CSCI 1430 Final Project Report:
Anime Style Transfer onto Detected Faces

Attack on Titan: Scott Kim, Liza Kolev, Phong Nguyen, Morgann Thain.
Brown University

Abstract

Our project sought to combine two things: face recognition and style transfer to carry over anime character qualities onto those recognized faces. We used a pre-trained DNN from OpenCV to recognize and locate the faces, cropped those, used the cropped image as the content for style transfer using a pre-trained VGG, and overlaid that output on the original image. This VGG came from TensorFlow's models, specifically a VGG-19 trained on ImageNet. Our results are displayed below with varying levels of success, which we will talk about in the results section.

1. Introduction

Our goal is to stylize our own human faces into looking cool like our favorite anime characters. However, it's difficult to derive style from anime faces, to apply style to faces in general, and to bridge the gap between anime and human faces in terms of proportions, texture, and feel. Our approach is to detect our faces, find a bounding box around them, then use our anime input to stylize just the areas in our face bounding boxes. Solving this problem provides a way for more a diverse set of people to enjoy seeing themselves represented through their specific favorite characters.

2. Related Work

For our face detection and location, we read through V. Argawal [1] in order to determine the architecture that would be most effective for our purposes, ultimately settling on MTCNN in order to more accurately detect multiple faces in our images, described by Zhang, K., Zhang, Z., Li, Z., Qiao, Y. [3]. For style transfer, we looked through a vareity of papers, ultimately settling on X. Hou et al. [2] due to its flexibility and our familiarity with the VGG architecture. We planned to implement CycleGAN for style transfer afterwards, as shown in Zhu, J.-Y., Park, T., Isola, P., Efros, A. A. [4], but ultimately we didn't have the time, as we would have needed to create a whole dataset for anime faces by hand.

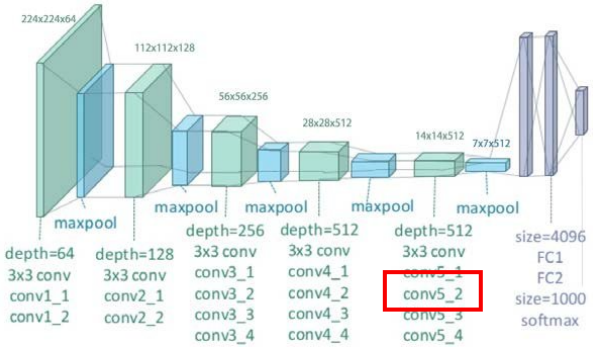
3. Method

Our project is essentially a combination of face detection and style transfer with bounding box splicing as the connection between the two. First, we detect faces in an input image with the faces we want to stylize; this image will defines the main edges and content of the result image. Then, we draw a bounding box around the faces and crop the image down to contain the face by itself. We then run our style transfer net on just the bounded faces and splice those back into the original content image to create our result.

To style transfer, we utilized a CNN, specifically using the layers of the well-known VGG19 network. We loaded it without its classification head, and extracted the style from specific layers of the VGG19 network. Specifically, we used the first convolution layers from each of the 5 blocks in the VGG19 architecture. To calculate the style of each of the layers, we calculated the gram matrix of the chosen layers, with the equation below.

$$G^l_{cd} = \frac{\sum_{ij} F^l_{ijc}(x) F^l_{ijd}(x)}{IJ}$$

We also perform this extraction from the content image, which is the same as the input image for the face detection, however, we only use the 2nd convolution layer of the 5th block of the VGG19 architecture.



Once we have extracted information from the style and con-

Figure 1. **Face Net Tradeoffs:** DNN bad on small faces

Figure 2. **Face Net Tradeoffs:** MTCNN better

tent images, we then use gradient descent for our style transfer algorithm. We calculate mean square error for our image's output relative to each target, then take the weighted sum of these losses. Our optimizer was chosen as the Adam optimizer. To also reduce high frequency artifacts, we also added a regularization term on the high frequency components of the image by using sobel filters and used this to factor into our loss.
Once both components are finished, we simply replace the area of the face in the original image with the stylized image.

4. Results

4.1. Technical Discussion

What about your method raises interesting questions? Are there any trade-offs? What is the right way to think about the changes that you made?

- **Face Detection Net Tradeoffs:** While MTCNN is great for frontal face detection and serves the purpose of style transferring to photos of multiple people, DNN is much better for frontal face detection in video, especially when a person turns their head or is a bit further back. However, DNN is, at least experimentally, much worse at detecting small faces in images, potentially because of the lower resolution it uses that makes small faces downsized to being unrecognizable through aliasing (See Fig. 1,2). So, the code should simply be adapted so that when it's used on videos, angled faces, and big faces, DNN is used. Otherwise, MTCNN should be used.

- **Bounding Box Use:** The bounding box idea fits our initial Snapchat filter motivation, but looks a bit clunky. In our experience, style transfer worked a bit differently onto backgrounds and onto faces. Generally, it was more effective onto backgrounds; artifacts and somewhat random changers in color were less obtrusive, and the style came through more clearly. Inspired by this, we actually played around with doing the opposite of our project: doing style transfer on everything except the faces. We could even use these techniques with things like blur filters to provide focus to faces, among other possibilities. Finally, we also experimented with detecting the faces in our anime face style images and using that to crop them before using them as input, which sometimes made the style a bit clearer. The Bounding Box gives us all this variability and is extensible to detection of other objects onto which style transfer might work a bit better.

- **Bounding Box Improvements:** There are some minor manual adjustments we could do that might make it more aesthetic: 1. Make a gradient in style weight as we leave the style box so it's a softer transition 2. enlargen the boxes to always include all of the face, ears, forehead, hair, chin,... 3. make the box more round/oblong.

- **VGG vs cycleGAN:** We used VGG as the architecture in our implementation, but it had a few limitations. We could not use multiple style images so that we have much better learned style to transfer to content images. Also, the end result does not create a truly "anime" style face, as it looks more like it's stylized rather than completely drawn like in an anime. For improvements, we thought of using cycleGAN. It provides image-to-image translation features and also takes in more than one style image as input. It has the potential to learn mappings between human faces and anime faces, however in this project we did not have enough data on each category, as anime face datasets available online were

Figure 3. **Extracting Style from Faces:** Manga Produces Clear Style

Figure 4. **Extracting Style from Faces:** Titan Style is Scary

composed of mostly anime female characters. There are results online that show results of a CycleGAN providing better results for anime style transfer, as it looks fully drawn and simplistic, like a drawing would be.

- **Extracting Style From Faces:** We enjoyed but had a few struggles with trying to learn the styles of anime faces (and also stuff like each others' faces, paintings, etc.). The Attack on Titan titan faces for example, especially with the style weight hyperparameter set higher, would produce kinda horrifying images in which random vague eyes and teeth and so on would appear subtly and creepily in random parts of the stylized result faces (See Fig 4). For more clearly stylized characters with face and hair color palettes and with highly structured faces (e.g. Dio), the style could come out more clearly and give cool effects, albeit still either being a bit intrusive or being not super noticeable. This was most notable in using manga faces as style input images, since their black and white style and streaky hair is very clear and was reproduced very well in our result images (including the black hair with white streaks/highlights! See Fig 3). Since our motivation was to look like our favorite characters, I think this was a good route, but style transfer might generally do better sampling from pictures in which the color palette and texture is clear, and any substructures in the image (like the swirls in a Van Gogh) can kind of be distributed across the result image and still look good (unlike eyes and teeth).

- **Combining Anime and Human Faces:** As discussed above, this was a bit tough, and another problem is that Anime faces have much different proportions to real humans and we weren't able to emulate that just through style transfer, which made our result faces a bit less anime-like. Finding a way to incorporate this would probably greatly improve our results.

- **Applying Style Transfer to Faces:** As pointed out in the Bounding Box Use discussion, style transfer onto faces gave us some issues (See Fig 5). A few things worked well, namely when we could learn the hair streak style from manga and then effectively apply it to humans' hair. However, not much other features transferred super well. Sometimes we were able to get the distinct lines and more blocky nature of anime faces, but not always. Many of our results looked not significantly stylized except with some color blotches. This seems to currently be a tough problem for the style transfer community in general, and I think a lot of improvement could happen here.

4.2. Societal Discussion

1. *Critique: If the project is utilized by people merchandising without having the approval from the original copyright holder and actually brings profit, it will be a negative example for copyright protection. And since it will probably be hard to track who profits through pirating, the loss for the original copyright owner could be hard to compensate.*

   Answer: I think this would be a bit difficult for us, an unknown group of 4 people. However, probably the

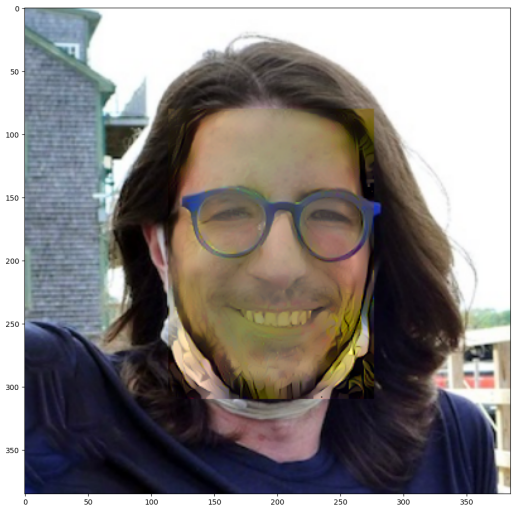


Figure 5. **Style Transfer to Faces:** Face Style + Face Content + Glasses + Facial Hair + Masks → Bad Output

best way to do this would be to just contact and get permission from the animation studios. We could even partner with animators to get the original images for training data as well as solve copyright issues.

2. *Critique: The style of the anime work itself could probably represent a smaller range of diversity in terms of races. Therefore, the CNN trained will produce images that could not successfully capture the features of the groups of people underrepresented in the anime.*

   Answer: As mentioned in our Technical Discussion, we noted that even in previous Snapchat filters we definitely saw a race categorization problem. This may work similarly through our project, especially since we found that our style transfer worked a bit worse onto complicated faces like James' which has glasses and especially beards it seemed to not play well with. It's a bit unclear how our style net understands facial features, hair, etc., but this is something that would definitely warrant a lot more testing.

3. *Critique: The project may be utilized by individuals attempting to avoid copyright infringement through the creation of counterfeit material. While not technically illegal, these individuals would be skating the boundaries of what is and is not legal.*

   Answer: Somewhat similar to our first critique, it would help a bit to have our process sanctioned by animation studios so that they at least expect this behavior. However, we even had a result that was basically the AoT

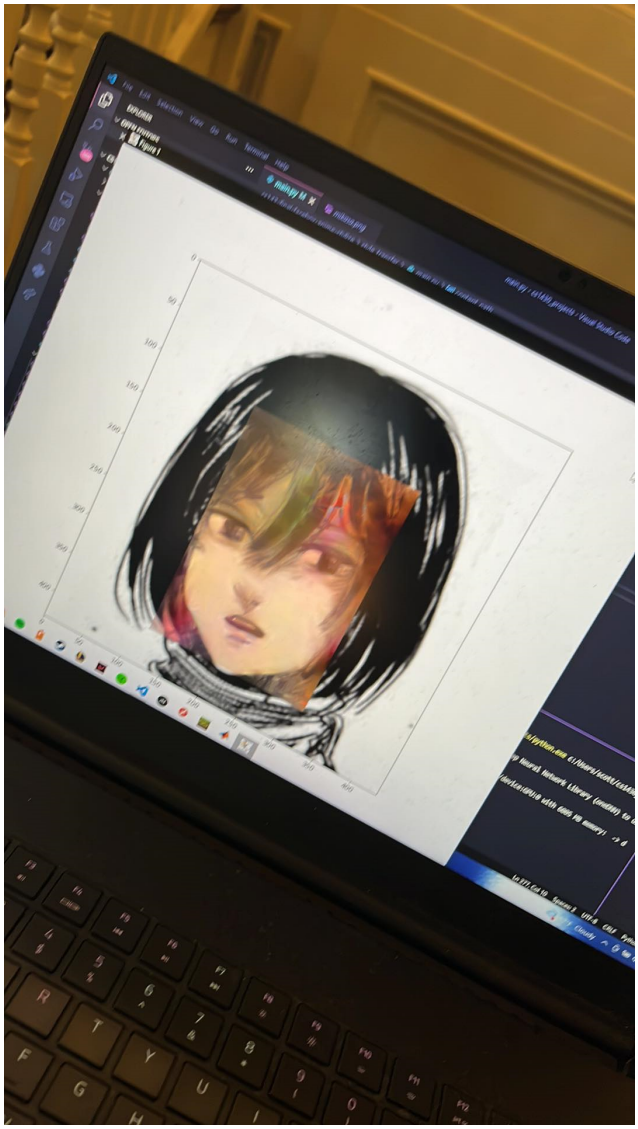


Figure 6. Effective Colorization of Manga!

Mikasa manga image ending up just colorized the panel really well by using Scott's face as a style image. I could easily imagine people stylizing other creator's work and passing it off as their own (See Fig 6). However, copying, tracing, and so on are just general problems in the anime and art communities, and I don't think it is in our control our scope to control people trying to copy other people's work.

4. *Critique: Depending on the data set of human facial images, consent may be an issue. Infringement upon intellectual property of the individual is possible.*

   Answer: This may be true for how VGG19 was trained, but our process now only uses a single content input image with faces and a single style input image, so this

is not an issue for us.

5. *Critique: The proposal mentions that the training data of the anime images will be manually collected, bringing the potential risk of inadequate amount of training data for a CNN. Therefore, we would like to suggest the team probably come up with more measures to collect the data in order to make it more sufficient for training.*

   Answer: As said for the first critique, we could contact and partner with animators of many different animes. They will have many images of each character with different facial expressions as well as different angles. This would allow us to not only have more training data but also would allow us in the future to work on the live style transferring through video.

5. Conclusion

What we did: We created two things and combined them: 1. VGG style transfer and 2. face detection with bounding boxes. **Why it matters:** This is basically a form of a Snapchat filter, but it's much more customizable since you can choose the image that founds the style. We've used Snapchat's anime filter before, and the least fun part is that it basically transformed your face into 1. something not recognizable as you and 2. a face from a small set of archetypes that seemed heavily influenced by your race and gender expression. Our setup is a bit slower and cruder, but it leaves your face and its structure greatly intact, and allows you to style it according to whatever characters or style you desire. **Impact:** While our face detection and styling is certainly inhibited by biases in data and a certain way of extracting style from an image, our setup allows people of all races and genders to identify with and represent themselves through their favorites characters and art, and leaves themselves recognizable as themselves through the filter. There's a lot to improve, especially speed, resolution, and quality-wise, but I think our product fills a nice niche.

References

[1] Vardan Agarwal. Face detection models: Which to use and why?, 2020. Used for making decisions about whether to use MTCNN or DNN for bounding boxes. 1

[2] Xianxu Hou, Yuanhao Gong, Bozhi Liu, Ke Sun, Jingxin Liu, Bolei Xu, Jiang Duan, and Guoping Qiu. Learning based image transformation using convolutional neural networks. *IEEE Access*, 6:49779–49792, 2018. Used for learning about, understanding, and implementing style transfer. 1

[3] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, oct 2016. Further in-depth paper on how MTCNN works for face detection and location. 1

[4] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2017. Had plans to implement CycleGAN using this paper, but ultimately didn't have time or a dataset. 1

Appendix

Team contributions

Scott Kim Poster, Presentation Speech, Testing examples, Abstract, Related Work

Liza Kolev MTCNN Face Recognition; Bounding Box creation; Live video bounding box creation for both DNN and MTCNN; References; SRC

Phong Nguyen Pretty much all of the Style Transfer Work, Method and Technical Discussion

Morgann Thain Initial+DNN Face Recognition; Bounding Box creation; original and stylized image splicing/combining; integrating Face Bounding Box code with style transfer code; supporting multiple faces, playing with diff images and Bounding Box set ups; general refactoring, testing, debugging; Report: Intro, half of Method, Results–Technical Discussion and SRC, Conclusion.