A Neural Algorithm of Artistic Style
E4040.2018Fall.NEUR.report

Chao Yin cy2507, Fan Gao fg2425, Yang Shi ys3047
Columbia University

***Abstract*—*A Neural Algorithm of Artistic Style* [3], an algorithm that can separate and recombine content and style of images was introduced by Gatys. *et al*. This method can produce a synthetic image that combines the content of an arbitrary photograph with the appearance of some well-known artworks. We first apply their method to implement the image style transfer. Based on their setting, the style transfer of our images is not so satisfied. The problem is that the resolution i.e. size of the content image and style image will affect the transfer process. We modify the loss function and choose appropriate content and style representation, which leads to a successfully style transfer. Through this project, we gain insights into how content and style representation affect the image synthesis process, and how the choice of layers affect the efficiency of computation. Moreover, we suggest a new loss function for possible improvements.**

I. Introduction

Deep learning becomes popular and attractive after year 2006 and has been applied in many fields such like face recognition, autonomous vehicles, etc.[5][7] It is pretty easy to conclude that majority of these topics are related to image processing tasks since the Convolutional Neural Networks (CNN) is one of the most powerful tools to do that. Also people start to try some interesting topics which can not be done by real people before with further research in this field. For example, an application called FotoRus suddenly took the top position in the App store due to it enables users to transform your own photos with artistic style effects. However, the image content can not be well separated from the style until Gatys. *et al.* proposed a novel algorithm *A Neural Algorithm of Artistic Style* [3]. Gatys. *et al.* [2] proposed a texture synthesis method using the same idea, which could extract the style representation from an image. *A Neural Algorithm of Artistic Style* could well perform the image style transfer, and also succeeds in separating and recombining the image content and style of natural images.

We first try to implement the algorithm from the original paper, i.e. *A Neural Algorithm of Artistic Style* on our images. We realize their method in TensorFlow and generate acceptable results. But the results are not so satisfied based on our understanding of style transfer. Thus we propose a new loss function, which produces a better result. We guess that their setting does not perform well on our images is due to the inconsistency of the size of our images and their images. The resolution of images used in the original paper is about 512×512. However, images we pick is with the size 160×256, which better matches the requirement of the pre-trained VGG network used, i.e. 224×224. Besides, we also study the separation of style representation from the image based on our loss function. Moreover, we study the effect of choice of the content representation and style representation on the efficiency of the algorithm.

II. Summary of the Original Paper

The paper [3] refer to the feature responses in higher layers of the network as the content representation and obtain a style representation of the input image, which captures its texture information but not the global arrangement by including the feature correlations of multiple layers. All the work were generated on the basis of the VGG-Network [6]. In layer l, there are N_l feature maps each of size M_l, where M_l is the height times the width of the feature map. Matrix $F^l \in \mathbb{R}^{N_l \times M_l}$ is the responses in a layer l where F^l_{ij} is the activation of the i^{th} filter at position j in layer l. Let $\vec{p}$ and $\vec{x}$ be the original image and the image that is generated, P^l and F^l are their feature representation in layer l. The content loss $\mathcal{L}_{content}(\vec{p}, \vec{x}, l)$ is defined as $\frac{1}{2}\sum_{ij}(F^l_{ij} - P^l_{ij})^2$.

The paper built a style representation that computes the correlations between the different filter responses. These feature correlations are given by the Gram matrix $G^l \in \mathbb{R}^{N_l \times N_l}$ where G^l_{ij} is the inner product between the vectorised feature map i and j in layer l, i.e. $G^l_{ij} = \sum_k F^l_{ik}F^l_{jk}$. Let $\vec{a}$ and $\vec{x}$ be the original image and the image that is generated, A^l and G^l are their style representation in layer l. The contribution of that layer to the total loss E_l is $\frac{1}{4N_l^2M_l^2}\sum_{ij}(G^l_{ij} - A^l_{ij})^2$ and the total loss $\mathcal{L}_{style}(\vec{a}, \vec{x})$ is $\sum_{l=0}^{L}\omega_l E_l$, where ω_l are weighting factors of the contribution of each layer to the total loss.

To generate the images that mix the content of a photograph with the style of a painting, let $\vec{p}$ be the photograph and $\vec{a}$ be the artwork. The loss function to minimize is $\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha\mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta\mathcal{L}_{style}(\vec{a}, \vec{x})$. The whole process can be illustrated in Figure 1 from Gatys's paper [4].

The key result of this paper is that the representations of content and style in the Convolutional Neural Network are separable. The paper produces new, perceptually meaningful images by mixing the content and style representation from two different source images (Figure 2 from the paper [3]).

III. Methodology

We slightly modify the loss function from original paper to obtain a better performance for our data. The content loss we used is

$$\tilde{\mathcal{L}}_{content}(\vec{x}) = \frac{1}{2N_lM_l}\sum_{ij}(F^l_{ij} - P^l_{ij})^2$$

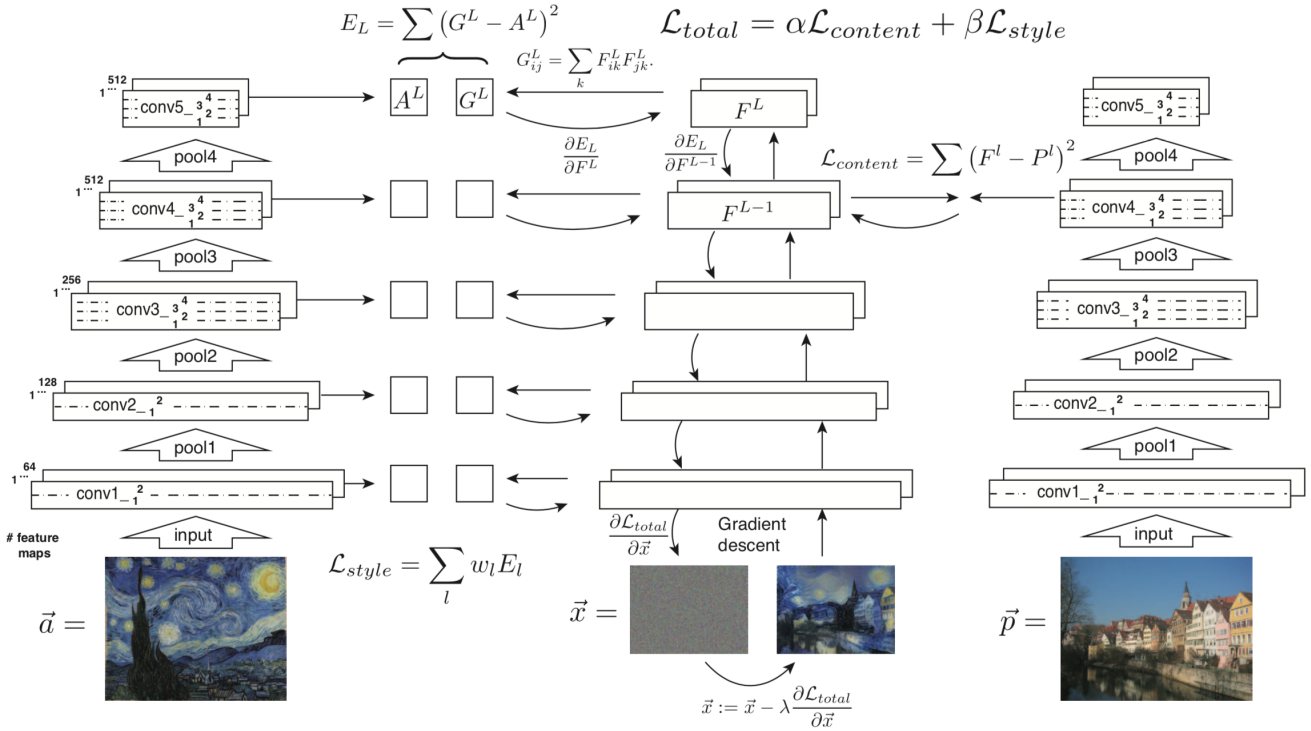


Fig. 1: Style transfer algorithm. First content and style features are extracted and stored. The style image $\vec{a}$ is passed through the network and its style representation A^l on all layers included are computed and stored (left). The content image $\vec{p}$ is passed through the network and the content representation P^l in one layer is stored (right). Then a random white noise image $\vec{x}$ is passed through the network and its style features G^l and content features F^l are computed. On each layer included in the style representation, the element-wise mean squared difference between G^l and A^l is computed to give the style loss $\mathcal{L}_{style}$ (left). Also the mean squared difference between F^l and P^l is computed to give the content loss $\mathcal{L}_{content}$ (right). The total loss $\mathcal{L}_{total}$ is then a linear combination between the content and the style loss. Its derivative with respect to the pixel values can be computed using error back-propagation (middle). This gradient is used to iteratively update the image $\vec{x}$ until it simultaneously matches the style features of the style image $\vec{a}$ and the content features of the content image $\vec{p}$ (middle, bottom).

and the style loss we applied is

$$\tilde{\mathcal{L}}_{style}(\vec{x}) = \sum_{l=1}^{L} \frac{1}{2LN_l^3 M_l^3} (G_{ij}^l - A_{ij}^l)^2$$

The total loss is still the combination of content loss and style loss:

$$\tilde{\mathcal{L}}_{total}(\vec{x}) = \alpha \tilde{\mathcal{L}}_{content}(\vec{x}) + \beta \tilde{\mathcal{L}}_{style}(\vec{x})$$

The reason that we propose a modified loss function is that it seems that we can not obtain a good result if just following their settings. We find that the images used from original paper has a resolution of about 512×512 while VGG network [6] requires the size of input image should be 224×224. The size of the content picture and the style picture will definitely affect the loss function and the quality of the final output.

IV. Implementation

We realize the algorithm in TensorFlow. We use the VGG-19 convolutional neural network [6] as a pre-trained model. This network is trained on more than a million images from the ImageNet database [1] and shows an outstanding performance on image processing. In this project, we only utilize 16 convolutional layers and first 4 pooling layers. For the pooling layers, we use average pooling. The struct of this CNN can be found in Figure 1. Therefore in this VGG network, total 20 layers could be used for content representation and style representation. And we named them by order from low to high: **conv1_1**, **conv1_2**, **poo1**, **conv2_1**, **conv2_2**, **poo2**, **conv3_1**, **conv3_2**, **conv3_3**, **conv3_4**, **poo3**, **conv4_1**, **conv4_2**, **conv4_3**, **conv4_4**, **poo4**, **conv5_1**, **conv5_2**, **conv5_3** and **conv5_4**.

The resolution of content image and style image we used in this project is 160×256, which is much more compatible to the required size of input image of VGG network, i.e. 224×224. We suggest using the style image that has the same size with content image's because during the estimation we compare the style representation of the style image and $\vec{x}$, which has the same size as the content image.

Our algorithm is described in Figure 3. We use Adam for optimizing the loss function. The parameters for Adam used in this project: $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The learning rate we use is 10. The ratio α/β controls the balance between

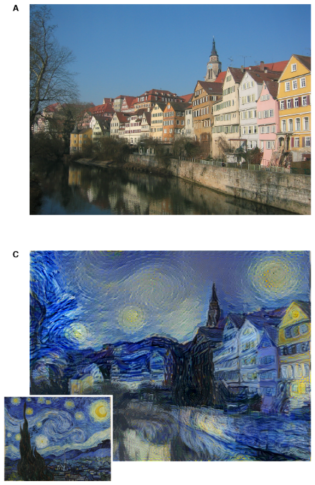


Fig. 2: Image that combines the content of a photograph with the style of a well-known artwork. The original photograph depicting the Neckarfront in Tubingen, Germany, is shown in A (Photo: Andreas Praefcke). The painting that provided the style for the respective generated image is shown in the bottom left corner. The painting is *The Starry Night* by Vincent van Gogh, 1889.

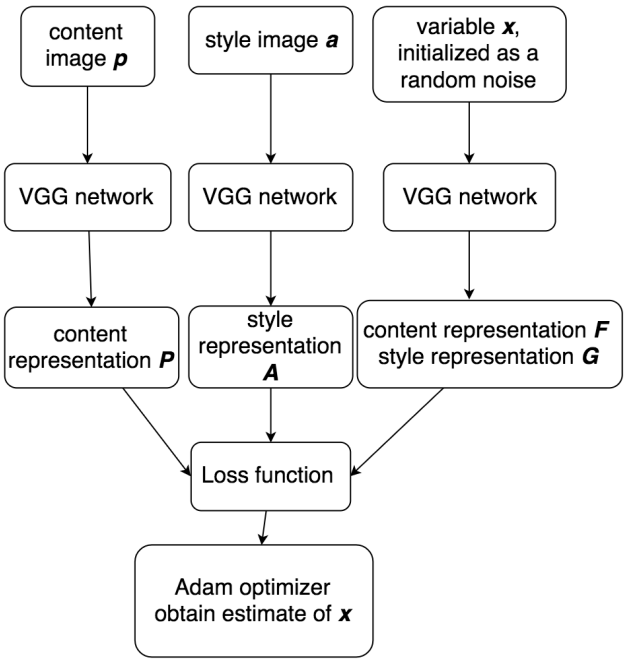


Fig. 3: Flow chart of the algorithm.

the content and style of the output image. The smaller ratio is, the more emphasis is on the style. For example, $\alpha/\beta = 10^{-5}$ appears like the style image while 10^{-1} results in a content-like picture. A reasonable choice of the ratio is 10^{-3} or 10^{-4}.

(a) content (b) style

Fig. 4: Content image & style image.

V. Results

A. Implement original paper's method

The content image we choose is a photograph of Hoover Tower on the Stanford University campus (Figure 4a). The style image we pick is Vincent Van Gogh's famous artwork *The Starry Night* (Figure 4b). The resolutions of both images are the same, i.e. 160×256. The experiments are performed on both the laptop and Columbia's Habanero system. The laptop is equipped with CPU (dual-core 2.5 GHz Intel Core i7). And we use 8 cores CPU on Habanero system to run our code.

We just follow the original paper's setting to perform the style transfer. We study the effect of the style representation and the trade-off between content and style. The results are showed in Figure 5. Here, we use the layer **conv4_2** for content representation. Based on the experiment results, we could say that when including style features from higher layers of the network, the style representation could influence a larger local area. And when decreasing the ratio, the synthetic image would emphasize more on the style part rather than the content part.

Among these synthesised images, the image (Figure 6) generated by the style representation (**conv1_1**, **conv2_1**, **conv3_1**, **conv4_1**, **conv5_1**) with ratio $\alpha/\beta = 10^{-4}$ is the best one.

B. Performance of our loss function

However, we are not so satisfied with the output gained by the original paper's method because the synthetic image (Figure 6) seems not very close to the pattern of the style image (Figure 4b). Then we propose our loss function, which emphasize more on the style effect. The synthetic image generated by optimizing our loss function is Figure 7. It seems that the style of this image is more similar to *The Starry Night* compared with the previous synthetic image (Figure 6).

The whole computing process takes about 16 minutes with 1000 iterations in our laptop. In Figure 8, we show how $\vec{x}$ evolved during the optimization. As we can see, as the number of iterations increase, both the content reconstruction and style reconstruction gradually forms.

C. Separate the style representation

The content reconstruction is much simpler compared with the style reconstruction. In this section, we study the style reconstruction based on our loss function. By the results (Figure 9), we conclude that as including more higher layers for style reconstruction, the generated image could more globally catch the style of the original painting.

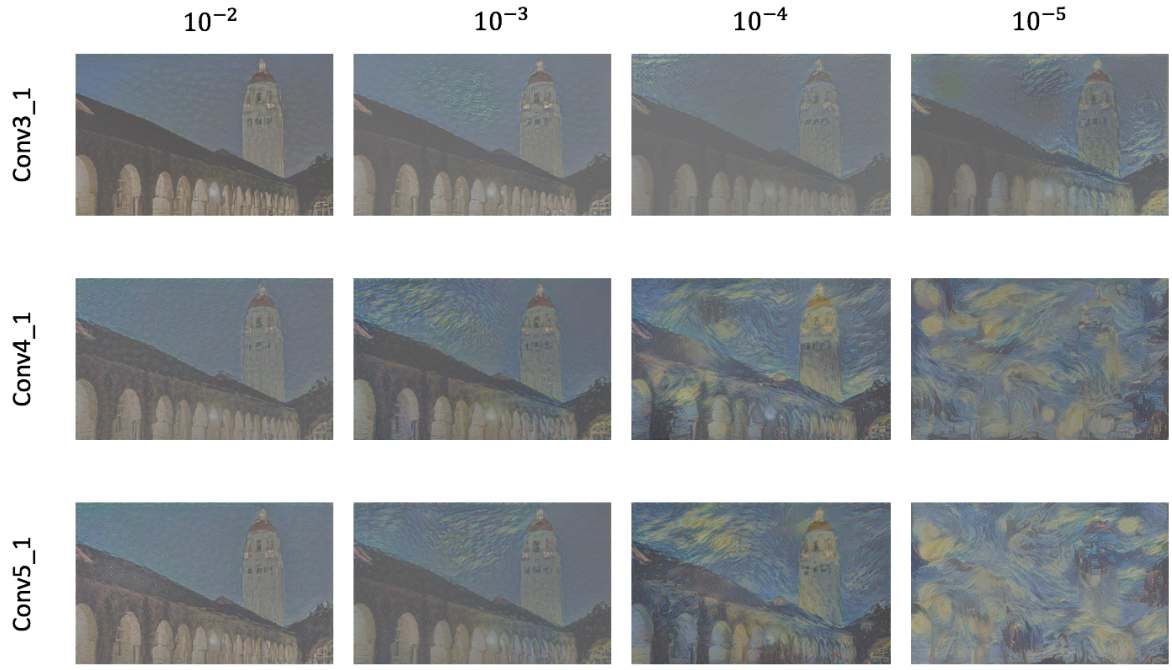


Fig. 5: Detailed results for the style effect. The rows shows the result of matching the style representation of increasing subsets of the CNN layers. The number for each row means the highest layers used for style representation. For example, **conv3_1** means that we use (**conv1_1**, **conv2_1**, **conv3_1**) for the style representation. The columns show different relative weightings between the content and the style reconstruction. The number above each column indicates the ratio α/β between the emphasis on matching the content of the photograph and the style of the artwork.

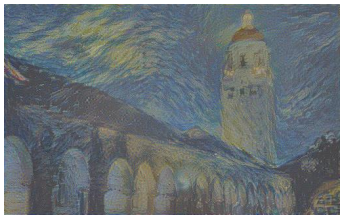


Fig. 6: The synthesised image produced by the original paper's method.

Fig. 7: The synthesised image produced by our method.

D. Efficiency

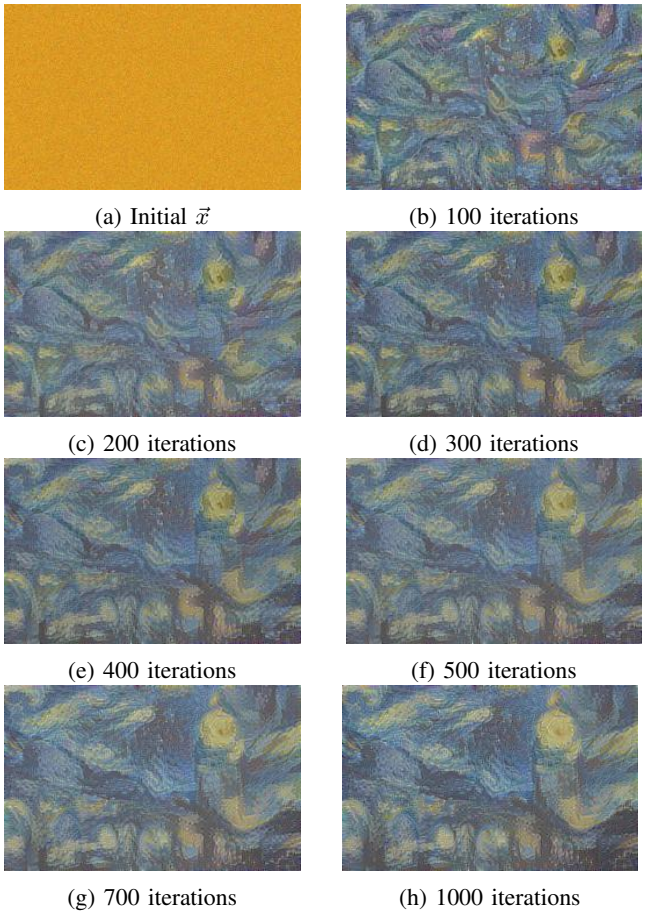
The higher layer we use, the more computing time we need. When we choose **conv4_2** as the content layer and (**conv1_1**, **conv2_1**, **conv3_1**, **conv4_1**, **conv5_1**) as the style layers, the total time for 1000 iterations is about 16 minutes. When the depth of highest layer increases 5, e.g. from **conv3_1** to **conv4_1**, or from **conv4_1** to **conv5_1**, the computing time is double. For the effect of size of images, if we use an image with double width and height, the computing time is also about double.

VI. Discussion

We first apply original paper's method on our images. The results are not satisfied probably due to the inconsistency of the size of the images. Then we propose our loss function, which leads to a better result. We also study the style separation based on our loss function, which plays an crucial part in image style transfer. Considering the consistency of units, we suggest a new content loss $\mathcal{L}^*_{content}(\vec{x})$ for the content reconstruction, i.e. $\frac{1}{2N_l^3 M_l^3}\sum_{ij}(F_{ij}^l - P_{ij}^l)^4$. Comparing with the style loss, they have a similar form, but the performance of this loss function is unknown, which remains to further explore.

VII. Acknowledgement

We would like to thank Professor Zoran Kostic for introducing the deep learning to us. We would also thank all TAs in this course for your sincere help. The experiments made use of Columbia?s Habanero shared research computing facility.



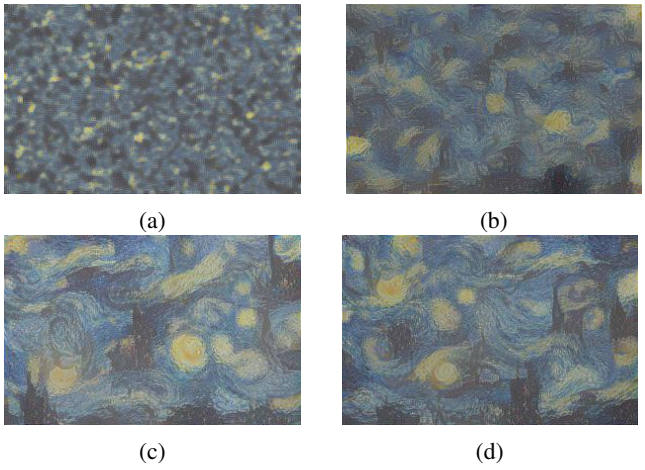
(a) Initial $\vec{x}$ (b) 100 iterations

(c) 200 iterations (d) 300 iterations

(e) 400 iterations (f) 500 iterations

(g) 700 iterations (h) 1000 iterations

Fig. 8: Learning process of $\vec{x}$. Each panel shows what the $\vec{x}$ looks like after some specific number of iterations.



(a) (b)

(c) (d)

Fig. 9: Style reconstruction. We reconstruct the style of the input image from a style representation built on different subsets of CNN layers ((a): (**conv1_1**), (b): (**conv1_1**, **conv2_1**), (c): (**conv1_1**, **conv2_1**, **conv3_1**, **conv4_1**), (d): (**conv1_1**, **conv2_1**, **conv3_1**, **conv4_1**, **conv5_1**)).

REFERENCES

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[2] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 262–270, 2015.

[3] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[4] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[7] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

APPENDIX A
INDIVIDUAL STUDENT CONTRIBUTIONS IN FRACTIONS

	cy2507	fg2425	ys3047
Last Name	Yin	Gao	Shi
Fraction of (useful) total contribution	33 %	34 %	33%
What I did 1	write the report	write code	do experiments
What I did 2	do experiments	analyze the problem	write the report
What I did 3	analyze the problem, contributed to the coding	do experiments	analyze the problem, contributed to the coding