Sensor Adaptation for Improved Semantic Segmentation of Overhead Imagery

Marc Bosch, Gordon A. Christie, Christopher M. Gifford
The Johns Hopkins University - Applied Physics Laboratory
Marc.Bosch.Ruiz@jhuapl.edu

Abstract

Semantic segmentation is a powerful method to facilitate visual scene understanding. Each pixel is assigned a label according to a pre-defined list of object classes and semantic entities. This becomes very useful as a means to summarize large scale overhead imagery. In this paper we present our work on semantic segmentation with applications to overhead imagery. We propose an algorithm that builds and extends upon the DeepLab framework to be able to refine and resolve small objects (relative to the image size) such as vehicles. We have also investigated sensor adaptation as a means to augment available training data to be able to reduce some of the shortcomings of neural networks when deployed in new environments and to new sensors. We report results on several datasets and compare performance with other state-of-the-art architectures.

1. Introduction

In recent years, semantic segmentation has received a lot of attention by the computer vision community. It aims to assign a class label to each pixel in the image by performing fine-grained inference. Semantic segmentation has become a stepping stone towards full scene understanding. It is a key enabler for certain applications, including self-driving vehicles, human-computer interaction, 3D semantic reconstruction, image/video editing, and surveillance tasks.

In the last few years, there have been many proposed architectures based on convolutional neural networks (CNNs). These include SegNet [3], RefineNet [14], PSPNet [16], DeepLab and its derivatives [5][6][7], among others. In particular, DeepLab has shown very promising results. However, we have observed that this architecture often fails to provide accurate/clear object boundaries. This has also been reported by others in [8]. One of the latest flavors of DeepLab includes atrous convolutions in the backbone feature extractor part of the network. This, and other filtering operations like pooling, often causes the network to miss certain information about natural edges. In overhead imagery, this is critical since many objects are closely-spaced.

Thus there is a need to obtain more refined segmentation maps.

In this work we propose to expand the DeepLab framework by extending it with a refinement module to produce segmentations with less over/under-segmentation outputs. In particular, we take the original DeepLabv3 backbone, which involves a ResNet101 architecture with atrous convolutions, and feed it to a multi-resolution module to progressively recover information as resolution is restored. Figure 1 shows an example segmentation product of the proposed model for an overhead image.

In addition to architecture changes, the other primary factor that influences the performance of neural networks is training data. As in other domains, performance on overhead imagery is sensitive to the training data used. In particular, how well it represents the test data. There are factors such as sensor characteristics, viewing geometry, resolution, and image modality that influence the performance across data sources due to data inconsistencies between the training and testing sets. Further, data annotation for overhead imagery is very costly and time consuming due to the large amount of objects in scenes and their large size. Therefore, it is critical to be able to re-use previous labeling efforts in as many contexts as possible. To this end, we set to exam-
ine how to “translate” our training data from a given source sensor to look more like the target sensor. The goal is to transfer visual characteristics from source sensor imagery to target sensor imagery. Generative Adversarial Networks (GANs) have been shown to accomplish this goal. Frameworks like pix2pix \[12\] and derivative networks \[13, 9, 2\] can successfully translate images from one domain to another. In this work, we have integrated a “translator” module into the image segmentation task to re-purpose training data to other sensors and platforms. Figure 2 shows the proposed conceptual framework. We have experimented with several simulated sensor types to quantify the gains of such a module. We report our results on several overhead test datasets.

In this paper we describe our proposed algorithm and the integration of sensor adaptation for the task of semantic segmentation for overhead imagery. Our main contributions can be summarized as follows:

- Proposed a semantic segmentation architecture using DeepLabv3 as a backbone and a multi-resolution processing pipeline.
- Proposed a two-network system to perform sensor adaptation for improved segmentation results.
- Evaluated several simulated sensor models to measure the added value of a sensor adaptation block.

The remainder of the paper is organized as follows: In Section 2 we review our proposed semantic segmentation algorithm. In section 3 we describe the sensor adaptation module. In section 4 we show our experimental results on overhead imagery. Finally, we conclude the paper by offering some remarks from our experimental observations.

2. Semantic Segmentation Model

In recent years, many semantic segmentation models have been published. Many follow a similar structure with a backbone deep neural network to extract features at one or more resolution scales and a collection of high-level layers to combine such features as powerful tools for prediction tasks. One of the most successful models, known as DeepLab, was introduced in \[7\] and expanded upon \[5, 6\].

2.1. DeepLab

The DeepLab framework leverages dilated convolutions and spatial pyramid pooling layers in the context of semantic segmentation. The baseline model consists of a modified ResNet architecture, where some convolution operations are replaced by upsampled filters or atrous convolutions allowing control of the resolution/scale at which features are extracted. In addition, having a tunable parameter or rate to control the sample spacing of the kernel allows the receptive field to change without adding more parameters into the model. Another key element of DeepLab is the addition of the atrous spatial pyramid pooling (ASPP) module that targets multi-scale processing by passing the feature maps to four parallel atrous convolutional layers with different rates. Having these layers allows the model to encode multi-scale context. Through experimentation, one of the shortcomings of this architecture is that results are not refined, or in other words, do not always coincide with natural edges in the scene, creating situations of over/under-segmentation.

2.2. Refined DeepLab

In overhead imagery, due to the small size of some of the objects of interest, accurate boundaries and object separation is critical to perform analytics and perceptual tasks. In this work we adapted the DeepLabV3 architecture and added a refinement block with the goal of obtaining segmentation masks that are more aligned with the natural edges of the scene. Our system consists of replacing the ASPP layers and adding a sequence of multi-scale refinement blocks that work on progressively higher resolution feature maps. Each multi-scale refinement modules consist of residual blocks, upscaler (2x factor), and a chained residual pooling similar to \[14\]. As mentioned earlier, this configuration is repeated until the full native image resolution is achieved. Figure 3 shows the block diagram of our modified DeepLabv3 architecture. The residual blocks consist of a sequence of three ReLU-Conv-BN basic units, with the first and last Conv layers using \((1 \times 1)\) kernels, and the convolutional block in the middle being a \((3 \times 3)\) kernel. The resulting residual, \(R(x)\), is added to the input, \(x\), to obtain the output of the residual block. This operation is repeated one more time, resulting in the final output \(y_1 = R_2(x + R_1(x)) + x + R_1(x)\).
After the residual blocks, the output $y_1$ is passed into the upscaler, which consists of two ReLU-Conv-BN units and an upsample operation in between that reverses the effect of a max-pooling by creating a $n \times n$ cell for each value. $n$ is the scale factor desired. The output of the upscaler is denoted as $y_2$ in Figure 3. We investigated two flavors of the upsample: padding zeros and direct copy. Zero-pad fills the $n \times n$ cell with zeros, while direct copy assigns the input value to all positions in the cell. Zero-pad: given a cell in a feature map with value equal to $[0.3; 0.3; 0.3]$, the result of the upsample operation is $[0.3; 0.3; 0.3]$, given $n = 2$. The direct copy results in $[0.3; 0.3; 0.3; 0.3; 0.3]$. In this work we set $n = 2$ for all of our experiments. After the scaler, the final block in each refinement module is a modified chained residual pooling. It consists of the max-pooling+Conv($3 \times 3$)+BN sequence, this structure is repeated twice followed by one last residual block (see Figure 3 for more details). The resulting output, $y_3$, is fed into the next multi-res refiner. Our network consists of a total of 4 refinement levels.

3. Source Adaptation for Overhead Imagery

Neural network performance applied to overhead imagery often produces underwhelming results compared to ground-based imagery. Even though viewpoints remain relatively consistent in overhead imagery, many images are collected off-nadir (oblique look angle). Other factors that degrade algorithm efficiency are ground sampling distance (i.e., GSD, or image resolution), time of day/year, viewing geometry, and geographical location, among others. Through experimentation, however, we have observed that the most critical contributing factor for loss in performance is due to training on imagery from one sensor and applying the trained model to images of a different sensor. One of our goals is to design semantic segmentation algorithms that can be deployed in new environments, regardless of the acquisition sensor, while incurring minimal accuracy degradation.

We tackled this problem by adding a secondary network that acts as a “translator” between the source sensor (sensor of which we have training data/labels available) and our target sensor by which test datasets are acquired. Intuitively, by training a segmenter with data that looks more similar to data to be used at test time we should reduce degraded network performance in new environments. Our goal is then to close the gap between the visual appearance of different sets of sensors for improved overall results. The proposed system with sensor adaptation (SA) added to the semantic segmentation network is shown in Figure 2. To the best of our knowledge, this is the first work that addresses this problem for overhead imagery.

Inspired by recent success of domain adaptation frameworks that focus on translating images from one domain to another, as in [12, 15], we used a cycle-consistent adversarial framework known as CycleGAN [17] to work with the overhead imagery. We explored several loss function configurations to adapt it to our problem.

We investigated adding a feature loss to preserve the content. The goal is to produce an image from the source sensor as if the target sensor had acquired it by solving the following problem:

$$I_o = \text{amin}_f (\alpha \text{Loss}_{adv} + \lambda (\beta \text{Loss}_{cycle} + (1 - \beta) \text{Loss}_{fm}))$$

where $\text{Loss}_{adv}$ is the “vanilla” GAN adversarial loss, defined as:

$$\text{Loss}_{adv} = E_{z \sim p_{model}} (\log (1 - D_{\theta_D} (G_{\theta_G}(z))))$$

where G, the generator, generates images $G(z)$ that look similar to images from the target sensor, and the discriminator, D, performs a binary (fake vs. real) classification on images from the target sensor and those generated by G.

The next term in the objective function is the $\text{Loss}_{cycle}$. It enforces cycle consistency between source and target images so that the learned image translation mapping should be able to take a source sensor image, translate it into a target sensor image, and bring it back to the source sensor’s content and style using an auxiliary generator $F$. In other words, it satisfies backward consistency by adding the target-to-source content loss:

$$\text{Loss}_{cycle} = E_{z \sim p_{model}} (|F(G(z)) - z|) + E_{x \sim p_{model}} (|G(F(x)) - x|)$$

Following [10, 14, 13], the $\text{Loss}_{fm}$ attempts to describe the visual attribute loss. Using a pre-trained VGG19 on ImageNet, feature vectors of generated and target images are extracted and compared according to:

$$\text{Loss}_{fm} = |T_{\theta_{vgg19}}(target) - T_{\theta_{vgg19}}(I_o)|$$
In early experiments we observed that both cycle loss and adversarial loss had the biggest contribution in generating high fidelity data, given the added overhead of the VGG19 network. In addition, we experimented with the style loss proposed in [10], but we did not observe any qualitative or quantitative improvement.

3.1. Implementation and Training Details

The segmenter of the proposed system uses DeepLabv3’s backbone, which consists of ResNet101 with atrous convolutions and batch normalization (BN) added to the network [7]. We replaced the ASPP module with our refinement module. The refinement module consists of four modules composed of residual blocks+upscaler (x2)+chained residual pooling block. The output of the final block is passed to the 1 × 1 final convolution that generates the final logits. All convolutional layers consist of 256 filters. Depending on the layer, kernel size is either set to 1 × 1 or 3 × 3. See Figure 3 for more details. The max-pooling unit in the chained residual pooling is a 5 × 5 operation, following [14]. In all our experiments, the network is trained for 10000 steps using a batch size of 4, and the PASCAL dataset is used to pre-train the backbone ResNet101.

The translator’s generator consists of 64 filters in its convolutional layers, as does the discriminator network [17]. We set the parameter controlling the contribution of the adversarial loss \( \alpha \) to 1, while \( \lambda \) is fixed to 10 to guide the network toward perceptual convergence. The translator is trained for 100 epochs in all of our experiments.

4. Evaluation

In this section we describe our experiments and present results for semantic segmentation of overhead imagery. We compare several methods using objective quality metrics such as mean intersection over union (mIoU), and F1 score, as well as show visual results of the proposed method. The mIoU measures ratio between the intersecting area of the predicted segmentation mask and the ground truth mask, and the area determined by the union of both masks. F1 score is defined as the harmonic mean of both precision and recall. We also conducted studies on the contribution of sensor adaptation for segmentation tasks. In this case the same train/test split was used to both train and evaluate the segmenter and the source adaptation GAN.

4.1. Overhead Imagery Datasets

We have conducted experiments on several public overhead imagery datasets. These include the ISPRS 2D Labeling dataset [11], USSTOCOM Urban 3D Challenge dataset [111], as well as other imagery from other airborne RGB sensors.

The ISPRS dataset was collected as part of the 2D Semantic Labeling Contest conducted by the International Society for Photogrammetry and Remote Sensing (ISPRS) [11]. In this work we have used the Potsdam collection, which includes aerial images from the historical city with large building blocks, narrow streets, and dense settlement structures. The Urban 3D Challenge dataset was released in 2017 as part of the large scale building segmentation challenge [111]. In this dataset satellite imagery is available with the corresponding binary masks for building annotations. In our experiments we focused on segmenting the following categories: buildings, vehicles, roads, and (low/high) vegetation. In the case of 3D Urban Challenge dataset, we focused on the buildings category. Although depth information was available through digital surface models (DSMs) and digital terrain models (DTMs), we have only used the RGB color information in this work.

4.2. Refined DeepLab Model Results

We have conducted our experiments on several overhead images, as described in the previous section. We have focused on the improvements of the refinement module over the baseline DeepLabv3 algorithm. The first and second rows of Table 3 show the performance gain in terms of mIoU and F1-score for all classes. The first and second rows of Table 4 show metric comparison between baseline and our method for individual classes (vehicles, buildings, and roads). In addition, we have included qualitative results of the proposed model. Figure 4 shows several examples for both building-only segmentation using the Urban 3D Challenge dataset, and multi-class segmentation using the 2D semantic labeling ISPRS dataset. Figure 5 shows visual comparisons between baseline DeepLabv3 and our work.

4.3. Sensor Adaptation Results

In this section we describe the sensor adaptation (SA) experiments. In this case we have focused the comparisons on building segmentation. We compared performance of the proposed segmenter with and without sensor adaptation. We have simulated three sensor characteristics to cover different sensor profiles with decreased visual correlation with respect to the original sensor. Our first simulated sensor consisted of a grayscale-only version of the data. We refer to it as “grayscale sensor”. For some sensors, only panchromatic EO (grayscale) imagery is available, thus we wanted to have a model that simulates lack of RGB color data. The second simulated sensor consists of swapping the position of the color channels from RGB to BRG in order to get images that looked different from the original. In addition, we dropped every other pixel for each color channel to simulate resolution loss between the two sensors (original and simulated). We refer to this simulated sensor as “BRG-type1”. Finally, our third simulated sensor, “BRG-type2”, consisted of swapping each color chan-
nel from *RGB* to *BRG*, as we did for “BRG-type1” and then applying different downsample factors for each channel to simulate more extreme resolution differences. We only kept one out of every eight pixels for the blue channel, every other pixel for the red channel, and only kept one out of every four pixels in the green channel. We then use nearest neighbor interpolation to align the three channels. Figure 6 shows a sample for the source sensor, the corresponding version for each of the three simulated target sensors for the same scene, and the building segmentation results for the segmenter with and without the translator module that performs the actual sensor adaptation.

In Table 1 we have included performance results of the segmenter (mIoU and F1-score) for both configurations - with and without SA. The test set contains more than 13000 buildings. In addition, in Table 2 we captured the performance gain of adding SA for building counts to quantify a more tangible benefit of SA. By running a simple analytics algorithm that counts the number of buildings in the scene, we compared both outputs to the ground truth (actual number of buildings).

### 4.4. Refined DeepLab and Source Adaptation Results

In this section we report our results after combining our proposed architecture and sensor adaptation. Table 3 shows mIoU and F1-score results for our proposed method with and without SA, as well as the original DeepLabv3/ResNet101 baseline framework. In addition, we added the recently proposed DeepLabv3+ with MobileNet as its backbone [8]. Table 4 presents the per-class (buildings, vehicles, roads) results. Finally, Figure 7 shows visual results on overhead imagery samples for all algorithms.
Table 2. Performance comparison between segmenter with and without SA for building counting task applied on three simulated sensors (Grayscale, BRG-Type1, BRG-Type2).

| Sensor         | Building Counts | Difference from Truth |
|----------------|-----------------|-----------------------|
| Ground truth   | 13292           | 0                     |
| Grayscale      | 14417           | +1125                 |
| Grayscale + SA | 12975           | -317                  |
| BRG-type1      | 16000           | +2708                 |
| BRG-type1 + SA | 14709           | +1417                 |
| BRG-type2      | 30819           | +17527                |
| BRG-type2 + SA | 16001           | +2709                 |

Table 3. Segmentation performance comparison (mIoU, F1-score).

| Algorithm        | mIoU  | F1-score |
|------------------|-------|----------|
| DeepLabv3 [7]    | 0.42  | 0.37     |
| Our Method       | 0.43  | 0.45     |
| Our Method + SA  | 0.51  | 0.47     |
| DeepLabv3+ [8]   | 0.44  | 0.39     |

5. Discussion

There are certain inherent challenges when processing images acquired with an airborne/spaceborne platform, including number of pixels that compose an object, separation space between objects (e.g., vehicles in parking lots), different viewing geometries, and geospatially diverse backgrounds. For these reasons, semantic segmentation from overhead imagery requires robust algorithms that can produce refined outputs. Our goal is to minimize over/under-segmentation as much as possible.

Another challenge is the lack of dense annotated data available from overhead platforms. Therefore, there is a need to recycle the current existing datasets. Being able to adapt existing datasets to different sensor platforms with different noise patterns, altitudes, and resolutions is key to being able to transfer current techniques applied to ground-based imagery to overhead imagery applications. Thus, we have designed a system that can provide accurate object segmentation while being able to re-use data from different sensors. From the results we have shown, our proposed method improves DeepLabv3 baseline by producing more refined and accurate results. In Figure 5, we have shown several examples of the added value of our refinement module. We show that there are fewer cases of over/under-segmentation compared to the baseline. Also, in terms of performance metrics (see Table 3), our approach can obtain more accurate segmentations than the baseline.

Adding sensor adaptation generated data allows for a larger improvement. Table 3 shows how we can boost the overall performance by significant differences. For instance, mIoU value jumps from 0.43 to 0.51.

When including other state-of-the-art algorithms in our comparisons, we observe that our proposed method remains competitive (with and without SA). Looking at the breakdown per class, we can see that it outperforms the other algorithms. See Table 4 for more details.

Finally, Figure 7 shows qualitative results of all compared algorithms, and we observe that our proposed methods is more visually similar to the ground truth.

In terms of source adaptation experimentation, we aimed to study its limitations by applying different (simulated) sen-
Table 4. Segmentation performance comparison (mIoU, F1-score) for several classes.

| Algorithm          | mIoU vehicles | mIoU buildings | mIoU roads | F1-score vehicles | F1-score buildings | F1-score roads |
|--------------------|---------------|----------------|-----------|-------------------|--------------------|----------------|
| DeepLabv3          | 0.32          | 0.34           | 0.43      | 0.36              | 0.31               | 0.41           |
| Our Method         | 0.31          | 0.41           | **0.56**  | 0.42              | 0.49               | 0.44           |
| Our Method + SA    | **0.51**      | **0.56**       | 0.45      | **0.50**          | **0.51**           | 0.41           |
| DeepLabv3+ (MobileNet) | 0.45      | 0.34           | 0.53      | 0.39              | 0.34               | 0.42           |

Figure 7. Comparison of results for several state-of-the-art models. Left column shows the original image chips, second column DeepLabv3/ResNet101 baseline results, third column our proposed improvements without source adaptation, fourth column shows outputs of our full model including source adaptation during training, fifth column shows DeepLabV3+ with MobileNet backbone, and right-most column shows the ground truth annotations for reference.

For all three sensor types, the segmentation algorithm using SA outperforms the non-SA version. In Tables 3 and 4, we see that the trend remains when applied to other (real) sensors in all but one class (roads). SA results show improvement with respect to the same system without the “trans-
lator” module. In addition, we performed a small exercise to translate the improvements to a more tangible outcome: building counting. For some applications, including urban planning and monitoring, population growth, or illegal land usage, building counts can offer an estimate of such activities. In order to count buildings, we counted connected components in the building mask from the segmentation and compared them to the manually annotated ground truth. For all three sensor types, our system with SA reported a much more accurate building count compared to the same algorithm without SA. In the worst case scenario, we were off by roughly 20% (sensor BRG-type2), while its counterpart without SA was off my more than 200% for the same sensor. This exercise was performed to demonstrate the added value of accurate segmentations even when using images from a very different target sensor. See Table 2 for more details.

During our experimentation with GANs for source adaptation, we observed that the cycle loss was critical and that a low value for $\beta$ was failing to produce good “translated” images. This was somewhat expected, as the feature matching loss compares features extracted with an ImageNet pretrained VGG19 network. A standard universal feature extractor does not efficiently capture visual attributes since there is little to no correlation between ground-based image features and those found in overhead imagery data. As we enter an era of more ubiquitous imagery being available from overhead sensors, methods that can digest and process all visual data accurately will be necessary. Being able to leverage all the available datasets for overhead applications will enable much improved results. Performing feature transfer from objects observed from the ground to the overhead domain can help speed up the transition of current state-of-the-art systems to overhead applications. Exploring ground-to-space sensor/domain adaptation represents future work.

6. Conclusion

Given recent progress in semantic segmentation using deep learning, overhead imagery is an area that can leverage these advances and use it for improved image exploitation capabilities. In this work, we have expanded upon the DeepLabv3 architecture by adding a refinement stage that better adapts to overhead context. Our model aims to learn proper semantic partitions by performing progressive multi-resolution processing in the feature maps produced by the DeepLabv3 backbone. In addition, we have added a “translator” model based on cycle consistency adversarial domain translation to transfer the visual characteristics of images from a source sensor to images acquired by a target sensor where the system needs to be deployed. Through a series of experiments we showed how the proposed refinement blocks help the segmenter obtain refined and more accurate semantic maps. We have also shown how sensor adaptation can further improve segmentation performance by training on data with similar visual characteristics to the target sensor. Finally, we have shown how the combination of the refinement steps and sensor adaptation leads to improved results for the task of semantic segmentation of overhead imagery.

References

[1] ISPRS 2d semantic labeling dataset. http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html
[2] A. Anoosheh, E. Agustsson, R. Timofte, and L. V. Gool. Combogan: Unrestrained scalability for image domain translation. In CVPR, 2018.
[3] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.
[4] M. Bosch, C. M. Gifford, and P. A. Rodriguez. Super-resolution for overhead imagery using densecrf and adversarial learning. In abs/1711.10312, 2018.
[5] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. CoRR, abs/1412.7062, 2014.
[6] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans. Pattern Anal. Mach. Intell., 40(4):834–848, 2018.
[7] L. Chen, G. Papandreou, F. Schroff, and H. Adam. Re-thinking atrous convolution for semantic image segmentation. CoRR, abs/1706.05587, 2017.
[8] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In ECCV, 2018.
[9] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In CVPR, 2018.
[10] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. arXiv, 2015.
[11] H. Goldberg, M. Brown, and S. Wang. A benchmark for building footprint classification using orthorectified rgb imagery and digital surface models from commercial satellites. In AIPR, 2017.
[12] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In CVPR, 2017.
[13] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In ECCV, 2016.
[14] G. Lin, A. Milan, C. Shen, and I. Reid. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In CVPR, 2017.
[15] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic
manipulation with conditional gans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[16] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. http://arxiv.org/abs/1612.01105, 2016.

[17] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Computer Vision (ICCV), 2017 IEEE International Conference on, 2017.