Material-Based Segmentation of Objects

Jonathan Dyssel Stets, Rasmus Ahrenkiel Lyngby(✉), Jeppe Revall Frisvad, and Anders Bjorholm Dahl

Technical University of Denmark, 2800 Kgs. Lyngby, Denmark
{stet,raly,jerf,abda}@dtu.dk
http://eco3d.compute.dtu.dk/

Abstract. We present a data-driven proof of concept method for image-based semantic segmentation of objects based on their materials. We target materials with complex radiometric appearances, such as reflective and refractive materials, as their detection is particularly challenging in many modern vision systems. Specifically, we select glass, chrome, plastic, and ceramics as these often appear in real-world settings. A large dataset of synthetic images is generated with the Blender 3D creation suite and the Cycles renderer. We use this data to fine-tune the pretrained DeepLabv3+ semantic segmentation convolutional neural network. The network performs well on rendered test data and, although trained with rendered images only, the network generalizes so that the four selected materials can be segmented from real photos.

Keywords: Semantic segmentation · Deep learning · Synthetic data

1 Introduction

Semantic image segmentation is the task of labeling the parts of an image that contain specific object categories. Previous research has focused mainly on seg-

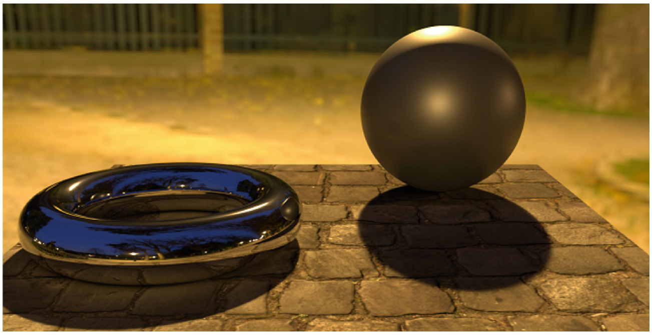

Fig. 1. Sample rendering of a scene. Two objects (torus and sphere) with materials assigned are placed on a cobble stone textured ground plane. All light in the scene comes from a spherical HDR image.

M. Felsberg et al. (Eds.): SCIA 2019, LNCS 11482, pp. 152–163, 2019.
https://doi.org/10.1007/978-3-030-20205-7_13

menting objects such as humans, animals, cars, and planes [3,4,6,16,20,22]. In this paper, we present a data-driven method for image-based semantic segmentation of objects based on their material instead of their type. Specifically, we choose to target glass, chrome, plastic, and ceramics because they frequently occur in daily life, have distinct appearances, and exemplify complex radiometric interactions. We generate synthetic images with reference data using the Blender 3D creation suite (Fig. 1) and train an existing Convolutional Neural Network (CNN) architecture, DeepLabv3+ [9], to perform semantic segmentation.

Generally, most vision systems perform best on Lambertian-like surfaces. This performance is because the appearance of an opaque, diffuse, and non-specular object is not dependent on the incident angle of surrounding light sources. Some materials can be hard to automatically detect due to their appearance. Specifically, objects with transparent or glossy materials are problematic as lights from the surroundings can refract or reflect upon interaction. Examples of these can be glass, plastics, ceramics, and metals whose appearances are highly determined by the surrounding illumination. Lightpath artifacts, such as specular highlights or subsurface scattering, can occur, resulting in a drastic change of appearance between viewing angles. These differences complicate the apprehension from a vision system resulting in false negative or inaccurate detections. Consequently, such materials are often avoided in research even though they often occur in real life settings.

With this paper, we show that it is possible to segment materials with complex radiometric properties from standard color images including the visually complex materials glass and chrome. This also means that the appearance difference between refraction (glass) and reflection (chrome) can be learned by a CNN. Finally, we provide a few examples visually indicating that segmentation of synthetic images of the four chosen materials generalize to real photos.

1.1 Related Work

Our study is inspired by researchers reconstructing the shape of glass objects using a dataset of synthetic images [28]. This work was based on an earlier investigation verifying that physically based rendering can produce images that are pixelwise comparable to photographs of specifically glass [27]. Considering this information, we wish to explore the potential of rendering of different materials with complex radiometric properties. In the following, we discuss existing work related to the topics of synthetic training data and material segmentation.

Synthetic Data. A large image set with reference data is required to properly train a CNN [17], but manually annotating the images to obtain reference data is a time-consuming process. Previous work has been successful in training with synthetic images and showing that the learned models generalize well to photographs [12,24]. Some of this work considers semantic segmentation [24], but the focus is labels related to an urban environment rather than materials. The appearance of materials depends on their reflectance properties, and rendering provides fine-tuned control over all parameters related to reflectance properties [21]. Image synthesis enables us to produce large scale, high precision, annotated datasets quickly. Several examples exist of such large scale synthetic datasets [18,26,29]. However, the ones including semantic segmentation reference data [18,26] do not have labels based on materials.

Materials in Data-Driven Models. As humans, we can typically identify a material by observing its visual appearance, but especially materials with complex reflectance properties have turned out to be difficult to segment. Several research projects address materials and their appearance in images. Georgoulis et al. [14] use synthetic images of specular objects as training data to estimate reflectance and illumination, Li et al. [19] recover the SVBRDF of a material also using rendered training data, and Yang et al. [30] use synthetic images containing metal and rubber materials as training data for visual recognition. These authors however do not consider segmentation. Bell et al. [5] target classification and segmentation of a set of specific materials like we do, but while our data is synthesized theirs is based on crowdsourced annotation of photographs.

2 Method

To make a good training set of synthetic images we have aimed at generating images that have a realistic appearance using a physical-based rendering model and realistic object shapes. Furthermore, we have strived at spanning a large variation by choosing largely varying environment maps.

2.1 Rendering Model

We generate a large synthetic dataset that consists of rendered images of a selection of shapes with different materials applied. This is done using the Cycles physically based renderer in Blender[1] v2.79. We construct a scene consisting of a textured ground plane, a number of shapes with applied materials and global illumination provided by a High Dynamic Range (HDR) environment map. To add additional variation, we randomly assign a camera position for each image. A sample rendering of the scene is shown in Fig. 1. The shapes, assigned materials, ground plane texture and environment map are interchangeable and controlled by a script. We describe each of the components in the following.

Shapes. We create a database of 20 shapes with varying geometry, while avoiding shapes that are too similar to the real world objects we later use for performance test. We strive to cover a broad range of shapes to both include convex- and concave-like shapes as well as soft and sharp corners to obtain a good variety of appearances. The shading is selected for each individual shape based on whether the material type maintains a realistic appearance for the given object. Each rendered image is based on one to three shapes being randomly positioned on the ground plane. We use five new shapes when rendering the test set.

[1] https://www.blender.org/. Accessed: January 30th 2019.

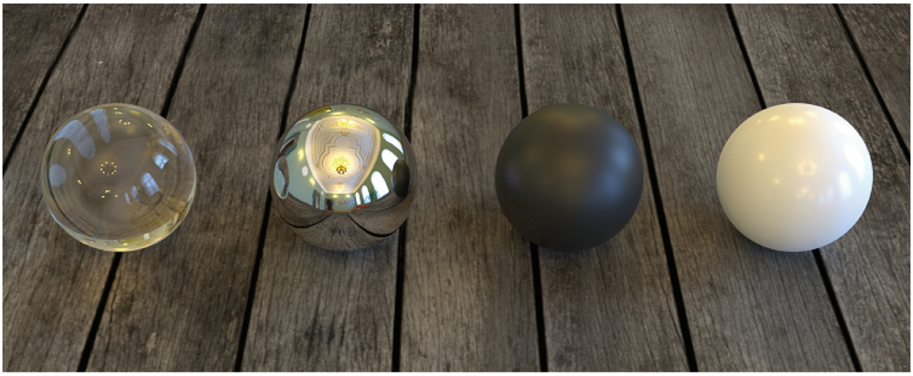

Fig. 2. Rendered samples of the four materials used in the dataset: glass, chrome, black plastic and white ceramic.

Materials. The following four materials are selected for evaluation: glass, chrome, black plastic, and white ceramic. These materials are targeted as we consider them to be complex in appearance while frequently occurring in real-life settings. We use built-in shaders provided by the Cycles renderer. Figure 2 exemplifies the appearances of the four materials.

Ground Plane. A ground plane is added to the scene and assigned a random texture from a database of 10 textures. The ground plane provides more accurate specular reflection of the nearby surface that real objects would usually stand on and better grounding of the objects due to inclusion of shadows. This adds an extra element of photorealism to the image. If caustics had been supported by the Cycles renderer, these would have appeared on the ground plane as well.

Environment Maps. Spherical HDR environment maps are used as the only illumination source in the scene. We use a total of seven environment maps and one of these is selected before each rendering. Both indoor and outdoor scenes are used to provide a variety in the type of illumination.

Images. The scenes are rendered as 640 × 480 RGB images with 8-bit color depth per channel. The images are rendered with 900 samples per pixel with a maximum of 128 bounces. To produce the reference label images of the scene, we switch off the global illumination and replace the material shaders with shaderless shaders. A color in the HSV-space is assigned for each material respectively by only varying the hue value. The result is an image with zero values for all background pixels and a color for all material pixels respectively as shown in Fig. 3. The label images are rendered with 36 samples per pixel and the images are post processed by thresholding a hue range to obtain a sharp delimiting border between label and background pixels.

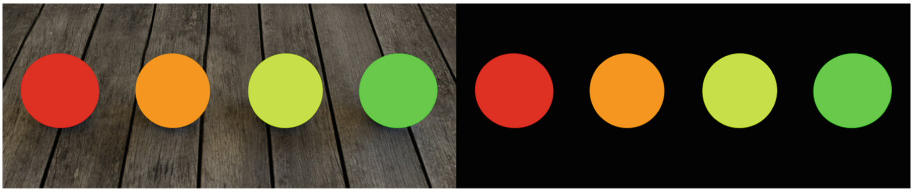

Fig. 3. Labels for **glass**, **chrome**, **plastic** and **ceramic**, respectively. Same image as in Fig. 2, but rendered with flat shading ("shaderless shader" resulting in a single uniform color). To the right, all illumination was removed from the scene. (Color figure online)

Dataset. We used our rendering model to generate a total of $m = 26{,}378$ scene images with accompanying label images. The following hue color values were used as labels for the materials: 0.0 = glass, 0.1 = chrome, 0.2 = black plastic, 0.3 = white ceramic. Each of the material-label colors has a Value, as specified in the HSV color-space, of 1.0 and the background has a Value of 0.0. Each pair of RGB and label images are accompanied by a metadata-file listing the objects and materials present in the respective scene. Based on a finding that $1/\sqrt{2m}$ is the optimal number of samples in the validation set [2], we choose a 99% to 1% training-validation split of the renderings. Additionally, we rendered a test set of 300 images with the same four materials but with four shapes that were neither in the training nor in the validation set. The ground plane textures and environment maps remain the same set across training, test, and validation set.

2.2 Segmentation Model

We decided to use DeepLabv3+ which is a state-of-the-art semantic segmentation network [9]. It is a goal for us to show that it is generally possible to segment materials based on synthetic images, and we therefore decided not to change the network's architecture or specialize it in any way. By doing so, we demonstrate both the broadness of DeepLabv3+'s application domain and the model's ability to learn real things from physically based renderings. We postulate that this ability is transferable to other kinds of networks and applications precisely because we did not design the system specifically to learn from rendered data.

DeepLabv3+ is as an encoder-decoder network. The encoder condenses the semantic information contained in the input image's pixels into a tensor. This information is then decoded, by the decoder, into a segmentation image with a class label assigned to each pixel and of the same size as the input image. The DeepLabv3 [8] network forms the encoder with its last feature map before logits as encoder output. This network is a combination of the Aligned Xception image classification network [10,23], which extracts high-level image features, and an Atrous Spatial Pyramid Pooling network [7], which probes the extracted features at multiple scales. Atrous convolutions are convolutions with a "spread-out" kernel that has holes in between the kernel entries. An image pyramid is constructed by varying the size of these holes. The decoder has several depthwise separable convolution layers [1], which take in both the encoder output and the feature map from one of the first convolutional layers.

The specific network structure is described in previous work [9]. We therefore only cover it in brief. Figure 4 illustrates the network layout. The encoder begins with extracting image features using a modified version of the Aligned Xception network. This version deepens the model and replaces all max-pooling layers with depthwise separable convolutions which significantly reduces the number of trainable weights compared to the original Xception model. The resulting feature extractor has 69 layers (including residual connection layers [15]). The output feature map has 2048 channels and is used to form the Atrous Spatial Pyramid. Three $3 \times 3 \times 2048$ Atrous Convolutions with three different strides are used together with one $1 \times 1 \times 2048$ convolution filter and one pooling layer. The combined output has five channels, one for each of the filters in the pyramid. The encoder then combines the channels into a one channel encoder output by applying a $1 \times 1 \times 5$ filter. This final output map is eight times smaller than the input image.

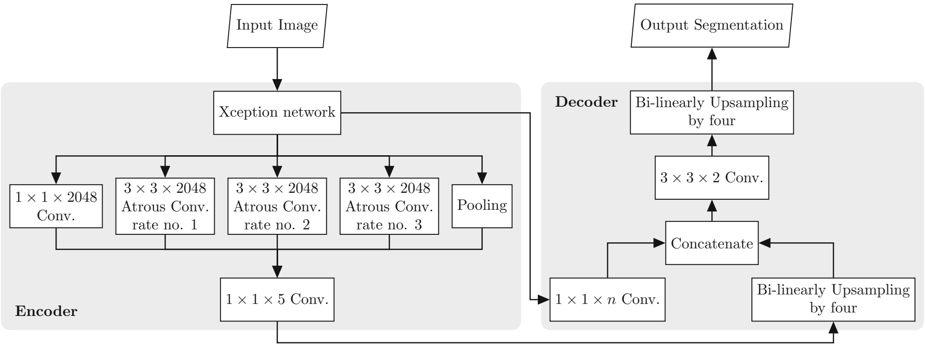
Input Image
Output Segmentation
Xception network
Decoder
Bi-linearly Upsampling by four
1 × 1 × 2048 Conv.
3 × 3 × 2048 Atrous Conv. rate no. 1
3 × 3 × 2048 Atrous Conv. rate no. 2
3 × 3 × 2048 Atrous Conv. rate no. 3
Pooling
3 × 3 × 2 Conv.
Concatenate
1 × 1 × 5 Conv.
Encoder
1 × 1 × n Conv.
Bi-linearly Upsampling by four

Fig. 4. Illustration of the semantic segmentation network.

The decoder up-samples the encoder output by a factor eight such that its size matches that of the input image. It starts by taking out one of the low-level feature maps from the Xception network, after the input image's size has been reduced four times, and applies a $1 \times 1 \times n$ filter to collapse it into one channel (where n is the number of channels in the feature map). Then the encoder output feature map is bi-linearly up-sampled by a factor of four and concatenated together with

Table 1. DeepLabv3+ settings

Property	Setting
Base learning rate	0.002
Learning rate decay factor	0.05
Train batch size	3
Atrous rate no. 1	6
Atrous rate no. 2	12
Atrous rate no. 3	18
No. of training steps	80,000

the Xception feature map. A 3 × 3 depthwise separable convolution is applied to the now two channel map which reduces it into one channel, and the result is upsampled four times to match the size of the input image. This result is the predicted semantic segmentation image.

Implementation. We adapted the DeepLabv3+ TensorFlow-based implementation[2] by Chen et al. [9] to train on our dataset. The Xception network is pre-trained on ImageNet [11,25] and the DeepLabv3+ network is pre-trained on the PASCAL VOC 2012 dataset [13]. Table 1 records our model settings.

3 Experiments and Results

We conducted three individual experiments. First, we tested the model on the 264 rendered images in our validation set. These images had never before been "seen" by the segmentation network but contained the same kind of objects as those in the training set. Second, we rendered a test set of 300 images with the same four materials but with new shapes that are not present in the training or validation set. The network's performance on this test set indicates whether it learned to distinguish physical appearance or if it somehow gets biased on object geometry. Third, we tested the model's performance on photographed real objects made of one of the four materials. This experiment investigated if the network can generalize from rendered images to actual photographs.

All components of our experimental setup are produced with off-the-shelf components. We use an open source rendering tool to produce our synthetic image data and use non-specific and straightforward geometry for our scenes. Floor textures and environment maps are downloaded from HDRI Haven[3] with gratitude. The TensorFlow Github (See footnote 2) kindly made the segmentation network available.

In general, our predictions show promising results. The mean Intersection Over Union (mIOU) score is used to indicate the performance on the rendered datasets. The validation set yielded an mIOU score of 95.69%, and the test set yielded 94.90%. The score is not computed for the real images since we did not have the ground truth semantic segmentations for these images. The scores indicate that the network is relatively good at predicting labels and that it seems not to be dependent on the shapes of the objects.

The following paragraphs showcase examples and discuss results from our three studies.

Validation Set. Figure 5 exhibits predicted segmentation masks from images in the validation set. We observe that the prediction is surprisingly good, even

[2] https://github.com/tensorflow/models/tree/master/research/deeplab. Accessed: January 30th 2019.

[3] https://hdrihaven.com/. Accessed: January 30th 2019.

for difficult materials such as glass and chrome. The networks ability to distinguish these kinds of objects is impressive, as the objects are not as such directly visible but instead reveal their presence by distorting light coming from the environment. The segmentation is, however, not perfect. Segmentation labels tend to "bleed" into each other when objects touch. This effect is seen in the middle row of Fig. 5. As seen in the bottom row, some environment maps caused over-saturation of the chrome material at certain view angles which caused the network to identify the material as white ceramic. Small objects and thin structures are difficult to segment and apt to disappear in the predictions.

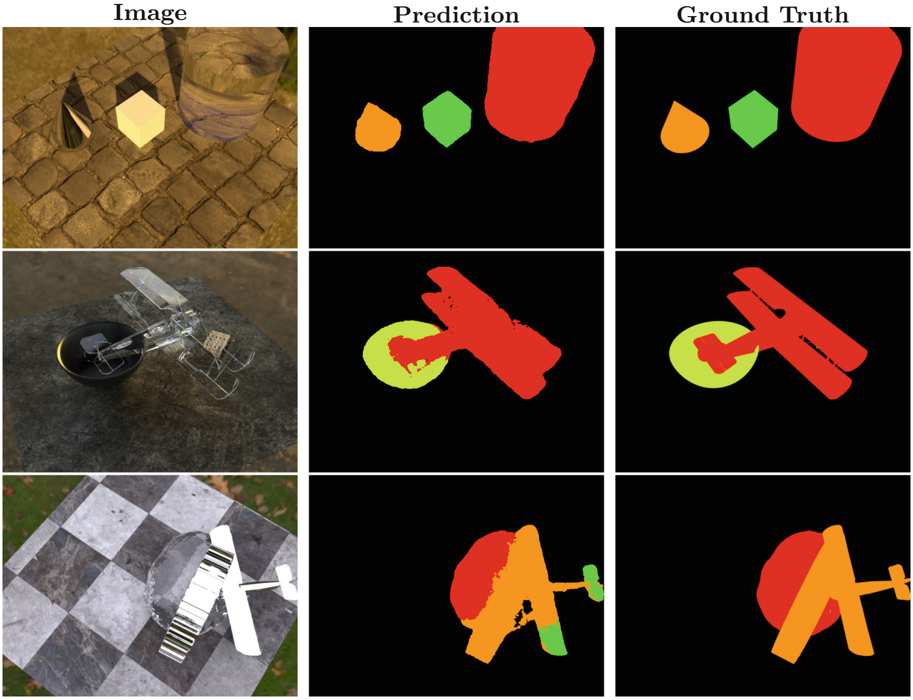

Fig. 5. Examples of segmentation results obtained on the validation set.

Test Set. The network's performance on the test set, with never before seen objects, are shown in Fig. 6. We observe that the performance is on par with that observed on the validation set in Fig. 5. This indicates that the material predictions are independent of object shape.

Real Images. Beyond the rendered test set, we captured three real images to test if the network can generalize to this data. Note that we do not have ground truth for these images, so the evaluation is solely by visual inspection. Results obtained on real images are in Fig. 7. Keeping in mind that we trained

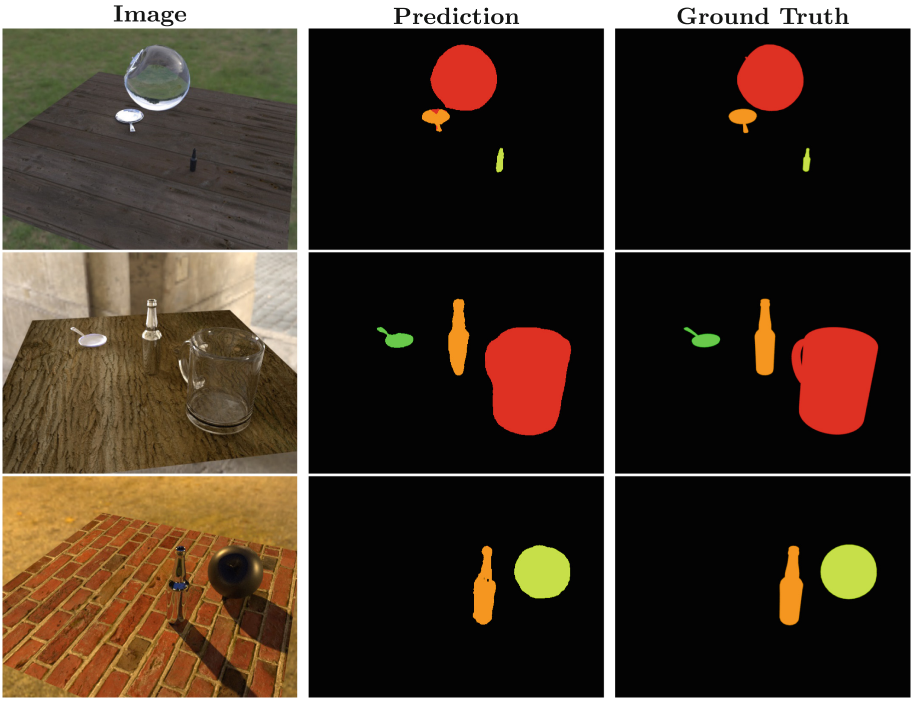

Fig. 6. Examples of segmentation results obtained on the test set.

the network for material segmentation only on rendered images, we find the results to be rather convincing. They are not perfect, but they are promising for the future potential of training networks with synthetic images in general and for material segmentation in particular.

The network found the glass objects with a good segmentation border, despite them being difficult to see with the human eye. Both chrome and plastic were segmented but with a few misclassified pixels as seen in the predicted images. The ceramic material seems to be the most difficult, and we suspect this is a result of our training data. Even though we rendered with a rather large number of samples per pixel, the images seem not to be fully converged and therefore do not reveal all specular highlights. Missing specular highlights is primarily an issue for the appearance of the ceramic material. Through further testing with real images, we also noted that the performance is highly dependent on the background. Non-textured surfaces, such as an office desk, confuse the network, which mistakes them for a specular material, often chrome, plastic, or glass. Additionally, the segmentation fails if the background is white, which we also believe to be an artifact of the too diffuse ceramic material.

The synthetic data could be improved in multiple ways. The materials we render are perhaps too "perfect" in their appearance. Adding random impurities, such as small scratches or bumps, to the materials would give a more realistic

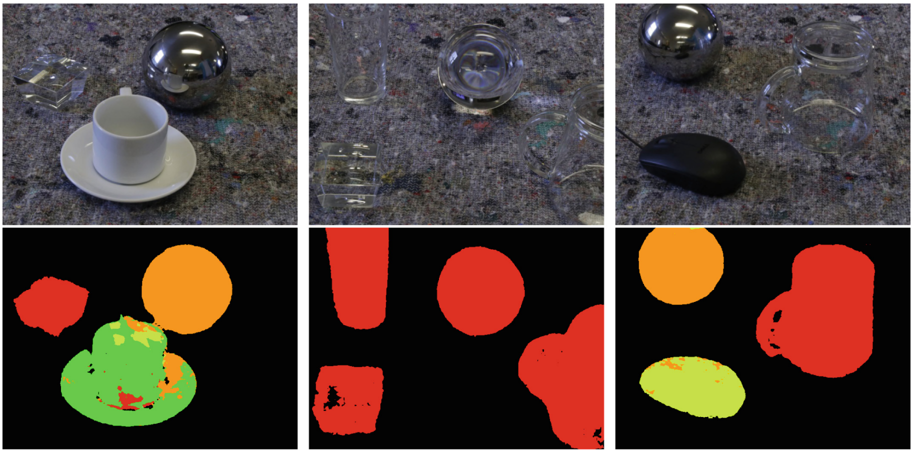

Fig. 7. Examples of segmentation results obtained from real images.

appearance. The environment maps are approximately 1k in resolution, resulting in a blurry background which gives a clear distinction between foreground objects and background. Real images with an in-focus background consequently result in predictions on this background. Thus, we believe higher resolution environment maps can help the network better distinguish between foreground and background in real photos. Finally, it is problematic that our included materials can occur in the environment maps without being labeled, which could hamper the network. This problem could potentially be mitigated either by making sure the environment maps are devoid of the targeted materials or by also generating those synthetically with material labels. Despite these current issues, we believe that the results of our study deliver a solid proof of concept with promising potentials for future work within semantic segmentation of complex materials.

4 Conclusion

We targeted the problem of segmenting materials in images based on their appearance. We presented a data-driven approach utilizing recent deep learning technology and rendering techniques to train a model on synthetic images. The learned model generalizes well to real photographs. The method allows us to detect specific materials in three channel color images without multi-spectral information. We achieved this feat using open source software which is freely available and requires no exceptional hardware components. Thus, the approach is available to anybody with a computer and a modern graphics card. Based on our results, and on the previous work that also uses synthetic training data, we firmly believe that physically based rendering is a vital component in the training of the deep learning models of tomorrow. Synthetic data generation is likely to push the boundaries of what deep learning can achieve even further.

Acknowledgements. The models we use are from turbosquid.com and the Make-Human software. Environment maps used to render our training data are from hdrihaven.com and textures for the ground plane are from texturehaven.com.

References

1. Abadi, M., et al.: TensorFlow: a system for large-scale machine learning. OSDI **16**, 265–283 (2016)
2. Amari, S., Murata, N., Müller, K.R., Finke, M., Yang, H.H.: Asymptotic statistical theory of overtraining and cross-validation. IEEE Trans. Neural Netw. **8**(5), 985–996 (1997)
3. Athanasiadis, T., Mylonas, P., Avrithis, Y., Kollias, S.: Semantic image segmentation and object labeling. IEEE Trans. Circuits Syst. Video Technol. **17**(3), 298–312 (2007)
4. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(12), 2481–2495 (2017)
5. Bell, S., Upchurch, P., Snavely, N., Bala, K.: Material recognition in the wild with the materials in context database. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3479–3487 (2015)
6. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. arXiv preprint arXiv:1412.7062 (2014)
7. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: 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)
8. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
9. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 833–851. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_49
10. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1800–1807 (2017)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255 (2009)
12. Dosovitskiy, A., et al.: FlowNet: learning optical flow with convolutional networks. In: IEEE International Conference on Computer Vision (ICCV), pp. 2758–2766 (2015)
13. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The PASCAL visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
14. Georgoulis, S., et al.: Reflectance and natural illumination from single-material specular objects using deep learning. IEEE Trans. Pattern Anal. Mach. Intell. **40**(8), 1932–1947 (2018)

15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
16. Kanade, T.: Region segmentation: signal vs semantics. Comput. Graph. Image Process. **13**(4), 279–297 (1980)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
18. Li, W., et al.: InteriorNet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In: British Machine Vision Conference (BMVC) (2018)
19. Li, Z., Sunkavalli, K., Chandraker, M.: Materials for masses: SVBRDF acquisition with a single mobile phone image. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11207, pp. 74–90. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01219-9_5
20. Liu, Z., Li, X., Luo, P., Loy, C.C., Tang, X.: Semantic image segmentation via deep parsing network. In: IEEE International Conference on Computer Vision (ICCV), pp. 1377–1385 (2015)
21. Nielsen, J.B., Stets, J.D., Lyngby, R.A., Aanæs, H., Dahl, A.B., Frisvad, J.R.: A variational study on BRDF reconstruction in a structured light scanner. In: IEEE International Conference on Computer Vision Workshop (ICCVW), pp. 143–152 (2017)
22. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: IEEE International Conference on Computer Vision (ICCV), pp. 1520–1528 (2015)
23. Qi, H., et al.: Deformable convolutional networks-COCO detection and segmentation challenge 2017 entry. In: ICCV COCO Challenge Workshop, vol. 15 (2017)
24. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The SYNTHIA dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3234–3243 (2016)
25. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
26. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic scene completion from a single depth image. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 190–198 (2017)
27. Stets, J.D., et al.: Scene reassembly after multimodal digitization and pipeline evaluation using photorealistic rendering. Appl. Opt. **56**(27), 7679–7690 (2017)
28. Stets, J.D., Li, Z., Frisvad, J.R., Chandraker, M.: Single-shot analysis of refractive shape using convolutional neural networks. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 995–1003 (2019)
29. Xu, Z., Sunkavalli, K., Hadap, S., Ramamoorthi, R.: Deep image-based relighting from optimal sparse samples. ACM Trans. Graph. (SIGGRAPH) **37**(4), 126 (2018)
30. Yang, J., Lu, J., Lee, S., Batra, D., Parikh, D.: Visual curiosity: learning to ask questions to learn visual recognition. Proc. Mach. Learn. Res. (CoRL) **87**, 63–80 (2018)