Depth-aware and Laplacian-steered Instance Style Transfer

Lingjun Zhao
University of Michigan
Ann Arbor, USA
lingjunz@umich.edu

Junkai Zhang
University of Michigan
Ann Arbor, USA
junkaiz@umich.edu

Abstract

Neural Style Transfer (NST) based on Convolutional Neural Network (CNN) has recently achieved amazing results and received significant attention. Semantic Segmentation based on Deep Learning has also demonstrated great performance in image segmentation field. We are interested in combining Style Transfer with Fully Convolutional Network (FCN) to make Image Style Transfer be instance-aware. Additionally, in the traditional optimization objective, low-level features and depth information of the content image are absent, leading to the loss of many depth and detailed information and hence decreasing the artistic quality of the stylised image. As a remedy, we propose to introduce two loss functions into the image synthesis: Laplacian Loss and Depth Loss. By incorporating these loss functions, we obtain a new optimization objective for the Instance Style Transfer. Experiments show that such strategy produces more appealing stylised image with more depth and detailed information.

1. Introduction

Do you want Pablo Picasso to paint your favorite picture? Style Transfer can help to realize your dream, mapping the style of a source image onto a target image. With the development of Deep Learning, Gatys et al. [5] first developed CNN to apply famous painting styles to natural images, which opened up a new field called Neural Style Transfer.

Among all the research areas in Neural Style Transfer, we are interested in Instance Style Transfer [10], which is built on semantic segmentation and aims to stylize a single user-specified object within an image. In this way, we are able to stylize some specific objects within one image using different amazing styles.

However, if we just substitute the original object with the stylised object within an image, the detail and depth information of the original object will be lost due to the limit sensing capacity of the low-level CNN which we use to extract the content image. In order to improve the quality of the stylised image, we propose to introduce a depth loss and a Laplacian loss to the overall loss function. By doing this, we are able to stylize a specified instance within a natural image while keeping its depth information and edge details, as shown in Figure 1.

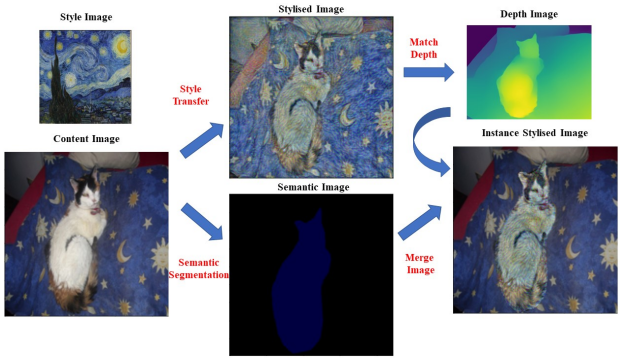

Figure 1. Depth-aware and Laplacian-steered Instance Style Transfer includes three parts: Style Transfer, Semantic Segmentation and Image Merging.

2. Related Work

2.1. Semantic Segmentation

Before the development of deep learning (DL) model, many algorithms have been proposed for the task of image segmentation. Some of the earliest methods include region growing [19], active contours [12], and Markov random fields [20] were developed.

In the past few years, DL model greatly improved the performance of segmentation task [18]. Fully Convolutional Network (FCN) was one of the first DL models implemented in Semantic Segmentation region. Proposed by Long *et al.* [22], FCN was the first end-to-end predictor that outputs pixel-wise dense prediction for arbitrary-sized inputs and is trained from a supervised pre-training. Besides FCN there are other recent segmentation models. First, there are Encoder-Decoder-based models, including Seg-

Net [1], U-Net [25]. SegNet uses max-pooling indices in the encoder process to improve the resolution of the final result [1]. U-Net proposed by Ronneberger *et al.* [25] is inspired by FCN and is developed for medical purpose. U-Net is achieved through the use of skip connections, which allow the network to "skip" over multiple layers and directly concatenate lower-level features with higher-level ones, which helps to preserve the spatial resolution of the input image. HRNet [26] is another popular model for image segmentation that has recently been developed. It differs from SegNet and U-Net, in that it is designed to recover high-resolution representations throughout the encoding process. To do this, HRNet connects high-resolution convolution streams in parallel and repeatedly exchanges information across resolutions. This allows HRNet to maintain high-resolution representations throughout the encoding process, rather than just recovering them at the end. There are also other models based on different architectures that we list in following but do not show more details here. Multi-Scale and Pyramid Network Based Models, including Feature Pyramid Networks(FPN) [16], Pyramid Scene Parsing Network [27]. Recurrent neural network based models, including ReSeg [23], and Graph LSTM [15]. Attention based models, including Reverse Attention Network [9].

2.2. Neural Style Transfer

Neural Style Transfer was first proposed in [5], in which a deep CNN was used to transfer the style of a style image onto a content image. The authors of [6] modify the transfer algorithm to be color preserving. In [11], a simplified loss function to compute the similarity between images is used to speed up style transfer. Recently, the authors in [13] use the pre-trained text-image embedding model of CLIP to realize the modulation of the style of content images only with a single text condition. With the development of Transformer, in [24], a novel Style Transformer network is proposed to break both content and style images into visual tokens to achieve a fine-grained style transformation.

Related to our work, in [2], the authors present a method for targeted style transfer that simultaneously segments and stylizes single objects selected by the user. In order to improve the artistic quality of the stylised image, the authors in [17] Introduce a depth loss term which can make the stylised image possess the similar depth information as the content natural image. In [14], the authors add a Laplacian loss term which can make the stylised image possess more details than before.

3. Method

3.1. Semantic Segmentation

In the original version of the FCN, Long *et al.* [22] used VGG16, AlexNet, and GoogLeNet as backbones. In the recent years, many other DL networks have been proposed with better performance, including Resnet [8], Xception [3], and DarkNet [21]. After considering limited computational resource and overall performance, we ultimately selected Resnet-34 as our backbone network of FCN. For our FCN, we first fine-tuned the pre-trained Resnet34 network on PASCAL VOC 2012 [4]. Next, we added upsampling layers on top of the Resnet34 backbone. These upsampling layers were initialized with Bilinear Interception and were trained to further refine the features for image segmentation. FCN-8s was implemented because of its better performance compared with FCN-32S and FCN-16S. 8, 16, and 32 refers to the upsampling rate after combining the intermediate output from Resnet. Detailed network structure is illustrated in Figure 2

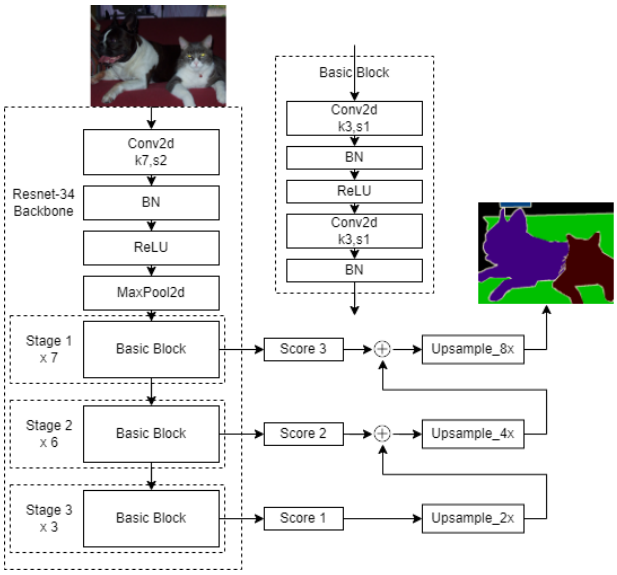

Figure 2. FCN structure. Score1, 2, and 3 are Conv2d(512,21,1,1), Conv2d(256,21,1,1), Conv2d(128,21,1,1) that decreases the dimension of Resnet output to number of class, which is 21 in our case. Upsample_8x,_4x,_2x are ConvTranspose2d(21,21,16,8,4), ConvTranspose2d(21,21,4,2,1), ConvTranspose2d(21,21,4,2,1)

Finally, we added a pixel-wise softmax layer to the network, which produced a per-pixel probability map indicating the likelihood of each pixel belonging to a particular class. During training, we used the cross-entropy loss function to optimize the network parameters and the SGD optimizer. We also applied data augmentation techniques such as random crops to increase the diversity of the training data.

We evaluated the performance of our FCN on a held-out test set using common metrics for image segmentation such as mean intersection-over-union (IoU), frequency-weighted average accuracy, and mean pixel accuracy. Our results showed that our FCN achieved competitive performance on

the task of image segmentation, demonstrating the effectiveness of using Resnet34 as the backbone network.

3.2. Neural Style Transfer

Given a content image x_c and a style image x_s, the loss function is to measure the similarity of these two images. Frequently the loss function is composed of two terms: content loss $\mathcal{L}_c$ and style loss $\mathcal{L}_s$. In order to make the stylised image possess more details and information of the content image, we introduce depth loss $\mathcal{L}_d$ and Laplacian loss $\mathcal{L}_{Lap}$. Thus the total loss function consists of four different loss terms, as shown in Equation 1. α, β, θ and γ represent the weights for content loss, style loss, depth loss and Laplacian loss respectively.

$$\mathcal{L}_x = \alpha\mathcal{L}_c + \beta\mathcal{L}_s + \theta\mathcal{L}_d + \gamma\mathcal{L}_{Lap} \tag{1}$$

The content loss encourages the stylised image to match the scene structure of the content image. This loss is computed as the squared l_2 distance between two convolutional feature maps. Given a feature map of input image F_x and the feature map of content image F_c , both of shape (C, H, W), the content loss $\mathcal{L}_c$ is calculated as Equation 2.

$$\mathcal{L}_c = \sum_{c,i,j}(F^2_{c,i,j} - F^x_{c,i,j})^2 \tag{2}$$

The style loss encourages the texture of the stylised image to match the style image. We compute a weighted and squared l_2 distance between Gram matrices for several layers of the network. Given a feature map F of size (C, H, W), the Gram matrix G computes the sum of products between channels. Then we compare the stylised image's Gram matrix with that of the style image. Define the Gram matrix of input image feature map and style image feature map of at the l^{th} layer as $G^{x,l}$ and $G^{s,l}$, and the weight of the layer as w^l. Style loss of each layer can be computed and the total style loss $\mathcal{L}_s$ is a sum of over all style layers, as shown in Equation 3.

$$\begin{cases} G_{k,l} = \sum_{i,j} F_{k,i,j}F_{l,i,j} \\ \mathcal{L}^l_s = w^l \sum_{i,j}(G^{x,l}_{i,j} - G^{s,l}_{i,j})^2 \\ \mathcal{L}_s = \sum_l L^l_s \end{cases} \tag{3}$$

The depth loss function is used to measure the depth differences between the stylised image and the content target image. In order to preserve maximum depth information and potential structural features, we take the outputs of the depth estimation network and compute the distances as the depth loss. MiDaS is a network which computes relative inverse depth from a single image [7]. We choose to use MiDaS to learn the depth representations of the content image and the stylised image, and then compute the depth loss $\mathcal{L}_d$ as the squared and normalized Euclidean distance between depth representations, as shown in Equation 4.

$$\mathcal{L}_d = \sum_{i,j}(D^c_{i,j} - D^x_{i,j})^2 \tag{4}$$

The Laplacian loss is defined as the mean-squared distance between the two Laplacians. The Laplacian loss is computed by a small two-layer fixed CNN which includes an average pooling layer and a specified convolutional layer. The former layer smooth the input image which can make the Laplacian loss better reflect its true detail structures. The latter layer combines a Laplacian operator to detect the edges of the content image. In addition, we compute the Laplacian of an image on RGB channels, which means the Laplacian value is the sum of the three Laplacians. Finally, given the content image x_c and the stylised image x, we can compute the Laplacian loss $\mathcal{L}_{Lap}$ to measure the difference between their Laplacians, as shown in Equation 5.

$$\begin{cases} D(x) = D(x^R) + D(x^G) + D(x^B) \\ \mathcal{L}_{Lap} = \sum_{i,j}(D(x_c) - D(x))^2_{i,j} \end{cases} \tag{5}$$

3.3. Image Merging

Based on the results from Semantic Segmentation and Neural Style Transfer, we can perform Image Merging to realize the Instance Style Transfer. The strategy is to substitute the original instance pixels with the stylised instance pixels on the natural content image while keeping the background image pixels. There are two issues about the process of Image Merging. One is that an adaptive and reasonable style image should be chosen to match the content image. The other issue is that the boundary between the stylised and non-stylised regions ought to be processed carefully to ensure the smoothness of the whole stylised image.

4. Experiments

4.1. FCN Training

Optimization We train by SGD with fixed learning rate of 10^{-2} and weight decay of 10^{-4}. We choose the mini-batch size of 8, 16, ans 32 images. The result shows mini-batch size of 8 images with better performance.

Fine-tuning We train the whole model based on the pre-trained weight of Resnet-34. The weights of upsampling layers are initialized using Bilinear Interception.

Training Data We train our model on PASCAL VOC 2012 dataset. We also try to train on a larger data set named PASCAL 2012 Aug, which combines Semantic Boundaries Dataset (SBD) [7] and contains 11355 images. This yielded no noticeable improvement, which might result from the unbalanced label classes (too much background labels).

Implementation All models are trained and tested with Colab on NVIDIA Tesla T4.

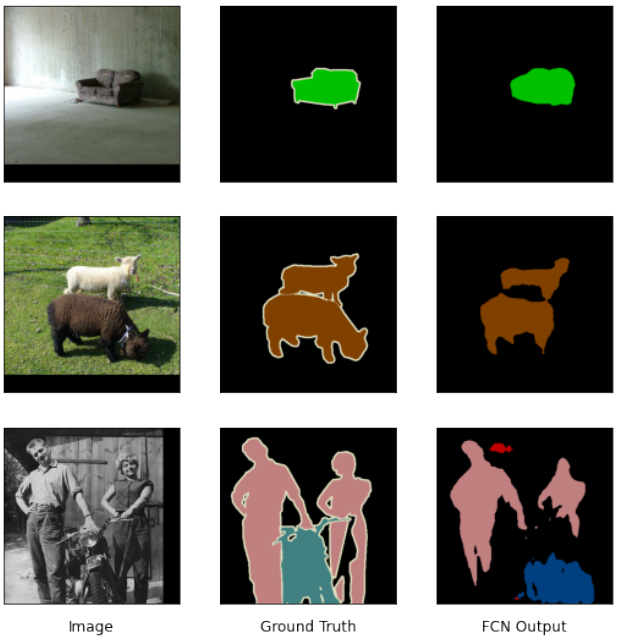

Figure 3. FCN performance on VOC 2012. The left column shows the original picture, the middle column shows the ground truth label, the third column shows the output pixel-wise label from FCN.

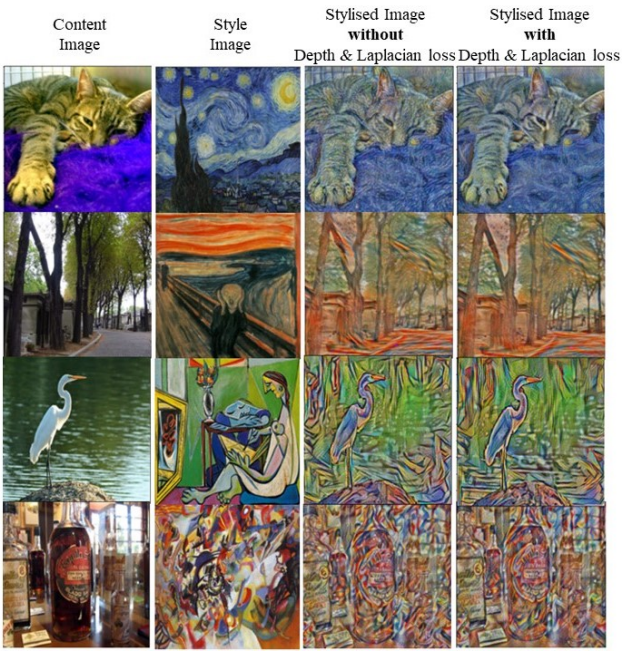

Figure 4. NST results from the original NST and the NST with depth and Laplacian loss terms. The latter stylised images are much more appealing than the previous images.

4.2. FCN Result

Metrics We use same metrics in [22], including pixel accuracy and Mean IoU. Confusion Matrix N is used to simplify the calculation. Let n_{ij} be the i-th row and j-th column of N. n_{ij} represents number of pixels of class i predicted to belong to class j. Using the confusion matrix N, the detailed calculation of accuracy is shown below:

- Pixel Accuracy: $\sum_i n_{ii} / \sum_i t_i$
- Mean IoU: $(1/n_{\text{cl}}) \sum_i n_{ii} / \left(t_i + \sum_j n_{ji} - n_{ii}\right)$

where n_{cl} is the number of classes, and $t_i = \sum_j n_{ij}$.

The pixel accuracy of FCN-8s is 0.8895 and mean IoU is 0.5831. Some comparison between output of FCN and ground truth label is shown in Figure 3.

4.3. NST Training

Optimization We train by Adam with a learning rate of 1, weight decay of 10^{-1} and the epoch size of 200.

Fine-tuning We choose a pretrained VGG19 to extract the features of the content image and the style image. Conv4-1 layer is to extract the semantic information of the content image with a fixed content weight of 1. Conv1-1, Conv2-1, Conv3-1, Conv4-1 and Conv5-1 layers are to extract the texture features of the style image with style weights of 10^8, $0.8 \cdot 10^7$, $0.5 \cdot 10^6$, $0.3 \cdot 10^6$, $0.1 \cdot 10^6$ respectively. MiDaS and Laplacian Net are used to extract the depth and detail information of the content image and the stylised image with fixed weights of 10^4 and 10^2 respectively.

Training Data We use a series of famous painting images as our artistic style, and the SBD Dataset (Augemented VOC2012 Dataset) as our content image.

Implementation All models are trained and tested with Colab on NVIDIA Tesla T4.

4.4. NST Result

We compare the results of the original NST in [5] and the NST with depth and Lapalacian loss terms, as seen in Figure 4. By introducing the depth and Laplacian loss terms, the stylised image demonstrate more stereo and edge details than the original one.

4.5. Image Merging Result

Combining techniques in Semantic Segmentation and Neural Style Transfer, we perform the Image Merging on the results from above, and the stylised image results show that such kind of Instance Style Transfer skill is be able to make the natural images more appealing and amazing, as shown in Figure 5.

5. Conclusions

By combining the Resnet-34 with FCN, we improve the semantic segmentation accuracy of our model. The intro-

Figure 5. Instance Style Transfer results.

duction of depth and Laplacian loss in NST helps to maintain the spatial structure and three-dimensional properties of the content image, while preserving fine details and textures in the generated image. These losses enhance the texture detail of the object in the foreground, making it more distinct from the background and resulting in better visual results. By combining with FCN, which is able to recognize objects in images, NST is able to create interesting collage-style images.

In the aspect of future work, we are aware of the poor merging performance on the boundary between the stylised region and the non-stylised region. We consider there are two main reasons for this problem. One is that the precision of Semantic Segmentation is inadequate, and the other is that no special skill such as smoothness processing is applied to the boundary. Consequently, for future work, we are planning to improve the predicting accuracy of Semantic Segmentation by introducing some better neural networks or training on a larger dataset. Additionally, we consider to introduce Markov Field to the Image Merging problem to make the boundary more smooth and appealing.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:2481–2495, 12 2017. 2

[2] Carlos Castillo, Soham De, Xintong Han, Bharat Singh, Abhay Kumar Yadav, and Tom Goldstein. Son of zorn's lemma: Targeted style transfer using instance-aware semantic segmentation. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 1348–1352, 6 2017. 2

[3] François Chollet. Xception: Deep learning with depthwise separable convolutions. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:1800–1807, 10 2016. 2

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 2

[5] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:2414–2423, 12 2016. 1, 2, 4

[6] Leon A Gatys, U Tübingen, Matthias Bethge, Aaron Hertzmann, Adobe Research, and Eli Shechtman. Preserving color in neural artistic style transfer. 2016. 2

[7] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision (ICCV)*, 2011. 3

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:770–778, 12 2015. 2

[9] Qin Huang, Chunyang Xia, Chihao Wu, Siyang Li, Ye Wang, Yuhang Song, and C. C. Jay Kuo. Semantic segmentation with reverse attention. *British Machine Vision Conference 2017, BMVC 2017*, 7 2017. 2

[10] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *IEEE Transactions on Visualization and Computer Graphics*, 26:3365–3385, 11 2020. 1

[11] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. 2

[12] Michael Kass and Andrew Witkin. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988. 1

[13] Gihyun Kwon and Jong Chul Ye. Clipstyler: Image style transfer with a single text condition. 2

[14] Shaohua Li, Xinxing Xu, Liqiang Nie, and Tat-Seng Chua. Laplacian-steered neural style transfer. 2017. 2

[15] Xiaodan Liang, Xiaohui Shen, Jiashi Feng, Liang Lin, and Shuicheng Yan. Semantic object parsing with graph lstm. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9905 LNCS:125–143, 3 2016. 2

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. 12 2016. 2

[17] Xiao-Chang Liu, Ming-Ming Cheng, Yu-Kun Lai, and Paul L Rosin. Depth-aware neural style transfer. 2017. 2

[18] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:3523–3542, 1 2020. 1

[19] Richard Nock and Frank Nielsen. Statistical region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1452–1458, 11 2004. 1

[20] Nils Plath, Marc Toussaint, and Shinichi Nakajima. Multi-class image segmentation using conditional random fields and global classification. 1

[21] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:6517–6525, 12 2016. 2

[22] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:640–651, 11 2014. 1, 2, 4

[23] Francesco Visin, Adriana Romero, Kyunghyun Cho, Matteo Matteucci, Marco Ciccone, Kyle Kastner, Yoshua Bengio, and Aaron Courville. Reseg: A recurrent neural network-based model for semantic segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 426–433, 11 2015. 2

[24] Jianbo Wang, Huan Yang, Jianlong Fu, Toshihiko Yamasaki, and Baining Guo. Fine-grained image style transfer with visual transformers. 2

[25] Weihao Weng and Xin Zhu. U-net: Convolutional networks for biomedical image segmentation. *IEEE Access*, 9:16591–16603, 5 2015. 2

[26] Yuhui Yuan, Xiaokang Chen, Xilin Chen, and Jingdong Wang. Segmentation transformer: Object-contextual representations for semantic segmentation. *arXiv*, pages 1–23, 9 2019. 2

[27] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:6230–6239, 12 2016. 2