Semantic segmentation for 3D reconstruction of ASINs for AR

Xi (Brian) Zhang* xizhn@a9.com
Tony Hwang* thwang@a9.com
Arnab Dhua adhua@a9.com
Himanshu Arora arorah@a9.com
Sundar Vedula svedula@a9.com

Abstract

As part of our 3D model generation process for products sold by Amazon, we estimate pixel-accurate segmentation masks on multi-viewpoint product images taken in a controlled environment. Classical computer vision (CV) techniques for image segmentation work well on objects with simple geometries and Lambertian-like reflectances, but require manual parameter tuning and lack robust performance on objects with complex material properties. In this work, we present a deep learned semantic segmentation network tailored to our use case. Our experimental results demonstrate an improvement in accuracy on segmenting previously challenging object categories including objects with reflective and translucent surfaces as well as fine edge detail. Additionally, we produce two ground truth segmentation data sets consisting of diverse examples of product images captured from a densely sampled hemisphere of viewpoints.

1 Introduction and background

Precise image segmentation is a challenging prerequisite step for visual hull-based 3D reconstruction. To generate a 3D model of a product sold at Amazon, we capture images in a hemisphere of viewpoints around the product, estimate the segmentation mask from each viewpoint, then use these masks along with corresponding calibrated camera poses to estimate the 3D structure of the product. We produce such models at scale to be used in Amazon's AR View [**?**] mobile experience which allows customers to preview the fit and aesthetic of a product in their own space. A higher quality segmentation mask will improve the 3D structure estimation and accordingly the accuracy of the previewed asset. Our segmentation task is constrained to two classes: foreground and background. We have an intractably large variety of types of products sold on Amazon that comprise our single foreground class. Also, we are segmenting images taken against a mostly featureless background with strong non-diffuse lighting that sometimes induces specular highlights. Prior to using the proposed network, classical CV techniques based on background subtraction and iterative morphological refinement [**?**] were used for segmentation in our model generation pipeline. Our neural network-based segmentation algorithm has replaced the classical algorithm in production, increasing the generalizability of our model generation process to a much greater diversity of product types.

1.1 Related work

Semantic segmentation is pixel-level partitioning of an image into distinct classes. In recent years, various convolutional neural network (CNN) architectures have been proposed to address this task with great success. Fully Convolutional Networks (FCN) [**?**] was the first work to modify the

*Authors with equal contribution

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

state-of-art in classification networks in 2015 by replacing fully connected layers with convolutional layers and adding skip connections to obtain class heatmaps for arbitrary input image sizes. To better retain spatial information, other CNN architectures emerged, such as U-Net [**?**] and SegNet [**?**] which employed the encoder-decoder architecture to incrementally increase the receptive field during encoding, and then gradually recover resolution in the decoder. Another mechanism for retaining global and local detail is spatial pyramid pooling, found in PSPNet [**?**] which concatenates the output of multiple dilated convolutions to aggregate sub-region information at multiple scales. Recently, DeeplabV3+ [**?**] utilized many of the novel components mentioned above while optimizing for compute efficiency with depthwise separable convolutions.

1.2 Novel contributions

Our semantic segmentation network is based on DeeplabV3+ as this was the state-of-the-art at the time of development. The modifications that we made to this architecture are motivated by the following factors. The input images are taken in a well-lit, mostly featureless environment, with background image data available to us. The application runs on large batches of images, with the greatest priority being segmentation accuracy, not speed. No real-time performance was required. As such, we have made the subsequent modifications to the DeeplabV3+ architecture. Our input volume is the concatenation of the input image and its corresponding background image. We use a heavier feature extractor of ResNet50 instead of Xception. We replace all depthwise separable convolution with standard convolution. We add additional scales to the ASPP module. Finally, we replace bilinear upsampling with fractionally-strided convolution ("deconvolution"). Please see Section 2 for additional detail on our reasoning.

In order to train the network for our use case, we have produced two data sets that contain product images, corresponding background images, and hand-annotated masks. The MARVIN Segmentation 10k data set consists of images taken from a hemisphere of viewpoints from 243 products with diverse physical properties, while the MARVIN Kitchen 5k data set consists of images of 78 products from the kitchen and home categories. Access to these data sets is available internally at Amazon upon request.

2 Proposed network

DeepLabv3+ [**?**], the foundation of our network, combines novel components from other semantic segmentation networks. It uses the encoder with Atrous Spatial Pyramid Pooling (ASPP) module from DeepLabv3 [**?**] and depthwise separable convolution, then adds a decoder module. The ASPP module captures multi-scale contextual information by extracting convolutional features at four scales through atrous convolution. Atrous (a.k.a. dilated) convolution involves spacing out the values of the convolution kernels at different rates as a method of increasing the receptive field without increasing the computation requirements. Depthwise separable convolution is an approximation of standard convolution that performs 2D convolution on each input channel separately and stacks the results, then reduces the output volume to the same shape as standard convolution using a 1x1 convolution across all channels. It can be used to perform similar feature extraction with a large reduction in computational cost. The decoder module refines fine geometry details at object boundaries by using features at corresponding scales from the network backbone alongside the encoder output.

Our input images are captured by our in house MARVIN imaging setup which consists of a fixed curved mechanical arm spanning a 90 degree arc with evenly-spaced mounted cameras that point at a circular rotating platen which the product is placed upon. This setup is contained within a white enclosed housing with light panels mounted to its walls [**?**]. We perform our segmentation task on product images captured by MARVIN. The products may span an arbitrary set of categories. Background images that may contain small mechanical structures, such as parts of the platen, are also captured.

We make a few noteworthy modifications to DeepLabv3+ to better accommodate the semantic segmentation task for our use case:

- We use the concatenation of object image and background image as a 6-channel input volume. Compared to just the object image, or the subtraction of background image from object image, this results in better segmentation results in cases with thin object structures,

shadows, reflections, or visible background clutter. We find it worthwhile to incur the computational penalty of a 6-channel input volume vs. a 3-channel input volume.

- We replace the Xception encoder network found in standard DeepLabv3+ with ResNet50. For our use case, we use a larger feature extraction network and do not require the performance benefits of depthwise separable convolution. We believe that depthwise separable convolution is an approximation of convolution, and thus would only benefit us in terms of decreasing computation requirements. We found that ResNet50 performed better than Xception with our MARVIN images.
- In the ASPP module and the decoder, we replace all depthwise separable convolution with regular convolution. Again, we are willing to learn more parameters in order to improve our accuracy.
- We expand the number of atrous convolutions used in ASPP from four to eight. Instead of using $\{6, 12, 18\}$ as the rates of atrous convolution, we use rates of $\{2, 4, 6, 9, 12, 15, 18\}$ to capture more multi-scale contextual information.
- We further replace all the bilinear upsampling operations in the decoder with fractionally-strided convolution to preserve the ability to learn additional information during the upsampling processs.

The proposed network structure is illustrated in Figure 1.

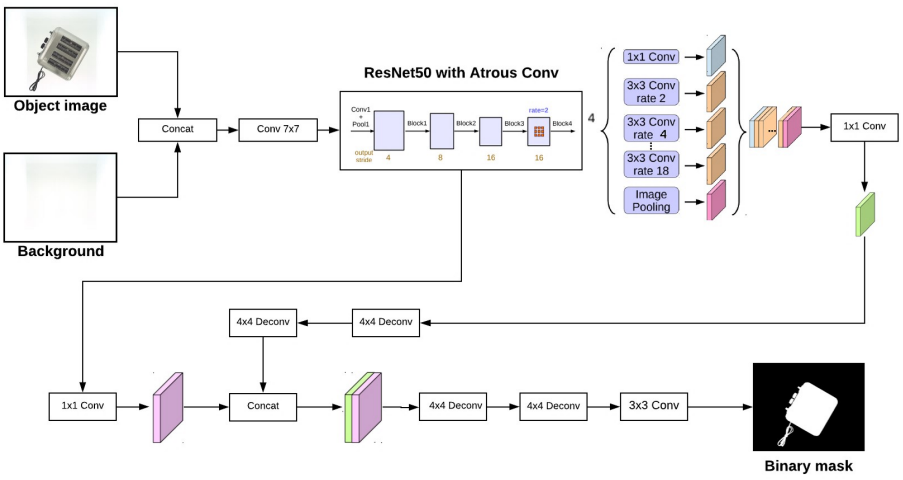
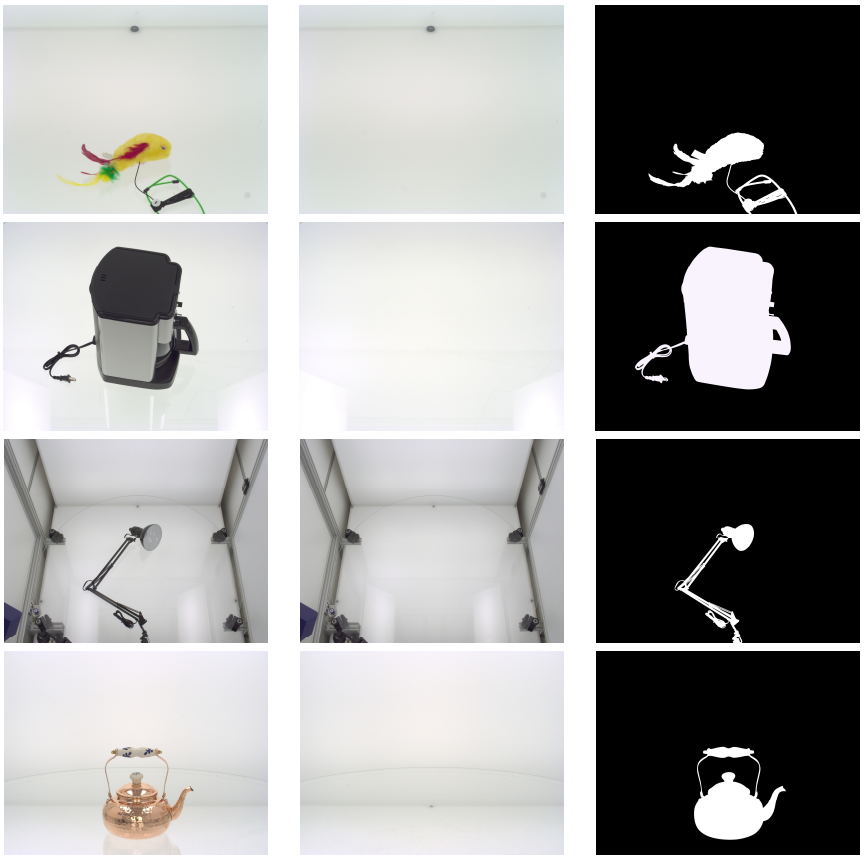


Figure 1: Network structure diagram.

3 Data sets

Our ground truth segmentation data sets were manually generated in house. They were created all from images captured by our product imaging rig, MARVIN [**?**] . For each product image, we also include a background image taken separately by the same camera without the product in the scene, as well as the manually created mask. From the MARVIN Segmentation 10k dataset, we have 238 products with 38 randomly sampled viewpoints each, and 5 products with 190 viewpoints each. From the MARVIN Kitchen 5k dataset, we have 78 products with 70 sampled viewpoints each, with higher density sampling at certain latitudes of the hemisphere. Please see Figure 2 for examples.



Original image Background image Mask image

Figure 2: Segmentation data set examples

4 Evaluation

4.1 Training details

For all of our experiments, the network is trained until convergence (usually about 500 epochs). We initialize our ResNet50 with pretrained ImageNet weights prior to training. All images are scaled to 512×640. We started training with the learning rate $lr = 0.00005$, then decay it by half every 200 epochs. No weight decay is implemented in our final implementation, as we noticed a degradation of performance when experimenting with it. The batch size is set to 24 images per mini batch. We use $1/3$ of our image data for validation during training.

As more MARVIN data became available while we worked on improving segmentation, we present results from training our network on different data sets. Results from the prior classical CV segmentation algorithm are also presented for reference. A summary of the evaluated algorithms follows.

- classical segmentation - the object image and background image and morphological operations are used to create an initial mask estimate.
- deeplabv3plus_dualinput_resnet50 8k - our proposed deep learning-based approach, trained on 8k randomly sampled images from our MARVIN Segmentation 10k dataset
- deeplabv3plus_dualinput_resnet50 16k - our proposed deep learning-based approach, trained on 16k randomly sampled images from our MARVIN Segmentation 10k and MARVIN Kitchen 5k dataset

4.2 Qualitative Results and Discussion

In this section, we share representative results and make observations about the performance of our approach. In general, the network has learned very well how to subtract the background from the image. We almost never see false positives of the background being considered as foreground in the mask. Even the reflections and shadows cast by the object onto the glass platen are almost always labeled correctly as background. We believe that the network has learned that spatially, reflections and shadows will reside below the object, as evidenced by seeing corresponding pixels in a vertically-inverted image being classified as foreground just from the change in orientation, not image data. Examples of predicted masks where our network performs well are shown in Figure 3. Thin structures, reflections, shadows, and translucent surfaces are learned well by the network.

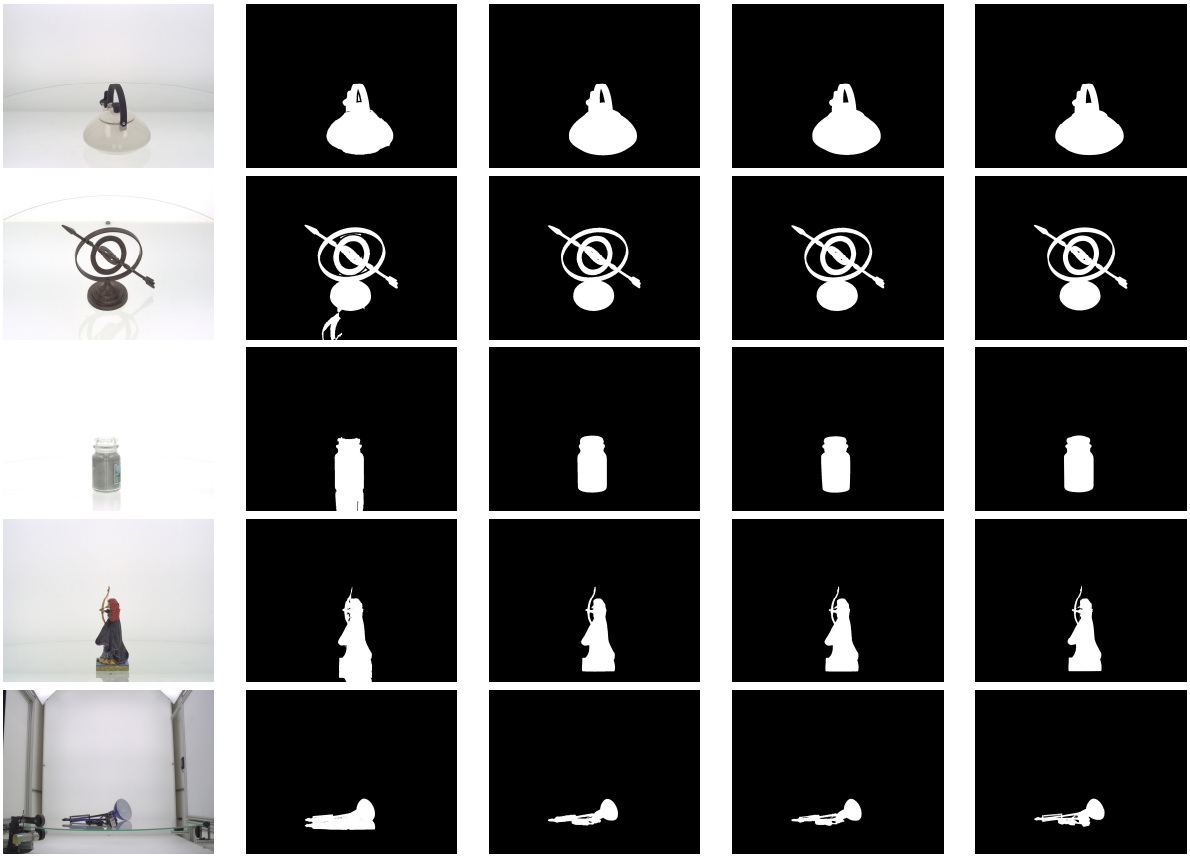
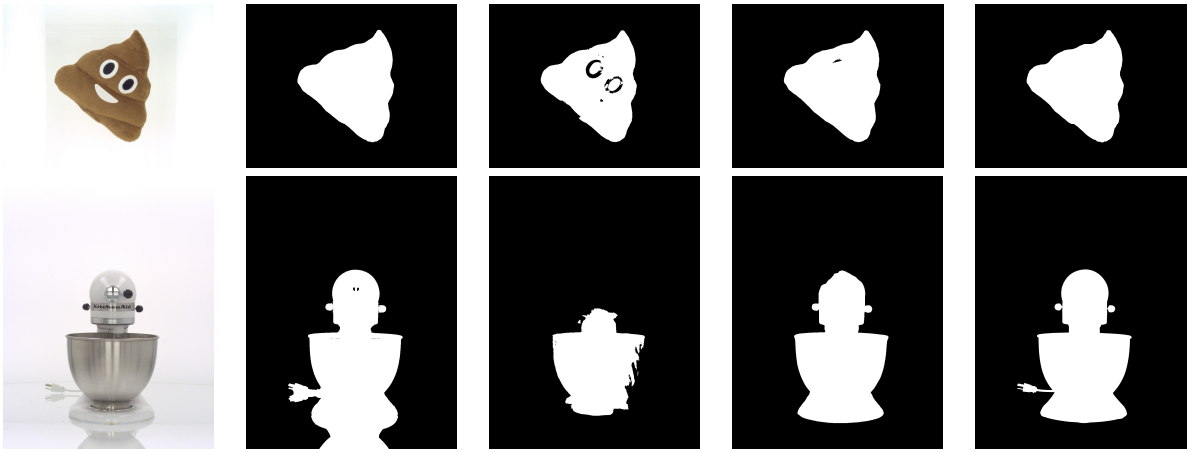


Figure 3: Segmentation results from classical and deep learned segmentation algorithms. The columns are **a.** original images, **b.** classical segmentation **c.** deeplabv3plus_dualinput_resnet50 8k **d.** deeplabv3plus_dualinput_resnet50 16k **e.** ground truth

The network has trouble mostly when a part of the foreground object has a white or shiny appearance similar to that of the background. The predicted masks shown in Figure 4 highlight the types of errors made by the network.

In comparing the results of our network trained on 8k images vs. on 16k images, we see that the latter tends to produce a more tightly cropped mask to the true object boundaries. We believe that this is due to the network having seen more types of features during training. This hypothesis is supported further by seeing the latter network also perform better on white surfaces and metallic objects, as it has been trained on many more examples of similar objects contained in the MARVIN Kitchen 5k data set.



a b c d e

Figure 4: Incorrect mask predictions by deep learned segmentation network. The columns are **a.** original images, **b.** classical segmentation **c.** deeplabv3plus_dualinput_resnet50 8k **d.** deeplabv3plus_dualinput_resnet50 16k e. ground truth

4.3 Quantitative Metrics

We compute standard segmentation metrics including Intersection over Union (IoU) and accuracy. Additionally, to better capture the algorithm's capability to capture boundary detail, we compute the average Hausdorff distance $\overline{d_H}$ from each segmentation mask's contour against the contour of the ground truth mask. For a contour X comprised of n pixels, we define $\overline{d_H}(X, Y)$ as the average shortest pixel distance from each point on X to another contour Y.

$$\overline{d_H}(X, Y) = \sum \frac{\{\sup_{x \in X} \inf_{y \in Y} d(x, y)\}}{n} \tag{1}$$

The test data set used to evaluate the algorithm consists of MARVIN images from 145 products not found in the training set. These products represent a variety types of objects of varying geometric complexity (e.g. thin structures, small holes, fine details, concavities), as well as material properties (e.g. transparent and reflective surfaces, white surfaces with similar appearance to the background) that is similar to the background). The metrics computed against this test set are found in Table 1

Table 1: Evaluation metrics

Method	**mIoU**	**mAccuracy**	**mHausdorff**
classical segmentation	0.8452	0.8491	28.8496
deeplabv3plus_dualinput_resnet50 8k	0.9373	0.9491	11.9959
deeplabv3plus_dualinput_resnet50 16k	0.9597	0.9697	7.9861

Mean metrics computed on our test set comparing our three methods.

5 Conclusion

We explore using a semantic segmentation network to generate high quality segmentation masks to be used in 3D structure estimation. A deep neural network based on the current state-of-the-art architecture DeepLabv3+ is proposed. We modify the network to suit our use case, favoring accuracy over speed, and replacing parts of the architecture accordingly while still leveraging the same general structure. We present two image segmentation datasets with highly accurate ground truth masks as well as background images. Our deep learning-based proposed approach has been deployed into production as of August 2018. It has improved the yield of our pipeline for generating 3D models for AR View by 30%.