Styfie: Multi-Style Transferred Selfies

Sarvesh Relekar * [1] **Saksham Bassi** * [1] **Rishil Kirtikar** * [1]
[1] Courant Institute of Mathematical Sciences, New York University

Abstract

Introduced in 2015, Neural Style Transfer has become a popular application of Computer Vision and has been utilized to synthesize stylized images. With great successful applications in the field of post-processing photo editing, Style Transfer has proved to be effective. It works on the principle of taking a content image and a style reference image as input, to generate a new image that retains the content of the original image but follows the style of the style reference provided. In this project, we stylize specifically multiple desired portions in an image separately. To find these desired portions in the input images, we train a Segnet model to perform segmentation. With the masks generated, we implement transferring of style to either the object or the background. Extending this methodology, we stylized both object and background separately using different style references. The project effectively showcases the application of combining Image Segmentation and Neural Style Transfer techniques. [1] [2]

1. Introduction

Post-processing photo editing using filters have been very popular for more than a decade. There has been active research in the domain of image synthesis and image editing. With the rapid progress in the field of Deep Learning, Neural Networks have shown increasingly good results in Computer Vision. Mobile applications like Adobe Lightroom, Google Snapseed, Prisma, Picsart provide an Auto-edit feature that edits the picture to enhance it based on Computer Vision techniques. Snapseed has a Portrait feature edit which tries to detect human objects in the image to effectively create Portrait-like photos.

In the field of Computer Vision, from image classification to semantic segmentation, Deep Neural Networks have made many breakthroughs in a fairly short amount of time. The applications of Deep Learning in the field of Computer Vision were revolutionized by the introduction of Convolutional Neural Networks. These advances have paved the way for boosting the use of computer vision in existing domains and introducing it to new ones. In many cases, computer vision algorithms have become a very important component of the applications we use on a daily basis. Convolutional Neural Networks(LeNet) was first used for the task of handwritten digit recognition. They were trained on the MNIST database of handwritten digits and performed extremely well in terms of the accuracy of results on both seen and unseen data. This is because, when they are trained on images, a representation of the image is developed. Due to this, the input image is transformed into representations that care about the actual content of the image rather than its detailed pixel values. Convolutional Neural Networks have a multi-layered architecture that is used to gradually reduce data and calculations to the most relevant set. This set is then compared against known data to identify or classify the data input. Earlier layers in the network learn low-level information in the images such as edges or noticeable patterns. Deeper layers in the network learn the high-level content in terms of objects and the background. This achieves significantly better performance as compared to the traditional Multi-Layer Perceptron due to their strong learning on representation capability.

These advances in Convolutional Neural Networks have paved the way for interesting applications. One important technique in Computer Vision is Style Transfer which involves the transfer of an image style to a different image while preserving the content of the target image. Following this technique, improvements in this domain have led to rendering images with a flavour of famous artistic styles. To obtain a representation of the style of an input image, a feature space originally designed to capture the texture information is used. Reconstructing from the style features that produce the texturized versions of the input image can capture its general appearance in terms of colour and localized structures. This multi-scale representation is referred to as style representation. The advent of creating a digital camera feel of background blur in the photos taken by mobile phones through Deep Learning models has been fascinating

[1] Code: https://github.com/sakshambassi/style-transferred-selfies

[2] Student IDs: sr5796, sb7787, rak9719

to everyone. This is implemented by a technique called Portrait Segmentation. For this, the concept of semantic segmentation is used to predict the label of every pixel in an image.

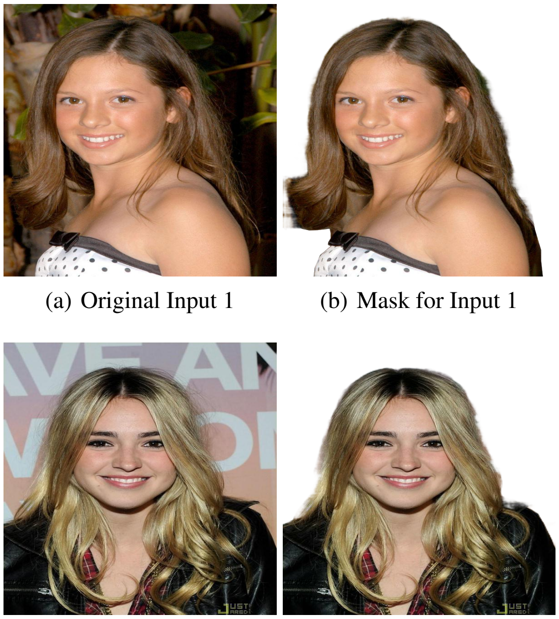
We were motivated to explore this further by fusing Style transfer on images in a multi-locality manner. We try to unite Image Segmentation and Neural Style Transfer techniques in this work. Stylizing of image happens differently in two localities (which are foreground and background) with different styles. The results include multi-style transferred selfie images. This has various real-world applications as this provides the ability to alter specific regions of the image with different styles. This is an interesting problem in non-photorealistic rendering. For Style Transfer, there are traditional methods that use handcrafted features in the form of mathematical representations. Some of these algorithms achieve remarkable results but they have many disadvantages such as failure to capture low-level features. Deep neural networks provide an alternate way where these low-level features are learned.

There are many applications of Style Transfer in photo and video editing. They range from sharing stylized selfies to augmenting user-generated music videos. The Style Transfer models can be easily embedded on mobile devices, allowing for applications that can transform images and videos in real-time. One of the most common use cases for this is in the professional quality photo and video editing applications. These have become widely accessible and easier due to the advancement in deep learning approaches. With the continued improvement of AI accelerated hardware, endless doors are now opened in design, content generation and the development of creative tools.

2. Related Work

Neural Style transfer was introduced by (Gatys et al., 2015) in the research: A Neural Algorithm of Artistic Style which introduces additional learning of style representations using CNNs. Spatial information is preserved in the case of content representation, and in the case of the style representation, all spatial components are removed by averaging across the spatial dimensions. The proposed network uses two losses - content loss and style loss. Content loss is calculated based on the Mean Squared Error (MSE) value between the activations of the target image and content image. Style loss is calculated based on the MSE value between the style image and target image. Deeper layers learn the style effects and their outputs have stylized features/patterns infused. (Jing et al., 2018) reviews various types of style transfers as well as the currently available algorithms for Neural Style Transfer. It focuses on discussing the impact of Convolutional Neural Networks in neural style transfer and other methods for style transfer that do not require CNNs. Another unique method proposed in (Kurzman et al., 2019) of Neural Style Transfer that maps different styles to different object classes in real-time. The method involves generating a segmentation mask of the object class to be styled followed by styling the image globally. The final output is then obtained by combining the segmentation mask and the styled object to obtain styling only for the specific object class and not the whole image. Recent advances in Style transfer include, Stylized Neural Painting (Zou et al., 2020) paper which proposes an image-to-painting translation method that generates painting artworks of the respective input images/photographs. Instead of pixel-wise estimation, the paper deals in the vectorized environment and produce a sequence of physically meaningful stroke parameters that can be further used for rendering. The work improves their results by stylising the generated image(painting) with the input style image provided. (Chen et al., 2019) describes a method to extract information selectively within a boundary area to make high-quality segmentation output in real-time but faces issues in segmenting portraits within multi-person shoots and single-person shots with occlusion due to having a lesser number of semantic channels. (Badrinarayanan et al., 2016) proposes a novel encoder-decoder model which upsamples encoder output which involves storing the max-pooling indices used in the pooling layer. The decoder generates a well-segmented output image. The model is reasonably good in performance and is space-efficient. The encoder is based on VGG16 architecture with only forward connections. This leads to very few parameters with initially trained weights. The last decoder layer has a softmax classifier which classifies each pixel value. In (Liu et al., 2017), a novel approach is introduced for neural style transfer that integrates depth preservation as additional loss, preserving overall image layout while performing style transfer. (Den) explores another new approach to fully automatic colour image segmentation, called JSEG. Initially, the colours in the image are quantized to represent several classes that can be used to differentiate regions in the image. Then, image pixel colours are replaced by their corresponding colour class labels, thus forming a class-map of the image. A criterion for “good” segmentation using this class-map is proposed. Applying the Criterion to local windows in the class-map results in the “J-image”, in which high and low values correspond to possible region boundaries and region centres, respectively. A region growing method is then used to segment the image based on the multi-scale J-images. Experiments show that JSEG provides good segmentation results on a variety of images.

In (Zhang et al., 2013) a style transfer algorithm is proposed via a novel component analysis approach, based on various image processing techniques. First, inspired by the steps of drawing a picture, an image is decomposed into three components: draft, paint, and edge, which describe the content, main style, and strengthened strokes along the boundaries. Then the style is transferred from the template image to the source image in the paint and edge components. A coarse-to-fine belief propagation algorithm is used to solve the optimization problem. To combine the draft component and the obtained style information, the final artistic result can be achieved via a reconstruction step. Neural style transfer has recently received significant attention and demonstrated amazing results. An efficient solution proposed by (Johnson et al., 2016) trains feed-forward convolutional neural networks by defining and optimizing perceptual loss functions. Such methods are typically based on high-level features extracted from pre-trained neural networks, where the loss functions contain two components: style loss and content loss. However, such pre-trained networks are originally designed for object recognition, and hence the high-level features often focus on the primary target and neglect other details.


(a) Original Input 1 (b) Mask for Input 1

(c) Original Input 2 (d) Mask for Input 2

Figure 1. Input to SegNet for Training

3. Methods

Image Models are the most popular techniques used in applications such as Image Classification, Object Detection, Portrait Segmentation and many others. Among Image models, Convolutional Neural Networks are the most widely used, however, newer models such as Vision Transformers are also becoming popular due to better performance and accuracy in these tasks. A powerful algorithm used in self-driving cars by companies such as Tesla is based on image segmentation architecture called SegNet. The method of portrait segmentation involved in this work also uses SegNet architecture.

3.1. SegNet

For the portrait segmentation task, we have used the SegNet model architecture. SegNet has an encoder and a corresponding decoder network, followed by a final pixel-wise classification layer. The Encoder network consists of 13 convolutional layers. The number of parameters in the SegNet encoder network is low as compared to other recent architectures. There is a decoder layer present for each corresponding encoder layer. So, the decoder network also has 13 layers. The final decoder output is fed to a multi-class softmax classifier to produce class probabilities for each pixel independently.

Input provided to the SegNet is in two forms, namely a normal clicked image and its corresponding masked version. The masked image will have the pixels containing the human torso having the same values as the original image, whereas all other pixels will have white colour values. Figure 1 demonstrates the 2 types of input (2 sample pairs) provided to our segmentation model. The Encoder network performs convolution with a filter bank to produce a set of feature maps which are then batch normalized and an element-wise ReLU is applied. This is followed by max-pooling with a 2x2 window and stride of 2. SegNet stores only the max-pooling indices i.e the locations of maximum feature value in each pooling window is memorised for each encoder map. The main advantages of such an approach are improved boundary delineation and fewer parameters.

The Decoder network up-samples its input feature map using the memorized max-pooling indices from the corresponding encoder feature maps. These features maps are convolved with a trainable decoder filter bank to produce dense feature maps. The high dimensional feature representation at the output of the final decoder is fed to a trainable softmax classifier which classifies each pixel.

3.2. Neural Style Transfer

The model used is based on the VGG-Network, a convolutional neural network that rivals human performance on a visual object recognition benchmark task. From our architecture, we have used the feature space provided by the 16 convolutional layers and 5 pooling layers of the 19 layer VGG-Network. The input image is encoded in each layer of the network by the filter responses to that image. Gradient descent is performed on a white noise image to find another image that matches the feature responses of the original image. Then the squared error loss is defined between the feature representations. A style representation is built, on top of each layer of the network, to compute the correlations

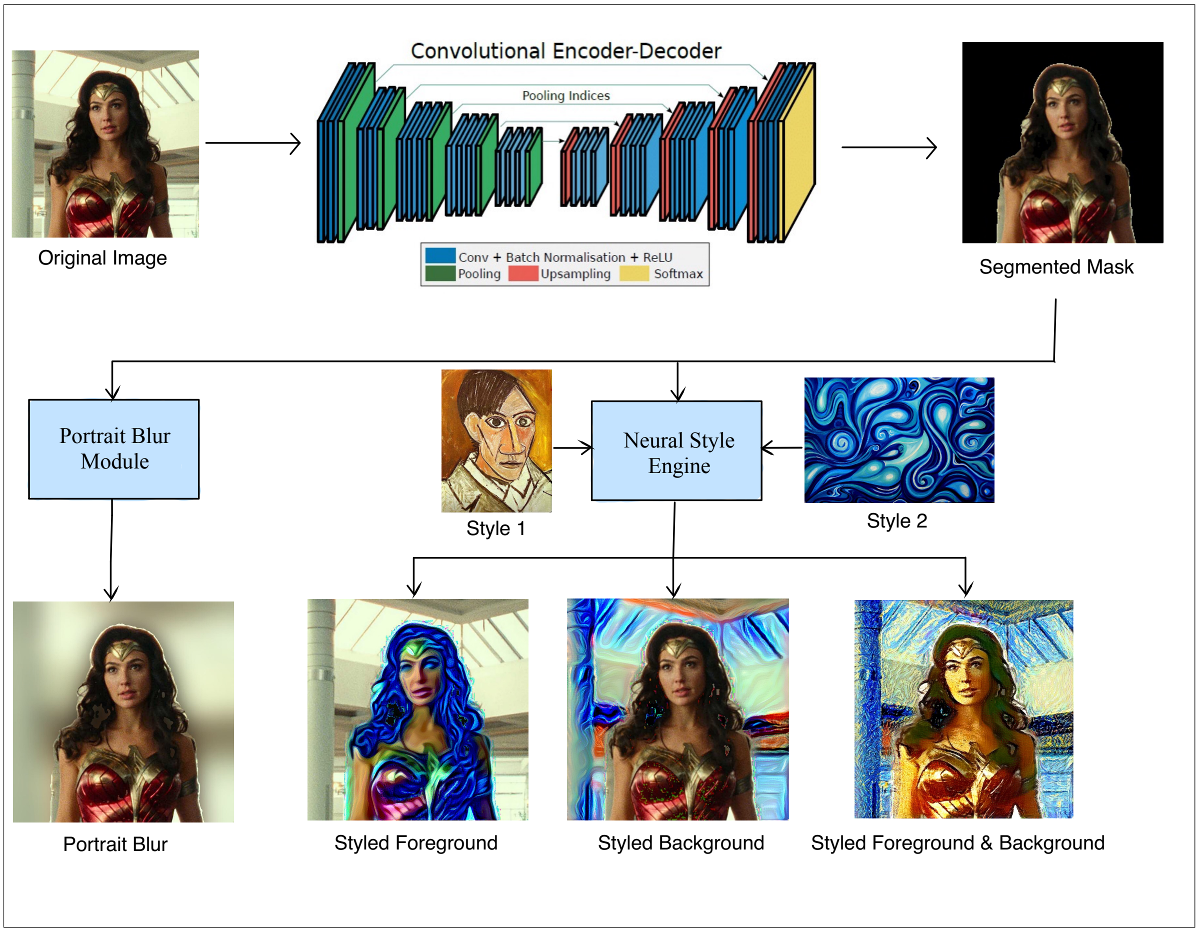


Figure 2. Styfie's System Flow

between the different filter responses. The expectation is taken over the spatial extent of the input image.

Gradient descent from a white noise image to find another image is used to generate a texture that matches the style of a given image. For this, the mean squared distance is minimized between the entries of the Gram Matrix from the original image and that of the image to be generated. The distance of a white noise image from the content representation of the photograph and the style representation of the painting is minimized.

4. Algorithm

The overall system flow of our work is illustrated in Figure 2. The input image is loaded into memory and resized to the required input dimensions of $256 \times 256 \times 3$. It is then provided to the trained portrait segmentation model which generates the segmented mask of the input. The segmented image is provided as a mask to our Neural Style Engine and Portrait Blur Module. Neural Style Engine performs style transfer on the original image using the original image as well as its portrait segmented mask. It also takes in the input of two different style images and generates three types of style transferred outputs comprising of style transfer applied to only the foreground, only the background and two different styles applied to both the foreground and background simultaneously. The Neural Style Engine works on three types of loss functions: style loss, content loss and total variation loss. The style loss maintains the style of the reference(style) image in the generated image. The content loss works on maintaining the content of the base image in the generated image. Total-variation loss is designed to keep the generated image locally coherent. VGG-16 model with pre-trained weights on Imagenet is used to predict features of background and foreground masks of the image separately, along with features predicted on the base image. Total Loss is calculated by adding style losses from all the feature layers and adding total variation loss and content loss. The Style Transfer model over the iterations minimizes this loss using `scipy.optimize` which uses the L-BFGS-B algorithm (shown in Algorithm 1). The Portrait Blur module is essentially implemented background blurring using Gaussian smoothing. It takes the input of the original image and its portrait segmented mask and generates the portrait blurred variation of the original image.

Algorithm 1 mask_neural_optimizer

```
Require: Input: base_image_path $p$
  $x$ = preprocess_image($p$)
  for $i$ in $iterations$ do
      $x, min\_val$ = scipy.fmin_l_bfgs_b($evaluator.loss$, $x$.flatten(), $fprime = evaluator.grads$)
      $img$ = inverse_process_image($x$.copy())
      imageio.imwrite($fname$, $img$)
  end for
```

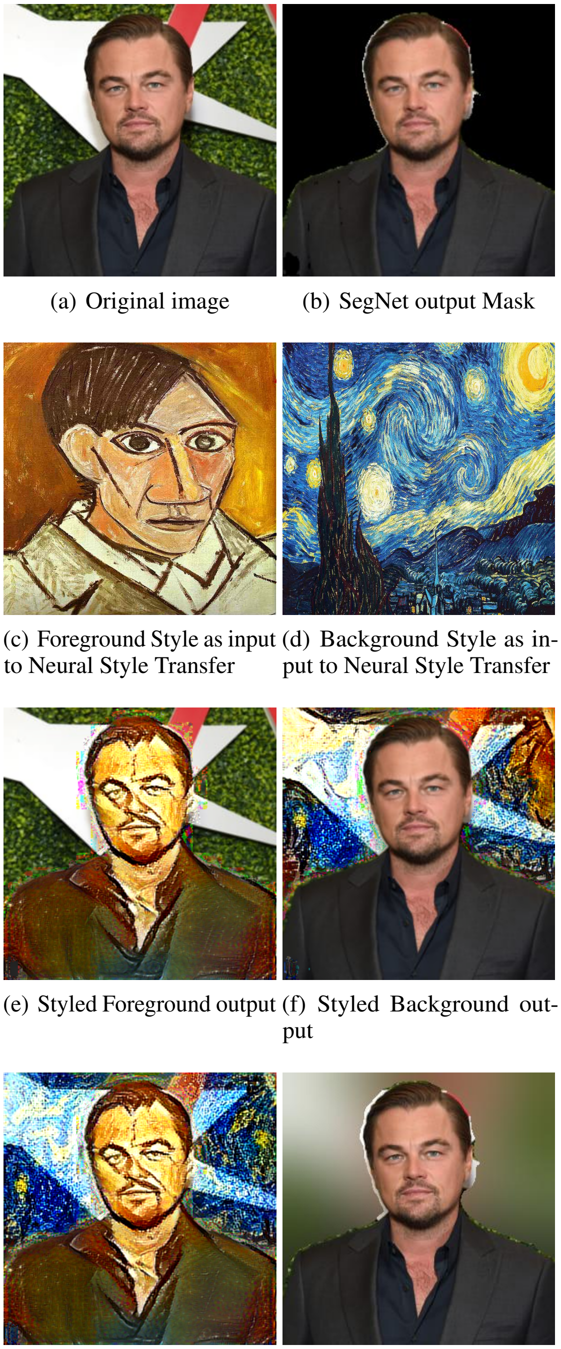
Algorithm 2 preprocess_image

```
Require: Input: image_path $p$
  $img$ = load_img($p$, $target\_size$ =($nrows$, $ncols$))
  $img$ = keras.img_to_array($img$)
  $img$ = np.expand_dims($img$, $axis = 0$)
  $img$ = vgg16.preprocess_input($img$)
  return $img$
```

5. Experiments

We used the Matting Human Datasets by AiSegment.com[3] to train the SegNet model for the portrait segmentation task. A few examples of the input to the model in the form of the original image and its segmented mask have been provided in Figure 1. 10,000 images and their corresponding segmented masks were selected at random from a total of 34,427 images in the data set. Each image was reshaped into dimensions stated in the algorithm. The SegNet model has a total of 89 million parameters which comprise 88 million trainable and 23 thousand non-trainable ones. The model has trained over 10 epochs with each epoch requiring between 7.5 minutes to 10.4 minutes per epoch. The average training time per epoch was 8.4 minutes. We used the Adam optimizer for gradient descent. The loss function used for this task is the Binary Cross Entropy function and the metric used is the Mean Intersection Over Union as the metric to gauge the performance of the model.

Due to the model's huge depth (36 layers), we trained the model on NYU's SLURM HPC environment. The entire training process took around 1 hour to complete, utilizing one node and five CPUs per task. The data set size exceeds 25 GB in storage. Loading the entire data set into memory, even after using generator functions was achievable only after using a minimum of 16 GB of RAM. Due to the large requirement of computational power to calculate the gra-



(a) Original image

(b) SegNet output Mask

(c) Foreground Style as input to Neural Style Transfer

(d) Background Style as input to Neural Style Transfer

(e) Styled Foreground output

(f) Styled Background output

(g) Multi-Style Transfer output

(h) Portrait Blur output

Figure 3. Input and Output Examples of SegNet and Neural Style Transfer

[3] Data Set Link: https://www.kaggle.com/laurentmih/aisegmentcom-matting-human-datasets

Metric	Binary Cross-Entropy Loss	Mean IoU
Training	0.065	0.300
Validation	0.109	0.308

Table 1. Metrics of trained SegNet model on Training and Validation data

dients during the backpropagation step for each layer, we used an NVIDIA Tesla V100 GPU with 32 GB of VRAM and 640 tensor cores to speed up the training of our model.

Upon training, the model achieves a Binary Cross-Entropy loss of 0.1092 and Mean IoU of 0.3082 on the validation data as referenced in Table 1. Initially, the decrease in validation loss is rapid as indicated in Figure 4 but tends to slow down in later epochs. The plot for Mean Intersection over Union shows an increasing trend with each epoch as noted in Figure 5.

Figure 3 shows an example of our input and final output. The SegNet model takes as input the original image as shown in Figure 3 (a), and provides as output its corresponding segmented mask as show in Figure 3 (b). The mask as well as the original image are forwarded to our Neural Style Engine which also takes as input two different styles such as the ones shown in Figures 3 (c) and 3 (d). The Neural Style Engine then performs multi-locality style transfer separately on both the foreground and background of our original input image as seen in Figures 3 (e) and 3 (f) and together on both the foreground and background in Figure 3 (g). The Portrait Blur module also works in parallel with our Neural Style Transfer Engine, taking the same inputs minus the foreground and background style images and outputs a portrait blurred version of the original image as show in Figure 3 (h).

6. Discussion

Our Neural Style Engine produces robust results for all the specified style transfer tasks. It manages to successfully apply for multi-localized style transfer on the foreground and background of any input image with different styles. It showcases the application of CNN based models to generate multiple cases of style transferred outputs. The usage of a portrait segmented mask to apply distinct styles in the localities of an image has a variety of applications in real life, used in various mobile photo-editing apps. Further enhancements on the segmentation technique can refine the masked output. This will lead to having a more refined Neural Style Transfer as the desired region (human torso in this case) will be better segmented. Hence, the foreground locality and the background locality of the input image will be separated to a more finer level.

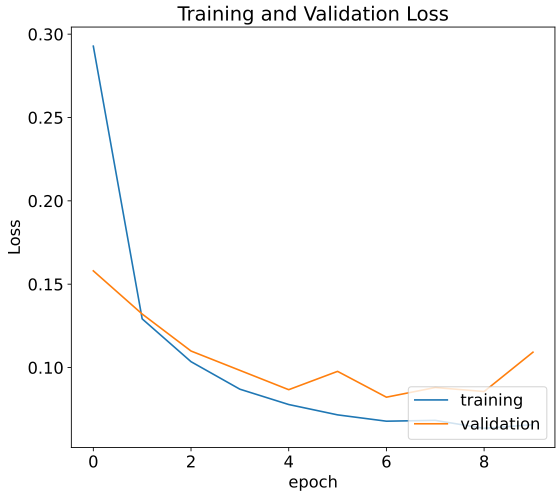


Figure 4. Plot of Binary Cross-Entropy Loss

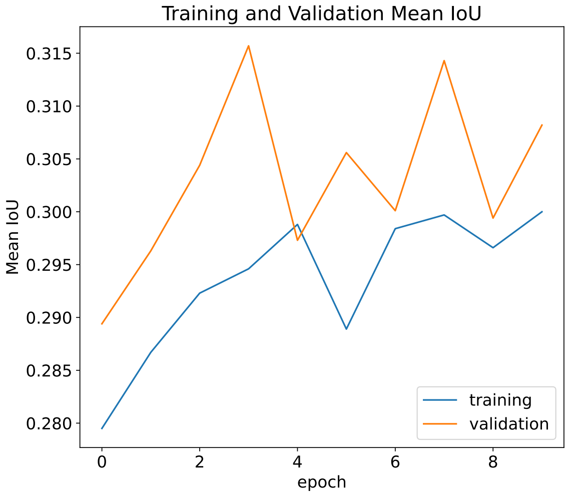


Figure 5. Plot of Mean Intersection over Union

7. Acknowledgement

This work was part of the Computer Vision-CSCI-GA.2271-001 course at New York University in Fall 2021. We would like to thank Prof. Rob Fergus for providing us with the opportunity to work on this project and amazing lectures. We are also very thankful to New York University for allowing us to utilise their HPC environment for training the deep learning models involved in our project.

References

Badrinarayanan, V., Kendall, A., and Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation, 2016.

Chen, X., Qi, D., and Shen, J. Boundary-aware network for fast and high-accuracy portrait segmentation, 2019.

Gatys, L. A., Ecker, A. S., and Bethge, M. A neural algorithm of artistic style, 2015.

Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., and Song, M. Neural style transfer: A review, 2018.

Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016. URL http://arxiv.org/abs/1603.08155.

Kurzman, L., Vazquez, D., and Laradji, I. Class-based styling: Real-time localized style transfer with semantic segmentation, 2019.

Liu, X.-C., Cheng, M.-M., Lai, Y.-K., and Rosin, P. L. Depth-aware neural style transfer. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, NPAR '17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450350815. doi: 10.1145/3092919.3092924. URL https://doi.org/10.1145/3092919.3092924.

Zhang, W., Cao, C., Chen, S., Liu, J., and Tang, X. Style transfer via image component analysis. *IEEE Transactions on Multimedia*, 15(7):1594–1601, 2013. doi: 10.1109/TMM.2013.2265675.

Zou, Z., Shi, T., Qiu, S., Yuan, Y., and Shi, Z. Stylized neural painting, 2020.