

Making yourself into a work of art

Kiah Hardcastle * and Julie Makelberge †

*Neurosciences Program, and †Graduate School of Business, Stanford University

Project Milestone for CS 230: Deep Learning, February 2019

Here we begin the implementation of a deep learning framework that will swap the face in an artistic painting with the face in a separate image. For example, given a headshot and a picture of the Mona Lisa, this framework would inpaint the headshot face into the face of the Mona Lisa, while retaining the artistic style of the Mona Lisa. While face swapping has been done before, many previous projects have simply focused on swapping faces in images that contain only a face (e.g. cropped images), or implanting faces without taking into account the pose of the face. Thus, our approach is novel in that we will replace faces in an image, while retaining the style and pose of the original face.

Neural style transfer.. give some introduction here

Approach and Results

Successful completion of our task requires several steps. First, we must identify the region containing the face in both images (art image = image A with face A, other image = image B with face B). Note that we assume that images with only one face are considered. Once face A and face B are identified and cropped out of the image, we must then align the faces. Specifically, we require that the two faces have similar positions for the eyes, nose, mouth, and chin. Using the aligned faces, we then generate a new face C, using neural style transfer (or visual attribute transfer, see below). Face C contains the content of face B, in the style of face A. Next, we will "un-align" Face C, or apply the opposite transformation applied on face A during alignment in order to get face C in the pose of face A, and replace face A with the new face C. Finally, we will smooth the boundaries between the excised region surrounding face C and the rest of the image.

Data collection. Fortunately, as many of our methods will rely on pre-trained networks, we do not need large numbers of new images with which to train our network. However, there are large datasets of artwork and faces available on the internet, which we plan to take advantage of here in order to demonstrate our approach. To generate our dataset of artwork, we use the Web Gallery of Art. This dataset is free to use for educational purposes and contains the meta data and links to images for more than 45,000 works of art, 31,000 of which are paintings. In order to create the dataset, we wrote a script to download these images (see github repo). To generate a dataset of faces, we have used our own headshots. However, we plan to use the celebA dataset if more face images are necessary.

Data pre-processing: face-cropping. The first step in our pipeline is to identify the face in both images (image A and image B), and generate new images A' and B', which are the cropped-out face of images A and B respectively. To accomplish this, we implemented an open-source face recognition from the OpenCV package. Following the procedure detailed in this blogpost: , we used Haar Feature-based Cascade Classifiers to detect the bounding box of the face. In brief, Haar features are similar to the filters in a CNN, but different in that they are not learned during training - they are pre-determined. Each filter can be used in a classifier to determine whether or not a face is present in a bounding box of the image. While the

classification based on one Haar filter is not great, combining the output of many filters (e.g. through boosting) will return a relatively good classification. The OpenCV implementation organizes the features into a cascade, so that a lot of filters are "tried" if the image within the bounding box seems promising (i.e. early filters classify the image as a face). The OpenCV implementation contains pre-trained filters for faces, which we can load and then use to detect face regions within an image (see code in the github). An example of an output of the model is given below in Figure 1:

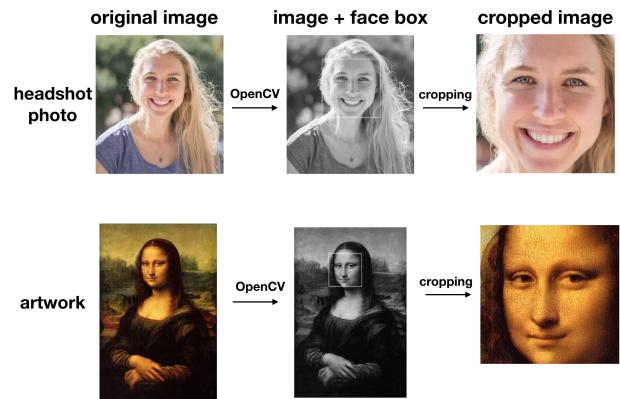


Fig. 1: Examples demonstrating face cropping pipeline.

Data pre-processing: face-alignment. The second step in our pipeline is to properly align the faces so that the pose of the face is similar between the two images. Specifically, we want the major face landmarks (eyes, nose, mouth, chin) to be in similar locations, and both faces should be facing forward. To accomplish this, we plan to use the method implemented in the paper "Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis". For now, we have chosen test images that are already facing forward and are similar in landmark position, but we will plan on completing this step more rigorously for our final project.

Style transfer between face images. The third step in our pipeline is to generate a third image that contains the content of face B (the non-artistic face) in the style of face A (the artistic face). There are several possible methods to accomplish this task. One simple method would be to use Neural Style Transfer in a manner similar to the Coursera CNN module homework. While this method is not face specific, and thus may not perform very well, it is relatively straightforward to implement and iterate on. A second method would be to follow the algorithm defined in the paper "Visual Attribute Transfer through Deep Image Analogy", by Jing Liao et al (CITE). In this paper, they [DESCRIBE METHODS IN PAPER HERE]. Finally, a third method would be to follow the algorithm described in the paper "Painting Style Transfer for Head Portraits using Convolutional Neural Networks" by

Selim et al. (CITE). Briefly, in this paper they [DESCRIBE METHODS IN PAPER HERE].

While ultimately we feel that the third method is the most promising, thus far we have tried Neural Style Transfer on the cropped images of face A and face B. In this method, we ... [briefly describe neural style transfer - use vgg19 trained on imagenet, describe content activations, style activations, and cost function that is minimized. in this problem we have increased the number of iterations in order to improve the quality of the generated image]

The output of Neural Style Transfer is shown below in Figure 2. As you can see, the output is quite poor. In particular, the color of the image is not similar to that of image A, and the style is not quite similar to that of the Mona Lisa painting.

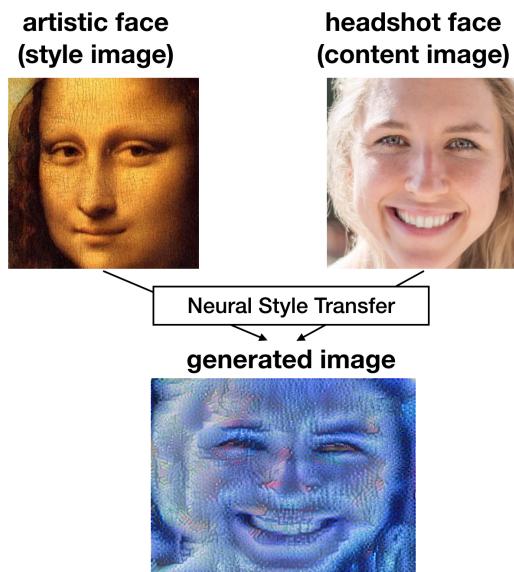


Fig. 2: Example output using Neural Style Transfer.

Re-aligning generated face with face in artistic image. Once the new face image has been generated, we must align this image with the original pose of face A in image A. In order to do this, we will apply the inverse transformation used to align the face. We are currently working on methods to align and "un"-align the faces within the images and plan on having this piece of the procedure implemented for the final project.

Face-replacement in artistic image. Finally, we must replace the original face A with the new, properly aligned face C. This task contains two components: first, we replace the pixels within the original bounding box in image A with the new image C. Second, we must then smooth the boundary between the inpainted pixels and the border. We have been able to complete the first task.

Summary

ACKNOWLEDGMENTS. We thank Andrew Ng and Kian Katanforoosh for teaching us the techniques used in this project, and to our project TA Kaidi Cao for guiding us towards workable solutions for our problem and pointing us to relevant literature.