Proceedings of the Joint Austrian Computer Vision and Robotics Workshop 2020 DOI: 10.3217/978-3-85125-752-6-27

Highly Accurate Binary Image Segmentation for Cars

Thomas Heitzinger, Martin Kampel
Computer Vision Lab, TU Wien, Austria
{thomas.heitzinger,martin.kampel}@tuwien.ac.at

Abstract. *We study methods for the generation of highly accurate binary segmentation masks with application to images of cars. The goal is the automated separation of cars from their background. A fully convolutional network (FCN) based on the U-Net architecture is trained on a private dataset consisting of over* 7000 *samples. The main contributions of the paper include a series of modification to common loss functions as well as the introduction of a novel Gradient Loss that outperforms standard approaches. In a specialized postprocessing step the generated masks are further refined to better match the inherent curvature bias typically found in the outline of cars. In direct comparison to previous implementations our method reduces the segmentation error measured by the Jaccard index by over* 65%.

1. Introduction

A majority of buyers and sellers of cars choose to use online platforms. The quality of pictures on such platforms has a considerable impact on a buyers likelihood to purchase and thus leads to a demand for visually appealing images. For most sellers it is financially infeasible to take professional photographs and it has instead become common practice to digitally edit them. A binary segmentation mask is created that segments the image into foreground (the vehicle) and background. This mask is used to either alter (e.g. blur) or entirely replace the background with an artificial scene. Due to the significant demand for high quality segmentation masks dedicated businesses offering this service have emerged. As each photograph is edited by hand, the total time until the segmentation mask is available to the dealership lies between one and two days. The delay in time generates non-negligible costs. Based on novel deep learning techniques that have advanced the state-of-the-art in recent years we study methods for the fully automated generation of segmentation masks with focus on the maximization of accuracy. This paper aims to improve the state of the CarCutter[1] service.

2. Related Work

The first application of convolutional networks to semantic segmentation with per-pixel prediction was made possible by the introduction of fully convolutional networks (FCN) [13]. Previously segmentation solutions repurposed convolutional network architectures [12, 4] intended either for classification or object detection and always included fully connected layers. These adaptations come with drawbacks on either speed or accuracy. By reinterpreting fully connected layers in classification networks as convolutional layers that cover the entire input region the network architecture is made independent of the dimensions of the input image. Instead of a class probability vector the reinterpreted network outputs a coarse heatmap for each class. In order to obtain predictions at the pixel level the coarse semantic information of deeper levels is repeatedly upsampled and added to the activations of shallower feature maps. This innovation was quickly expanded on and led to development of the U-Net architecture [11]. It introduces a symmetric encoder-decoder format consisting of a contracting encoder component and an expanding decoder component. This setup is chosen with the intention of learning a comparatively low dimensional image representation in the narrow region of the network (referred to as the *bottleneck*) that captures global context while at the same time dramatically reducing the number of learned parameters. Skip connections efficiently pass shallow encoder features with high localization accuracy to deep decoder layers that are rich in semantic information. Variations on networks of this

[1] https://www.car-cutter.com/ (accessed February 24, 2020)

type are often focused on the decoder component, while the standard approach for the encoder component is the repurposing of the convolutional stage of known, well performing networks, such as VGG-16 [14]. The variations in the decoder component essentially explore the trade-off between low memory requirements (and fast inference) and high accuracy. Architectures such as [16] also investigate the benefits of an additional ResNet [5] based refinement stage. Benchmarks show [2, 10] that almost all state-of-the-art solutions for a variety of image segmentation tasks are based on the U-Net architecture. It is also chosen by well performing entries [6, 15] to the Kaggle Carvana Image Masking challenge. TernausNet [6] was part of the winning entry in the challenge and uses a pretrained encoder based on VGG-11 [14] while [15] placed in the top 4% using an ensemble of five network with a pretrained ResNet-50 [5] encoder.

3. Dataset

Training was done on a private dataset consisting of 7614 pairs of RGB-images and binary segmentation masks. Some images contain additional cars in the background that are smaller by area. In these cases the solution is expected to only segment the main vehicle. The dataset exhibits a bias towards German car brands such as Volkswagen, BMW and Mercedes and contains a disproportionate amount of images with cars higher-than-average in cost. During preprocessing all images are resized to a resolution of $800\text{px} \times 600\text{px}$. Data augmentation is used to boost the available training data.

To our knowledge, the most closely related dataset is tied to the Kaggle Carvana Image Masking challenge[2]. The goal of this challenge is identical to ours. Its dataset contains roughly $100\,000$ image/mask pairs with resolution $1920\text{px} \times 1080\text{px}$. Compared to our dataset the samples are more uniform. Each picture contains exactly one vehicle which is placed in a fixed position and all photographs are taken by the same stationary cameras under identical lighting conditions. The winning entry of this challenge achieved a Jaccard index of 0.9947 which we consider to be an upper bound to the score achievable on our dataset.

[2]https://www.kaggle.com/c/carvana-image-masking-challenge (accessed February 21, 2020)

4. Methods

Segmentation is performed with a fully convolutional neural network of the U-Net architecture. Its implementation is similar to [16], with a pretrained convolutional stage of a VGG-16 network with batch normalization for the encoder and an additional ResNet-style refinement block after the decoder. Segmentation quality is evaluated using the Jaccard index which is the de facto standard metric for image segmentation methods:

$$\mathcal{M}_{\mathrm{J}}(P, T) \coloneqq \frac{|P \cap T|}{|P \cup T|}. \tag{1}$$

In our context T and P are subsets of target (ground truth) and predicted pixels in a segmentation mask. Images x, target masks t and predicted masks p are assumed to be non-binary with height N, width M and values in the range $[0, 1]$.

We study training with (modifications of) the loss functions Mean Squared Error $\mathcal{L}_{\mathrm{MSE}}$ and Binary Cross-Entropy $\mathcal{L}_{\mathrm{BCE}}$ as well as the Dice Loss [9] $\mathcal{L}_{\mathrm{DSC}}$ which is defined as

$$\mathcal{L}_{\mathrm{DSC}}(p, t) \coloneqq 1 - \frac{\epsilon + 2\sum_{(i,j)\in D} p_{ij} t_{ij}}{\epsilon + \sum_{(i,j)\in D} p_{ij} + t_{ij}}, \tag{2}$$

and is related to the Jaccard index. Here $D = \{1 \,..\, N\} \times \{1 \,..\, M\}$ is the domain of the segmentation masks and $\epsilon \ll NM$ is a small scalar regularization term.

4.1. Weighting Schemes

We propose modifications that improve upon the standard losses Mean Squared Error and Binary Cross-Entropy. The main idea is that not all areas of an image are equally important or equally difficult to segment. Loss functions that are the sum or mean of pixelwise losses can be modified to assign weights to each pixel in order to adjust for this inhomogeneity. We can use a map w of real weights with shape equal to t and p and define a modified version of Mean Squared Error as:

$$\mathcal{L}_{\mathrm{MSE}}(p, t) \coloneqq \frac{1}{NM} \sum_{(i,j)\in D} w_{ij} \left(p_{ij} - t_{ij}\right)^2. \tag{3}$$

An analogous modification can be made to Binary Cross-Entropy.

Notation The notation $\nabla_\sigma y$ expresses the convolution of a stack of feature maps y with the gradient of a two-dimensional Gaussian density with mean vector $(0,0)^{\mathrm{T}}$ and covariance matrix σI, where I is the identity matrix. In practice it is a convolution $\nabla_\sigma y = y * G_\sigma^\nabla$ with a kernel G_σ^∇ that is normalized and has shape $2 \times C \times C$ with $C \sim 4\sigma$. The operation doubles the channels of the tensor y. To ensure fast computation the convolution is implemented as a convolutional layer with frozen weights.

Median Frequency Balancing A simple and popular [3, 1] weighting scheme is *Median Frequency Balancing* (MFB). Each class (foreground and background in our binary setting) is assigned a weight to compensate for imbalance in the frequency of occurrence. The weights can either be computed individually for each sample or once for the entire dataset. In the individual case a foreground/background weight pair (w_f, w_b) for a target mask t is given by

$$w_f = \frac{N}{2\sum_{(i,j)\in D} t_{ij}} \quad \text{and} \quad w_b = \frac{N}{2\sum_{(i,j)\in D}(1-t_{ij})}. \tag{4}$$

An example of such a weight map is shown in the second column of Figure 1. If a single weight pair for the entire dataset is preferred then it is computed as the mean of all sample weights.

Boundary Proximity The separating boundary between foreground and background is the only area where the segmentation mask is non-constant. Consequently it is also the area where masks generated by neural networks exhibit the largest mistakes. In this approach pixels in close vicinity to the boundary are assigned larger weights. Such a method is already suggested by the authors of the original U-Net architecture [11], although with a less general approach. We calculate a weight map based on a pixel's distance to the separation boundary using convolution based edge detection. A large gradient in a segmentation mask indicates the presence of an edge. Based on this we define the weight map

$$w_{ij} = 1 + c\|(\nabla_\sigma t)_{ij}\|_2^2 .$$

A map of this type is shown in the third column of Figure 1. The parameter c is a scaling constant and is set to 5.

Gradient Ratio The typical location of segmentation errors can be characterized more concretely. Photographs are often taken in poor lighting conditions or with cheap camera equipment resulting in over- or underexposed areas. Common occurrences are bright reflections in a vehicles roof or dark shadows around its wheelbase (see Figure 2). Both scenarios can obscure the precise transition point between foreground and background. At the data level we are confronted with image patches that are either nearly entirely white or nearly entirely black, while the same patch in the ground truth segmentation mask contains a binary transition. Motivated by this observation we claim that the ratio between change in the mask and change in the corresponding image is a measure for prediction difficulty and use it to define a new weight map. Again we employ discrete gradients:

$$w_{ij} = 1 + c\,\frac{\|(\nabla_\sigma t)_{ij}\|_2^2}{\|(\nabla_\sigma x)_{ij}\|_2^2 + \epsilon}.$$

As previous the parameter c is a constant which is set to 0.1 and ϵ is a small regularizing constant with $\epsilon \ll NM$. The result of the convolution $\nabla_\sigma x$ is a stack of six feature maps, one for each combination of the three image channels and the two partial derivatives in the gradient. A weight map of this type is shown in the fourth column of Figure 1.

Comparative Results In Table 1 we show results for the pixelwise losses Mean Squared Error and Binary Cross-Entropy, first in their default state and then with the addition of one or more weighting extensions. To us a pixelwise loss is a function that sums over the losses of individual pixels. Consequently when the gradient is computed during backpropagation all terms except the ones belonging to the individual pixels vanish. We can argue that in such a loss function no pixel is *ignored* or treated lesser.

In direct comparison Binary Cross-Entropy outperforms Mean Squared Error in every test. From an information theoretic point of view it is the natural loss for binary classification problems. When using Mean Squared Error none of the proposed weighting schemes improved over uniform weights whereas the opposite holds true for Binary Cross-Entropy where the best results are achieved using a combination of Median Frequency Balancing and Gradient Ratio.

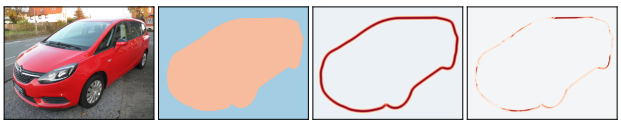

Figure 1. Comparison of weighting methods. A sample image *(left)* and weight maps for Median Frequency Balancing *(left middle)*, Boundary Proximity *(right middle)* and Gradient Ratio *(right)*.

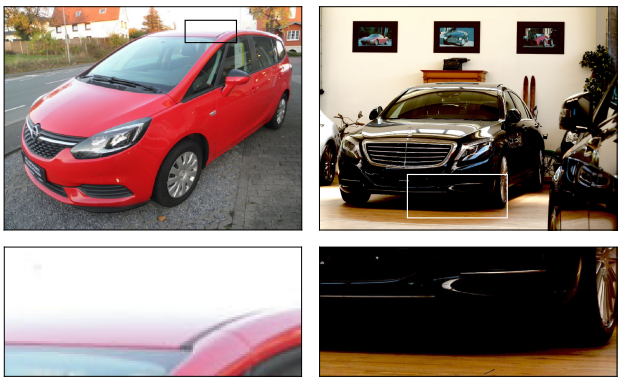

Figure 2. Examples of an overexposed bright region *(left)* and an underexposed dark region *(right)*.

	$\mathcal{L}_{\text{MSE}}$	$\mathcal{L}_{\text{BCE}}$
Default	$\mathbf{1.189 \times 10^{-2}}$	1.145×10^{-2}
MFB	1.284×10^{-2}	1.114×10^{-2}
Boundary Proximity	1.217×10^{-2}	1.126×10^{-2}
Gradient Ratio	1.296×10^{-2}	1.108×10^{-2}
MFB + Gradient Ratio	1.282×10^{-2}	$\mathbf{1.106 \times 10^{-2}}$

Table 1. Comparison of network performance after training with Mean Squared Error $\mathcal{L}_{\text{MSE}}$ and Binary Cross-Entropy $\mathcal{L}_{\text{BCE}}$ measured by $1-$Jaccard index.

4.2. Gradient Loss

We expand on the idea of using discrete gradients in loss functions and introduce the Gradient Loss. This loss is inspired by the H^1 Sobolev seminorm $|f|_{H^1} := \|\nabla f\|_{L^2}$. Instead of optimizing the plain value of the segmentation mask we optimize its gradient:

$$\mathcal{L}_\nabla(p, t) := \mathcal{L}_{\text{MSE}}(\nabla_\sigma p, \nabla_\sigma t) \tag{5}$$

In our tests we observe that neural networks trained with this loss produce masks with cleaner constant regions. We believe it allows them to learn that segmentation masks should be largely constant, i.e. for most areas $\nabla_\sigma p$ should be zero. It is not advisable to use the Gradient Loss on its own since it is based only on a seminorm. Depending on how the convolution treats missing values on the boundaries there might hold $\mathcal{L}_\nabla(p + c, t) = \mathcal{L}_\nabla(p, t)$ for constant values c (invariance to constant shifts). If the Gradient Loss is combined with Mean Squared Error we essentially obtain a discrete version of the H^1 Sobolev norm.

	$1-$Jaccard index
$\mathcal{L}_{\text{DSC}}$	9.237×10^{-3}
$\mathcal{L}_{\text{BCE}}$	1.145×10^{-2}
$\mathcal{L}_{\text{MSE}}$	1.189×10^{-2}
$\mathcal{L}_{\text{DSC}} + \mathcal{L}_{\text{BCE}}$	1.045×10^{-2}
$\mathcal{L}_{\text{DSC}} + \mathcal{L}_{\text{MSE}}$	9.633×10^{-3}
$\mathcal{L}_{\text{BCE}} + \mathcal{L}_\nabla$	1.082×10^{-2}
$\mathcal{L}_{\text{MSE}} + \mathcal{L}_\nabla$	1.224×10^{-2}
$\mathcal{L}_{\text{DSC}} + \mathcal{L}_{\text{BCE}} + \mathcal{L}_\nabla$	1.027×10^{-2}
$\mathcal{L}_{\text{DSC}} + \mathcal{L}_{\text{MSE}} + \mathcal{L}_\nabla$	$\mathbf{9.118 \times 10^{-3}}$
Previous solution	2.640×10^{-2}

Table 2. Network performance after training with composite loss functions measure by $1-$Jaccard index.

4.3. Composite Loss Results

It is common practice to combine the Dice Loss with a pixelwise loss [8] which results in segmentation maps with sharper boundaries. The combination of multiple loss functions is achieved by simple addition of the individual losses. Addition of the Dice Loss to the pixelwise losses uniformly results in a performance increase (see Table 2) due to its close relation to the Jaccard index. In these tests Mean Squared Error surpasses Binary Cross-Entropy by a significant margin while the incorporation of weighting schemes worsened results. The further addition of the Gradient Loss leads to mixed, but generally positive results. Although the effects on the combination of Dice Loss and Binary Cross-Entropy are minor, we achieve overall best results with the combination of the three losses Dice Loss, Mean Squared Error and Gradient Loss. The score outperforms the previous best result which was achieved with plain Dice Loss. Compared to a previous implementation used by the CarCutter service the segmentation error is reduced by 65%.

4.4. Postprocessing

The proposed weighting schemes and loss functions can only work if over- or underexposed regions are not completely devoid of texture. Otherwise a neural network may only learn to predict a pixel's probability to belong to the foreground class which inevitably causes non-sharp transition in the predicted masks. An alternate approach is the use of a custom postprocessing procedure.

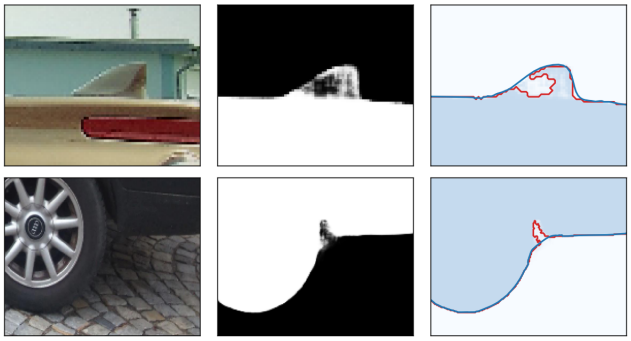

Figure 3. Two examples of active contour modeling. Crops of car images *(left)*, the inferred segmentation masks *(middle)* and contour segments *(right)* before active contour modeling *(red)* and after *(blue)*.

Inferred segmentation masks are thresholded and the resulting sharp separating contour between the foreground and background region is subjected to a refinement procedure. The contour is split into *certain* and *uncertain* regions depending on the neural networks certainty in its prediction. A contour region is considered to be certain if nearby values in the corresponding segmentation mask are close to 0 or 1, and uncertain otherwise.

Uncertain Regions These contour regions typically occur in over- or underexposed areas of an image and are iteratively adjusted using active contour modeling [7]. The method aims to minimize an energy functional of a spline contour in the inferred segmentation mask. Figure 3 shows the effects of the approach on two examples. The first example shows its positive influence while the second is a failure case.

Certain Regions Cars typically have large regions that are smooth for aerodynamic and aesthetic reasons. Edges that *are* present however can be rather sharp. The motivation of the following procedure, which we call *adaptive smoothing*, is to mimic this bias. We aim to perform a high degree of smoothing without displacing the contour by more than 0.5 pixels. The upper limit is enforce since based on the neural network assessment such a segment is already close to the ground truth target.

As an initial step the contour segment is split into separate sequences for x and y coordinates. The following procedure is applied separately to both. Let $\kappa = (\kappa_i)_{i=1}^N$ be such a sequence of real points. We use Gaussian filters G_{σ_i} with standard deviations σ_i that adept to the current position. A kernel G_{σ_i} is obtained by sampling a Gaussian density in the points $\mathbb{Z} \cap [-2\sigma_i, 2\sigma_i]$ and normalizing.

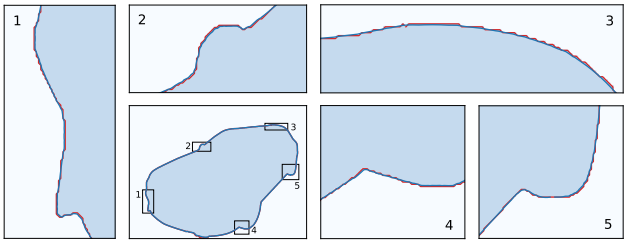

Figure 4. Comparison of a contour before postprocessing (red) and after adaptive smoothing (blue). Full segmentation mask and contours *(bottom second from the left)* and five enlarged regions.

The smoothed contour κ^s has equal shape to κ and is defined as

$$\kappa_i^s := \left(\kappa * G_{\sigma_i}\right)_i . \tag{6}$$

For the computation of the values σ_i we are looking for the largest kernel that displaces κ less than 0.5 pixels. A naive implementation of this idea has two issues: First the set $\left\{\sigma_i \in \mathbb{R}_{\geq 0} : \left|\kappa_i - \kappa_i^s\right| < 0.5\right\}$ might not be bounded and second this approach can lead to large jumps in consecutive entries of σ. For this reason we pose the definition with additional restrictions:

$$\begin{aligned}
&(B) \quad \sigma_1 = \sigma_N = 0, \\
&(C) \quad |\sigma_i - \sigma_{i+1}| \leq \alpha, \quad i \in \{1 \dots N-1\}, \\
&(M) \quad \sigma_i \in \mathbb{R}_{\geq 0} \text{ maximal s.t. } |\kappa_i - \kappa_i^s| < 0.5.
\end{aligned}$$

Under these conditions solutions exist and are unique. Requirement (B) enforces the fixed boundary conditions $\kappa_1 = \kappa_1^s$ and $\kappa_N = \kappa_N^s$ while (C) ensures continuity within the contour segment. The parameter α specifies an upper bound for the slope. In practice the setting $\alpha = 0.5$ performs well. For the implementation of this method it is advisable to only consider a discrete set of possible values for σ_i. A comparison of contours before and after postprocessing can be seen in Figure 4.

5. Conclusion

We studied methods for the generation of highly accurate binary segmentation masks, including weighting schemes that improved the performance of default loss functions and a novel Gradient Loss. In addition we developed a specialized postprocessing procedure that exploits a bias in our dataset. We created a solution that poses a significant upgrade over

a previous implementation of the CarCutter service. Direct comparison of masks generated by our implementation with ones generated by the service in April of 2019 showed a reduction of 65% in the segmentation error as measured by the Jaccard index. With the exception of our postprocessing procedure the presented methods are applicable to the general image segmentation task.

Acknowledgments

We would like to thank micardo GmbH[3] for a fruitful collaboration and the use of their private dataset. This work has been partly funded by the Austrian security research programme KIRAS of the Federal Ministry for Transport, Innovation and Technology (bmvit) under Grant 873495.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *2017 IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, June 2016.

[3] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 2650–2658, Washington, DC, USA, 2015. IEEE Computer Society.

[4] Feng Ning, D. Delhomme, Y. LeCun, F. Piano, L. Bottou, and P. E. Barbano. Toward automatic phenotyping of developing embryos from videos. *2005 IEEE Transactions on Image Processing (TIP)*, 14(9):1360–1371, Sep. 2005.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[6] V. Iglovikov and A. Shvets. Ternausnet: U-net with VGG11 encoder pre-trained on imagenet for image segmentation. *CoRR*, abs/1801.05746, 2018.

[7] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[8] M. Khened, V. Alex Kollerathu, and G. Krishnamurthi. Fully convolutional multi-scale residual densenets for cardiac segmentation and automated cardiac diagnosis using ensemble of classifiers. *Medical Image Analysis*, 51, 01 2018.

[9] F. Milletari, N. Navab, and S. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571, Oct 2016.

[10] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott. A perceptually motivated online benchmark for image matting. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1826–1833, June 2009.

[11] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *2015 Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]).

[12] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *2014 International Conference on Learning Representations (ICLR), CBLS, April 2014*, 2014.

[13] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *2017 IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(4):640–651, Apr. 2017.

[14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *2015 International Conference on Learning Representations (ICLR)*, 2015.

[15] J. Xu, H. Guo, A. Kageza, S. Wu, and S. AlQarni. Removing background with semantic segmentation based on ensemble learning. EAI, 9 2018.

[16] N. Xu, B. Price, S. Cohen, and T. Huang. Deep image matting. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 311–320, July 2017.

[3] https://www.micardo.com/ (accessed February 24, 2020)