this problem.

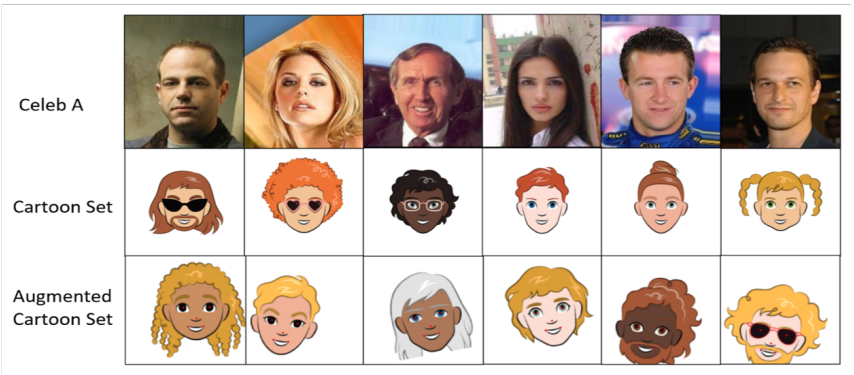


Figure 1: Dataset examples

3.1.2 CelebA

The CelebFaces Attribute contains more than 20 thousand images from more than 10 thousand people. Each image is welled annotated with bounding boxes, 5 points and 40 attributes. The 5 points indicates the position of two eyes, noses, and the edge of mouths. The 40 attributes describe some characteristics like attractive, hair colors, facial expressions.[15]

3.1.3 WIDER FACE

WIDER Face is used in our training for face detection, which is a public benchmark dataset for face detection.[16] . It contains 16101 high resolution images and over 200,000 face annotations with labelled bounding box coordinates.It the organized based on 61 event scenes and has high variability in pose, scale and environment illumination. In our task, this dataset is split into 80%/20% for training and validation usage respectively.

3.1.4 LFW+NET

For landmark detection procedure, we used the dataset from [17]. This dataset consists of 5590 images from LFW and 7876 images found from internet, providing with bounding box position and five facial points of eyes,nose and mouth.

3.2 Data Pre-Processing

3.2.1 Haircolor sorted Dataset

We created a pipeline to generate a balanced dataset that sorted by hair color (black, grey, brown, blonde). Each color contains 5000 CartoonSet images and 5000 real faces image from CelebA.

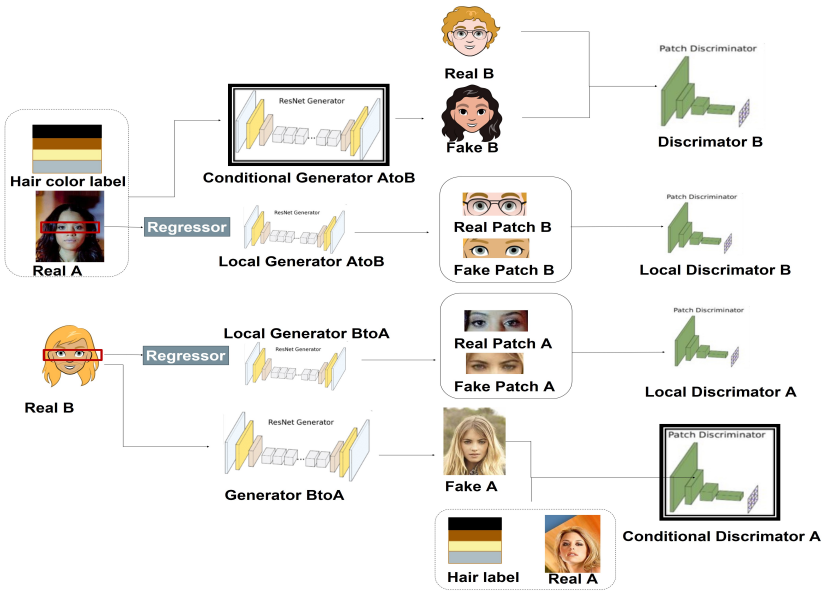


Figure 2: Architecture of Whole System Design

3.2.2 Data Augmentation

Because of the high similarity of Cartoon Set, where the size and location of face, eyes, mouth are the same, we implement data augmentation of random translation, random rotation within 15 degrees and random scale between 1.0 and 1.3 to achieve higher diversity. Besides, the same transformations are applied in their landmark to obtain corresponding landmark after augmentation as shown in Fig.1.

4 Method

4.1 Pipeline

We train our model with two stages to make automatic landmark-assisted framework. The whole system is shown in Fig.2 Based on base model of cycleGAN, we revised generator and discrimator by adding hair color constraint and facial landmark constraint. In hair color constraint, we use conditional generator and discriminator to add hair color label into network. In facial landmark constraint, we trained two types of network to keep consistency between landmarks. A landmark detection network pretrained by other dataset was used as regressor to detect landmark automatically in revised cycleGAN.

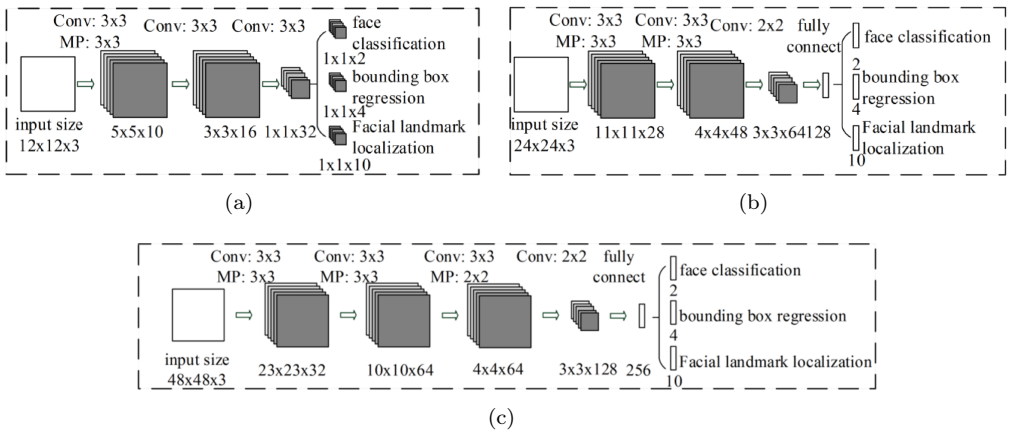


Figure 3: Architecture of PNet(a), RNet(b), ONet(c)[13]

4.2 MTCNN Landmark Detection Model

To achieve automatic face and landmark detection, we implement a cascade face detection network named MTCNN (Multi -task Cascaded Convolutional Networks) [13] . The contribution of this model is it joint face detection and alignment with multi-task learning. The overall structure of MTCNN can be divided into 3 cascaded stages with individual tasks:

- The first stage is to detect possible bounding boxes and related bounding box regression vector of face candidates. The network implemented here is a fully convolutions network named P-Net (Proposal Network). Due to the high overlapped rate of detected bounding box, the NMS (non-maximum suppression) is used to reduce redundant candidates.
- The second stage is a refinement of previous bounding box candidates. In is stage, feeding with the detected bounding boxes and regression vectors from PNet, the RNet (Refine-Net) returns classification score for rejecting false candidates while performing bounding box regression.
- After finish face detection, the third stage is focusing on detecting position of 5 face landmarks, defined as section 3.1, to represent further detail.

To sum up above procedure, the first and second stages perform a robust face detection, and the third stage of landmark detection is a detailed alignment of detected face.
As introduced above, the architectures of MTCNN, shown on Fig.3, is a cascaded combination of 3 lightweight CNN: PNet, RNet and ONet. The PNet and ONet are simply a fully-connected convolutions network with only 2 convolution layer. For increasing the diversity reduce computation, the RNet change the 5*5 kernel size into 3*3. Besides,the feature depth

is increased into 64 for better performance.
The whole task for each example can be considered as three task with individual loss function: classification, bounding box regression and landmark detection. Therefore, the loss implemented during each stage is weighted combination of three part

- *Classification Loss*: since face detection is a binary classification problem, the loss function is Binary Cross Entropy:

$$L_i^{det} = -(y_i^{det} log(p_i) + (1 - y_i^{det})(1 - log(p_i))) \tag{1}$$

- *Bounding Box regression* To obtain accurate bounding box location and window size, the loss function for regression is Euclidean distance of box corner and ground truth coordinate:

$$L_i^{box} = \|\hat{y}_i^{box} - y_i^{box}\|_2^2 \tag{2}$$

- *Landmark regression* Considering the landmark detection as regression problem, the loss for landmark detection is also Euclidean distance of 5 landmarks position and ground truth coordinate:

$$L_i^{landmark} = \|\hat{y}_i^{landmark} - y_i^{landmark}\|_2^2 \tag{3}$$

The combination weights: $W_{det}, W_{box}, W_{landmark}$ employed in each stage are:

First stage: $W_{det} = 1, W_{box} = 0.5, W_{landmark} = 0.5$;
Second stage: $W_{det} = 0.5, W_{box} = 1, W_{landmark} = 0.5$;
Third stage: $W_{det} = 0.5, W_{box} = 0.5, W_{landmark} = 1$

4.3 CycleGAN Baseline Model

So we want to generate cartoon style images from human faces. A very intuitive solution is using CycleGAN. As introduced before, CycleGAN can do image translation between two domains. So in our cases, we can use CycleGAN to do translation between real domain and cartoon domain.
We firstly implement an original CycleGAN. It has two generators (G_A, G_B), and two discriminators (D_A, D_B).

- The G_A aims to translate real face $(Real_A)$ into the cartoon domain $(Fake_B)$.
- The G_B aims to translate real cartoon $(Real_B)$ into the reality domain $(Fake_A)$.
- The D_A aims to tell the generated $Fake_B$ from $Real_B$.
- The D_B aims to tell the generated $Fake_A$ from $Real_A$.

For the detail structure of generators and discriminators, the backbone of generator is U-Net and the backbone of Fully-convolution network. Based on that, we make several improvement like substituting conventional convolutions into residue blocks or dense blocks.
We apply various loss the enforce the network to learn optimal parameters.

- The adversarial loss makes the generated output match the distribution of the desired domain. In the model, we have two adversarial loss which limit the style differences between the output from generators and a random image from the desired domain. A discriminator outputs two images. One corresponds to the generated image and another corresponds to the random image. The adversarial loss is actually the cross entropy loss between these two outputs of the discriminator.

$$L_{GAN}(G_{AB}, G_{BA}, A, B) = E_{b\sim B}[logD_B(b)] + E_{a\sim A}[log(1 - D_B(G_{AB}(a)))] \quad (4)$$

- The cycle consistency loss make sure the output of the generator corresponds to the original one. The generator can change styles but should not remove objects or induce bias. In our baseline model, we employ two cycle consistency loss which make sure the $Fake_B$ and $Fake_A$ can be converted to their original domain without too much difference. The cycle consistency loss is the L1 loss between the original image and the image output by the series generators.

$$L(G_{AB}, G_{BA}, A, B) = E_{a\sim A}[||G_{BA}(G_{AB}(a)) - a||1] \quad (5)$$

- The identity loss is introduced to limit too much color composition.

$$L_{identity}(G_{AB}, G_{BA}) = E_{y\sim p_{data}(y)}[||G_{AB}(y) - y||1] + E_{x\sim p_{data}(x)}[||G_{BA}(y) - y||1] \quad (6)$$

4.4 Generator Model Selection

The structure of generator is the base of the whole project performance. In order to reach results as better as possible, we test several structures of generators. There are Res-Net, U-Net, and, Dense-Net.

Res-Net is one of the most traditional structure of generators. The information is flowing in a fully-convolution network. To make networks deeper, there are residual connections, directly concatenating the upstream and downstream feature maps. The residual connections make the gradient larger by short connecting the computation graph. In our experiments, we use Res-net with 9 residual blocks. In the cycleGAN architecture, each generator has 11.378 million parameters.[18] U-Net is a classic image generator and translator. It is composed of a encoder and a decoder. In the encoder structure, there is a series of down-sampling convolutions. Generally, the sizes of images shrink into half after each convolution. In the decoder structure, there is a series of up-sampling transpose convolutions. Like what's in the encoder, the sizes of feature maps expand into double of their sizes. Besides a serial convolution-transpose-convolution structure, there is a skip connection which concatenate one encoder feature map and one decoder feature map with same size. The skip connection ensure the transpose convolutions can receive multiple layer information. In our experiments, we use 5 layer U-Net. [19] Dense-Net is a recent-released structure. The basic structure is also a serial convolution network. However, each layer returns the concatenation of its input and the convolution output using the input. So one single layer in the middle can directly receive all convolution output from previous layers. The grate advantage of Dense-Net is 1)

eliminating gradient vanish, 2) passing features more efficiently, 3) greatly decrease number of parameters. In our experiments, we use a similar structure than that of the Res-Net for better comparison. [20]

4.5 Hair Color Constraint

Since no label constraint is feed into generator and discriminator When training CycleGAN baseline model,inconsistency exists between prior information we had and generated face, as shown in Figure 4. To extend the usages of our model we additionally provide a one hot label mapping of real faces to cartoon faces. We make the whole architecutre conditional and train the models with four different hair colors, this can also make the user free to choose his or her hair color just control by label. Inspired by [21] which can generate MNIST digits conditioned on class labels, we also used similar structure as they did(shown in pipline).Instead of taking only real face image as input we added additional class label input for hair color to the generator and discriminator of CycleGAN. We revised generator and discriminator as conditional generator and conditional discriminator separately for one domain.

In order to acieve this,we concatenate one-hot labels and real face input before every layer of the resnet generator architecture.We also reapply label mask in the first layer of discriminator.And we optimize conditional generator and conditional discriminator as before, which could exert constraints on generated face.The objective function is as below:

$$\min_G \max_D = \mathbb{E}_x[\log D(x|y)] + \mathbb{E}_z[\log(1 - D(G(z|y)))] \tag{7}$$

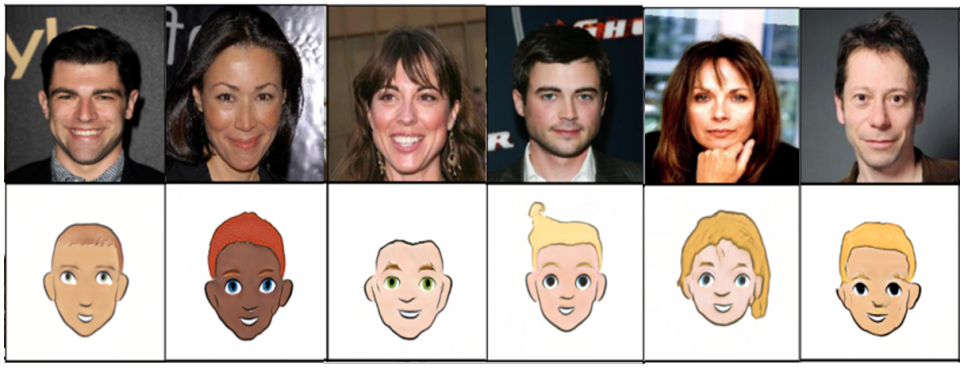


Figure 4: Hair color inconsistency between real face and generated face

4.6 Facial Landmark Constraint

By training the baseline model, we realize some drawbacks of the model. From Fig.5 and Fig.6 here, we can find all generated $Fake_B$ have a same view and also facial feature distortion.

Even the $Real_A$ have different angles, the generated results are monotonous and lack of variety. We also realize that the eye, mouth, and nose control are loose like the figure above. The translated image shows unaligned facial features. We will try two approaches to solve the problem. The first one is to introduce landmark consistency loss and three more local discriminator to constrain the pattern near these organs. The second one is to apply a transformation on cartoon images to align facial features.

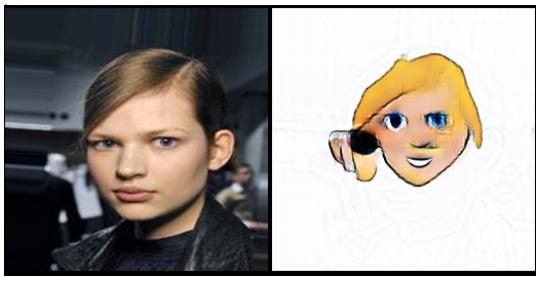


Figure 5: unaligned facial features

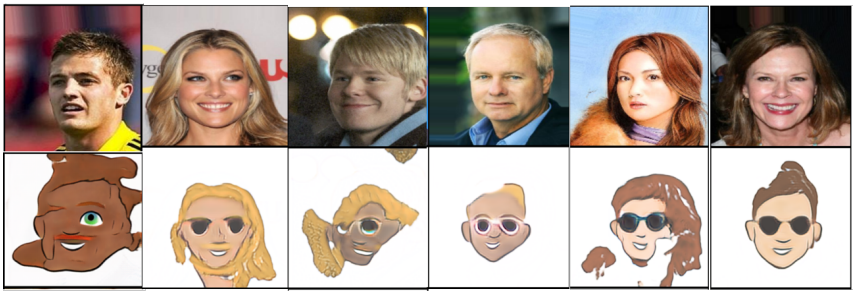


Figure 6: unaligned facial features

4.6.1 Automatic Landmark-Corrected Approach

Since we have got organs location information from real face dataset and cartoon dataset and both domains are faces, we can enforce location correspondence on their landmarks: detected landmark on generated face should preserve the landmarks of the input image.Landmark correspondence is optimized by designing landmark consistency loss and local patch discriminator. The revised loss of our cycleGAN model is computed as:

$$\mathcal{L}_{GAN} = \lambda_1 \mathcal{L}_{cycleGAN} + \lambda_2 \mathcal{L}_{GAN^{X \to Y}_{Local}} + \lambda_3 \mathcal{L}_{Landmark}$$

$\lambda_1, \lambda_2, \lambda_3$ give different weights for different loss part in cycleGAN.

Landmark Regressor in CycleGAN A pretrained MTCNN from WIDER FACE and CelebA is used in our project as face detector and landmark regressor. The MTCNN keeps robust detection results on both real face, fake face and generated face.
We used regressor in both train and inference time.Landmark regressor outputs 5-point location result which is used in loss calculation. In training time, we use landmark ground truth for human and cartoon as input and use regressor detecting landmark of generated fake human and fake cartoon face.In inference time, we use regressor detected result for both input domain and generated domain.

Landmark Consistency Loss We added give constraints on the real landmark and predicted landmark of generated face.We use $L2norm$ to calculate distance of landmark regressor output and input landmark information.The landmark consistency loss is computed as $L_{landmark}$:

$$\mathcal{L}_{Landmark} = \mathcal{L}_c(G_{(X,L)\to Y}) = \|Regressor(G_{(X,L)\to Y}(x,l)) - l\|_2$$

Where L indicates the input landmark heatmap ($l \in L$) and $Regressor$ refers to a pre-trained landmark regressor with 5-channel location output for respective domain.

Local Patch Discriminator In order to give an explicit structure constraint between the two domains, we also use three local discriminators on eyes, noses, and mouths respectively. In order to put local patch into local discriminator,we turn regressor output location into heatmap mask. Landmark heatmap mask is 2D Gaussian mask centered on organs' location.After applying landmark heatmap mask on face,we cut out three masked local patch.We added three local discriminators which have same structure as cycleGAN discriminator focusing on three different masked local patch as input instead of whole face. Optimization step for three different local discriminator is independent with each other.The objective function is computed as:

$$\mathcal{L}_{GAN}{}^{X\to Y}_{Local} = \sum_{i}^{3} \lambda_{l_i}\{\mathbb{E}_y[\log D_Y^{li}(y_p)] + \mathbb{E}_x[\log(1 - D_Y^{li}([G_{(X,L)\to Y}(x)]_p))]\}$$

where y_p and $[G(X,L)\to Y(x)]_p$ refer to local patches of cartoon face and generated cartoon faces respectively.

4.6.2 Transformation Approach

Inspired by the traditional computer vision approaches, we also try an alternative approach. The landmarks of real faces (real domain) are given. In the cartoon domain, all images are synthesized by combinations of different components. So the positions of landmarks in the cartoon domain is fixed. We can use a 2-dimensional transformation to map key facial features. As key facial features like eyes and noses and mouses are aligned, we probably don't need many local discriminators any more.

Figure 7: Eyes, nose mapping

As described above, we know the positions of landmarks. Suppose there is a transformation which is an 33 matrix. We have the following formula.

$$\begin{bmatrix} x_1^{'} & ... & x_n^{'} \\ y_1^{'} & ... & y_n^{'} \\ 1 & ... & 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_1 & ... & x_n \\ y_1 & ... & y_n \\ 1 & ... & 1 \end{bmatrix} \tag{8}$$

By using singular value decomposition (SVD), we can solve all parameters in this equation using more than 3 points. Then, we can apply the transformation to get the mapped cartoon image.

As we can see in the Fig.7, all key facial features are aligned. The net receives such kind of A-B pairs and minimize object functions as described before.

4.7 Automatic Mixed Precision

We know that there are three widely-accepted float number type, double precision, single precision, and half precision. A standard single precision float number occupies 32 bit to store its value. The double takes double and the half takes half. The traditional default data type in deep learning framework is single precision. However, the half precision computation usually can be computed with double speed. PyTorch offers a method ".half()" to manually cast a model/tensor to half precision.

There are some drawbacks existing in the half precision computation. The biggest one is gradient overflow and underflow. The dynamic range of half precision is significantly narrower than that of single precision. According to the statistics, there is about 5% gradient values locating outside the dynamic range of half precision. Mostly, the gradient underflow is a biggest issue. [22] The automatic mixed precision is released by Baidu and NVIDIA, which is a library called Apex, can process loss scaling. By using the Apex, the weights of neural

networks are stored in single precision. When processing forward propagation, the weights is automatically transformed to half precision to get a computation boost. After computing the loss, the loss is scaled up (typically) to enable there is no overflow and underflow. The loss is back propagated and the weight is updated after applying a scale down.
In part of our experiments, we use AMP to yield a computation boost.

5 Evaluation

Evaluation method we used is human evaluation based on generated face quality,which will be conducted in four group members to compare image by *Identity,Realistic* and *AsProfile* for results[1].

- *Identity* indicates whether our generated results keeps the identity of input human faces.
- *Realistic* refers to whether the generated results looks real in target domain.
- *AsProfile* is an overall evaluation from user, indicates whether it is a good result as their profile.

6 Results

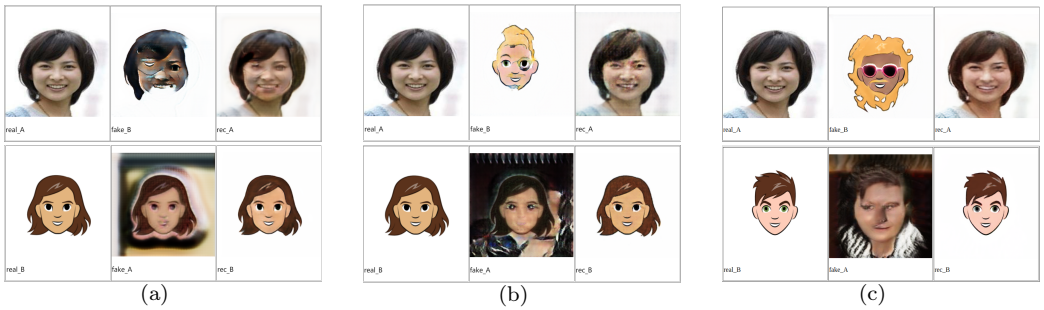
6.1 CycleGAN Baseline model

6.1.1 Dataset Selection

We use both CartoonSet 10k(100 thousand images) and CartoonSet 100k(100 thousand images). The training result in 4 epochs is shown below in Fig.8. We can find that based on larger images, the training will have more "real" results. The training based on 100k dataset will takes about 1 day for two RTX 2070 Super GPU. That's acceptable for parameter tuning. We turn to simplify the whole question. We decided to focus on hair color control and facial feature alignment. Considering the complexity of these two sub-questions, it's reasonable to train models with smaller datasets.

6.1.2 Generator Selection

We used both ResNet and U-Net on CartoonSet 10k for generator to visualize reconstruction result. The training result and loss function is shown as below in Fig.8 a and b. From (a), which is result of U-Net, we can see the shape preservation is better than (b), which is U-Net result. U-Net gets worse performance. We also see if changing serial convolutions in U-Nets to dense connection will help improvements. The result is even worse.
The underlying reason is, all feature maps in the final layer of Res-Net, are processed by all convolutions. However, the feature maps which pass by skip connections are not so well-processed. So the more skip connections a model has, the worse the image translation performance it will be.



(a) (b) (c)

Figure 8: Result of CartoonSet 10k and 100k

6.2 Landmark Regressor Detection

Our landmark regressor keep robust detection on both human face, cartoon face as well as generated fake cartoon face. As demonstrated in Fig.9, the result in first row shows the capability of face and landmark detection of human under various lighting environment and face poses. The second row shows the detection on cartoon image. and the third row are result form generated fake cartoon face.

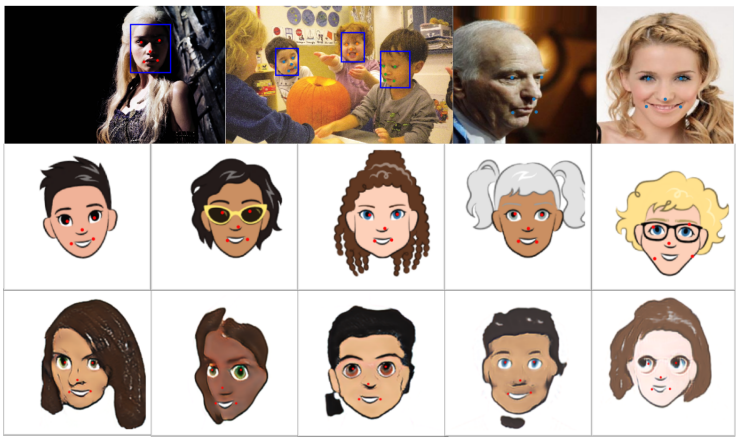


Figure 9: Face and Landmark Detection Result.

6.3 Hair Color Control

By adding hair labeling control on four color class, we solve the problem of inconsistency of face color. As shown in Figure 4 before, we can see hair color is selected randomly in generated face. After applying hair label we can see hair color corresponding to real face

and fixed problem in Figure 4. In four color types, we can see in Figure 10 color control also classify four types hair color.

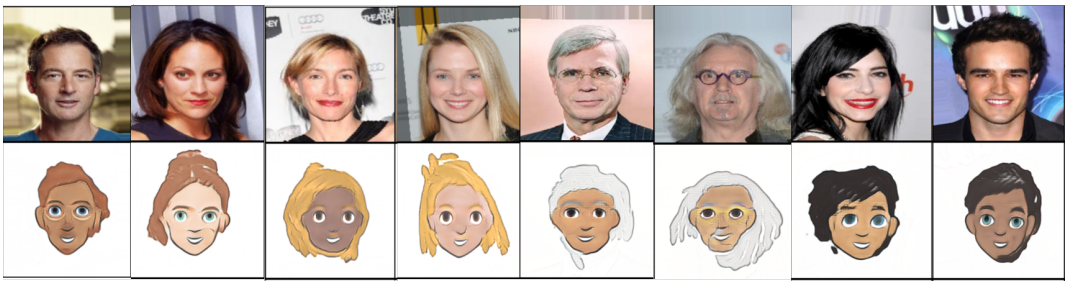


Figure 10: Hair color control result

6.4 Facial Landmark Constraint

Before applying landmark-corrected approach and transformation approach,we can see in Fig.6 apparent distortion in generated face. In eye patch part, lots of generated face wears glasses while no glass in original face.

6.4.1 Automatic Landmark-Corrected Approach

After correcting landmark location, we can see more accurate on organs' locationas shown in Fig.11.It's also clear that no more glass on eye patch corresponding to original face and also generate more variance on emotions.

Figure 11: Automatic Landmark-Corrected result

6.4.2 Transformation Approach

The results from transformation approach are shown below. Red lines are alignments of facial features. Fig.12 indicates that facial features are aligned well. Also, we can find that the contour of face is also preserved. In the middle figure, we can find the transformation

works. In the middle side-face image, the right eye is more far away from the camera. In the predicted image, the right eye is smaller than the left one. In the right one, we can find that the hair preserve the shape of hair line.
Besides, we can find the reflection in the hair, which is exactly what it is in the cartoon domain.

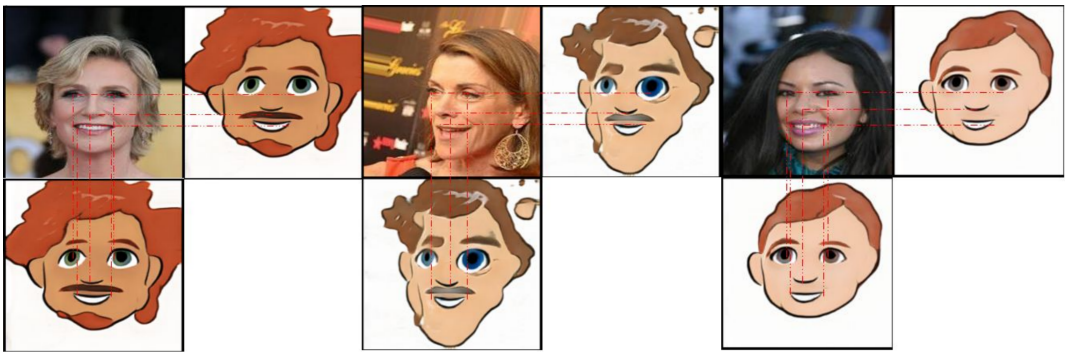


Figure 12: Transformation GAN image translation result

7 Conclusion

7.1 Model Comparison

Based on evaluation index, we compare different model generated face with cycleGAN baseline model. In hair color control,we use color label-conditioned GAN,which achieves higher *identity* especially on hair color. When training in same epochs, face *realistic* and $Asprofile$ have no apparent differences with baseline model. In facial landmark constraint, both local discriminator approach and transformation approach achieve higher $Identity$, *realistic* and $Asprofile$ in facial landmark location and shape compared to baseline model. However, transformation approach can keep more accurate location and shape consistency between input image and generated image, while local discriminator seems improve whole face consistency.

7.2 Improvement

We implemented a baseline CycleGAN and made several improvements. The baseline model is composed of two generators in charge of image translation and two discriminators in charge of recognize translated fake images.

- The first improvement is a regressor that is compatible with real domain and cartoon domain. The regressor is robust, it can help us generalize our predicted images.

- Our second improvement is a local discriminator around eyes. The discriminator takes eye region of a real image and the corresponding region in translated fake images. Comparing to only using a global discriminator, local discriminator helps restore appearances from different camera angles.
- We also tried an alternative approach, which is applying a transformation to the cartoon image to align facial features in cartoon images to the real images. The discriminator works as a global conditional discriminator as key features are supposed to be well aligned.
- By using augmentations and automatic mixed precision computation, we are able to generalize our model and boost our training.

7.3 Limitations

- Adding more elements on cycleGAN, training for cycleGAN is much slower than baseline model.
- Since there's no widely-accepted gold standard for image generation tasks,so we only can optimize our method on visual result.So no math quantity description is used in our evaluation.
- When given weights to different parts of loss and try to optimize it. It's much more difficulty to keep balance between different loss. So in training process visual result between epoch fluctuates a lot.

8 Future Work

It would be a challenging task to generate high-resolution image with rich detail while also a little limited in our model. In future work, it's also significant to adding more landmark details in regressor and discriminator. However, how to balance tradeoff between accurate constraints and training cost also need to be solved. After our implementation, we think it would be another interesting topic that how to generate better cartoon image when people wearing accessories such as hats and glasses. One possible solution may be including accessories detection into our landmark detector model.

References

[1] R. Wu, X. Gu, X. Tao, X. Shen, Y.-W. Tai, and J. Jia, "Landmark assisted cyclegan for cartoon face generation," *arXiv preprint arXiv:1907.01424*, 2019.

[2] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: A review," *IEEE transactions on visualization and computer graphics*, 2019.

[3] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[4] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, "Learning to generate images of outdoor scenes from attributes and semantic layouts," *arXiv preprint arXiv:1612.00215*, 2016.

[5] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2849–2857.

[6] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1857–1865.

[7] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[8] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models - their training and application."

[9] T. Cootes, G. Edward, and C. Taylor, "Active appearance model."

[10] P. Dollar, P. Welinder, and P. Perona, "Cascaded pose regression."

[11] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection."

[12] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, "Extensive facial landmark localization with coarse-to-fine convolutional network cascade."

[13] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, p. 1499–1503, Oct 2016. [Online]. Available: http://dx.doi.org/10.1109/LSP.2016.2603342

[14] A. Royer, K. Bousmalis, S. Gouws, F. Bertsch, I. Mosseri, F. Cole, and K. Murphy, "Xgan: Unsupervised image-to-image translation for many-to-many mappings," 2017.

[15] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[16] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3476–3483.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[20] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2016.

[21] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[22] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh *et al.*, "Mixed precision training," *arXiv preprint arXiv:1710.03740*, 2017.