Fine Hand Segmentation using Convolutional Neural Networks

Tadej Vodopivec[1,2] Vincent Lepetit[1] Peter Peer[2]

[1] Institute for Computer Graphics and Vision, Graz University of Technology, Austria

[2] Faculty of Computer and Information Science, University of Ljubljana, Večna pot 113, 1000 Ljubljana, Slovenia

Abstract

We propose a method for extracting very accurate masks of hands in egocentric views. Our method is based on a novel Deep Learning architecture: In contrast with current Deep Learning methods, we do not use upscaling layers applied to a low-dimensional representation of the input image. Instead, we extract features with convolutional layers and map them directly to a segmentation mask with a fully connected layer. We show that this approach, when applied in a multi-scale fashion, is both accurate and efficient enough for real-time. We demonstrate it on a new dataset made of images captured in various environments, from the outdoors to offices.

1 Introduction

To ensure that the user perceives the virtual objects as part of the real world in Augmented Reality applications, these objects have to be inserted convincingly enough. By far, most of the research in this direction has focus on 3D pose estimation, so that the object can be rendered at the right location in the user's view [19, 8, 15]. Other works aim at rendering the light interaction between the virtual objects and the real world consistently [5, 14].

Significantly less works have tackled the problem of correctly rendering the occlusions which occur when a real object is located in front of a virtual one. [9] provides a method that requires interaction with a human and works only for rigid objects. [17] relies on background subtraction but this is prone to fail when foreground and background have similar colors. Depth cameras bring now an easy solution to handling occlusions, however, they provide a poorly accurate 3D reconstruction of the occluding boundaries of the real objects, which are essential for a convincing perception. The human perception is actually very sensitive to small deviations from the actual locations in occlusion rendering, making the problem very challenging [21].

With the development of hardware such as the HoloLens, which provides precise 3D registration and crisp rendering of the virtual objects, egocentric Augmented Reality applications can be foreseen to become very popular in the near future. This is why we focus here on correct rendering of occlusions by the user's hands of the virtual objects. More exactly, we assume that the hands are always in front of the virtual objects, which is realistic for many applications, and we aim at estimating a pixel-accurate mask of the hands in real-time.

The last years have seen the development of different segmentation methods based on Convolutional Neural Networks [12, 2]. While our method also relies on Deep Learning, its architecture has several fundamental differences. It is partially inspired by Auto-Context [18]: Auto-Context is a segmentation method in which a segmenter is iterated, and the segmentation result of the previous step is used in the next iteration in addition to the original image.

The fundamental difference between our approach and the original Auto-Context is that the initial segmentation is performed on a downscaled version of the input image. The resulting segmentation is then upscaled before being passed to the second iteration. This allows us to take the context into account very efficiently. We can also obtain precise localization of

the segmentation boundaries, because we avoid using pooling.

In the remainder of the paper, we first discuss related work, then describe our method, and finally present and discuss our results on a new dataset for hand segmentation.

2 Related Work

Hand segmentation is a very challenging task as hands can be very different in shape and skin color, look very different under another viewpoint, can be closed or open, can be partially occluded, can have different positions of the fingers, can be grasping objects or other hands, etc.

Skin color is a very obvious cue [1, 7], unfortunately, this approach is prone to fail as other objects may have a similar color. Other approaches assume that the camera is static and segment the hands based on their movement [3], use a simple or even single-color background [10], or rely on depth information obtained by an RGB-D camera [6]. None of these approaches can provide accurate masks in general conditions.

The method we propose is based on convolutional neural networks [11]. Deep Learning has already been applied to segmentation, and recent architectures tend to be made of two parts: The first part applies convolutional and pooling layers to the input image to produce a compact, low-resolution representation; the second part applies deconvolutional layers to this representation to produce the final segmentation, at the same resolution as the input image. This typically results in oversmoothed segments, which we avoid with our approach.

3 Method

In this section, we describe our approach. We first present our initial architecture based on multiscale analysis of the input. We then split this architecture in two to obtain our more efficient, final architecture. We finally detail our methodology to select the meta-parameters of this architecture.

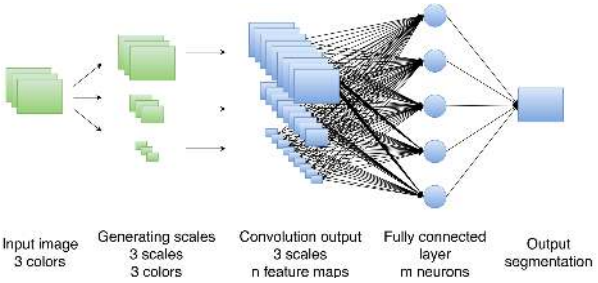

Figure 1: The architecture for the two components of our network. We extract features with convolutional layers without using pooling layers and map them directly to the output segmentation with a fully connected layer. For clarity, we show only one convolutional layer, and both the number of feature maps n and the number of neurons in the fully connected layer m are underrepresented.

3.1 Initial Network Architecture

As shown in Figure 1, our initial network was made of three chains of three convolution layers each. The first chain is directly applied to the input image, the second one to the input image after downscaling by a factor two, and the last one to the input image after downscaling by a factor four. We do not use pooling layers here, which allows us to extract the fine details of the hand masks.

The outputs of these three chains are concatenated together and given as input to a fully connected logistic regression layer, which outputs for each pixel its probability of lying on a hand.

3.2 Splitting the Network in Two

The network described above turned out to be too computationally intensive for real-time. To speed it up, we developed an approach that is inspired by Auto-Context [18]. Auto-Context is a segmentation method in which a segmenter is iterated, taking as input not only the image to segment but also the segmentation result of the previous iteration. The fundamental difference between our approach and the original Auto-Context is that the initial segmentation

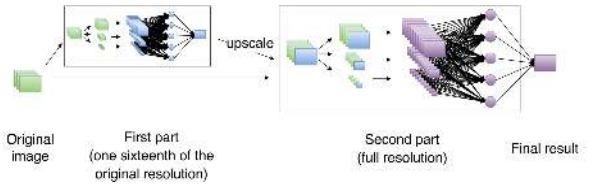

Figure 2: Two-part network architecture. As in Figure 1, only one convolutional layer is shown, and the number of feature maps and the number of neurons on the fully connected layer are underrepresented in both parts of the classifier to make the representation more understandable. The second part of the network receives as input the output of the first part after upscaling but also the original image, which helps segmenting fine details.

is performed on a downscaled version of the input image.

As seen in Figure 2, the first part performs the segmentation on the original image after downscaling by a factor 16, and outputs a result of the same resolution. Its output along with the original image is then used as input to the second part of the new network, which is a simplified version of the initial network to produce the final, full-resolution segmentation. The two parts of the network have very similar structures. The difference is that the second part takes as input the original, full resolution input image together with the output of the first part after upscaling. The first output already provides a first estimate of the position of the hands; the second part uses this information in combination with the original image to effectively segment the image. An example of the feature maps computed by the first part can be seen in Figure 4.

The advantage of this split is two-fold: The first part runs on a small version of the original image, and we can considerably reduce the number of feature maps and use smaller filters in the second part without loosing accuracy.

3.3 Meta-Parameters Selection

There is currently no good way to determine the optimal filter sizes and numbers of feature maps, so these have to be guessed or determined by trial and error. For this reason we trained networks with the same structure, but different parameters and compared the accuracy and running time. We first identified parameters that produced the best results while ignoring processing time and then simplified the model to reduce processing time while retaining as much accuracy as possible.

Our input images have a resolution of 752×480, scaled to 188×120, 94×60, and 47×30 and input to the first part of the network. The first two layers of each chain output 32 feature maps and the third layer outputs 16 feature maps. We used filters of size 3×3, 5×5, and 7×7 pixels for the successive layers. For the second part, the first layer outputs 8 feature maps, the second layer 4 feature maps, and the third layer outputs the final probability map. We used 3×3 filters for all layers.

We used the leaky rectified linear unit as activation function [13]. We minimize a boosted cross-entropy objective function [20]. This function weights the samples with lower probabilities more. We used $\alpha = 2$ as proposed in the original paper. We used RMSprop [4] for optimization.

To avoid overfitting and to make the classifier more robust, we augmented the training set using very simple geometric transformations: We used scaling by a random factor between 0.9 and 1.1, rotating for up to 10 degrees, introducing shear for up to 5 degrees, and translating by up to 20 pixels.

4 Results

In this section, we describe the dataset we built for training and testing our approach, and present and discuss the results of its evaluation.

4.1 Dataset

We built a dataset of samples made of pairs of images and their correct segmentation performed manually. Figure 5 shows some examples.

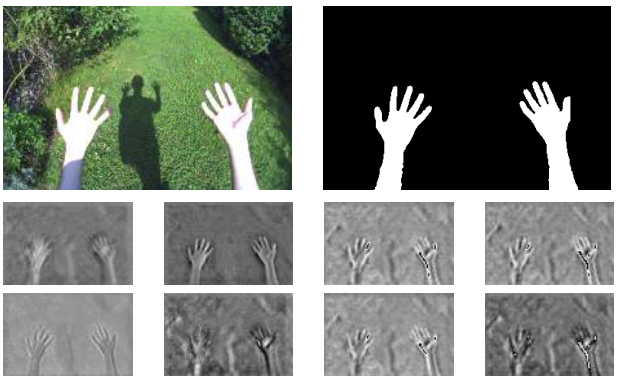

Figure 3: Original input image, its ground truth segmentation, and some of the resulting feature maps computed by the first part of the network.

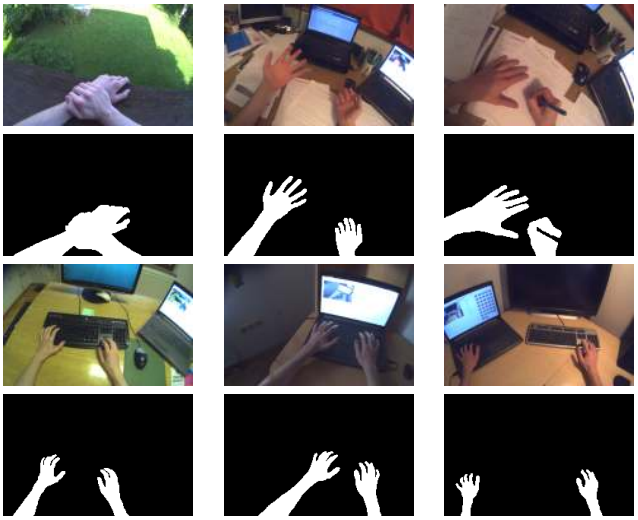

Figure 5: Some of the images from our dataset and their ground truth segmentations.

Figure 4: Example of output of the first part from the final architecture. Shades of grey represent the probabilities of the hand class over the pixel locations.

We focused on egocentric images, i.e. the hands are seen from a first-person perspective. Several subjects acquired these images using a wide-angle camera mounted on their heads, near the eyes. The camera was set to take periodic images of whatever was in the field of view at that time. In total 348 images were taken. 90% of the images were used for training, and the rest was used for testing.

191 of those images were taken in an office at 6 different locations under different lighting conditions. The remaining 157 images were taken in and around a residential building, while performing everyday tasks like walking around, opening doors etc. The images were taken with an IDS MT9V032C12STC sensor with resolution of 752×480 pixels.

4.2 Evaluation

Figure 6 shows the ROC curve for our method applied to our egocentric dataset. When applying a threshold of 50% to the probabilities estimated by our method, we achieve a 99.3% accuracy on our test set, where the accuracy is defined as the percentage of pixels that are correctly classified. Figure 7 shows that this accuracy can be obtained with thresholds from a large range of values, which shows the robustness of the method. Qualitative results can be seen in Figures 8 and 9.

4.3 Meta-parameter Fine Selection

In total, we trained 98 networks for the reduced resolution segmentation estimation and 95 networks for

(a) (b) (c) (d)

Figure 9: Images and their segmentations. (a) Original image; (b) Upscaled segmentation predicted by the first part of our network; (c) Final segmentation; (d) composition of the final segmentation into the original image.

the full resolution final classifier.

After meta-parameter fine selection for the first part of the network, we were able to achieve the accuracy of 98.3% at 16 milliseconds per image, where the first layer had 32 feature maps and 3×3 filter, the second layer 32 feature maps and 5×5 filters, and the third layer had 16 feature maps and 7×7 filters.

With further meta-parameters fine selection for the second part of the network, we obtained a network reaching 99.3% for a processing time of 39 milliseconds. The first layer (of the second part) had 8 feature maps, the second layer 4 feature maps, and the third layer 1 feature map. All layers used 3×3 filters.

During this fine selection process we noticed that the best results were achieved when the number of filters was higher for earlier layers and filter sizes

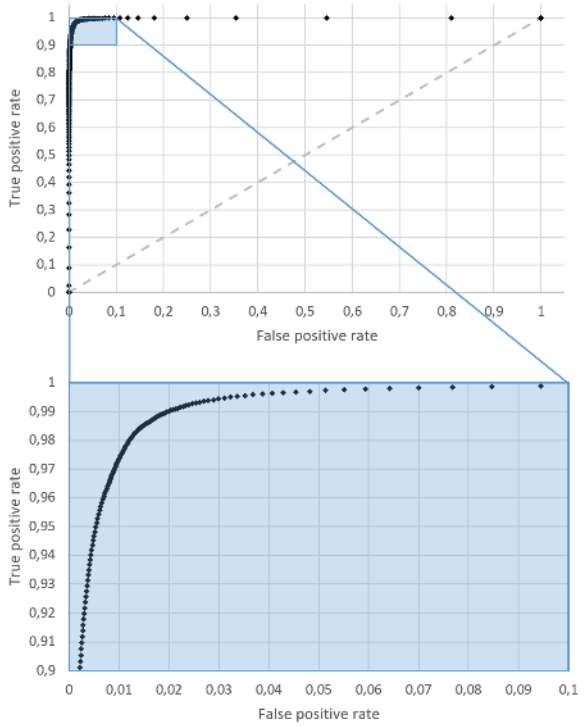

Figure 6: ROC curve obtained with our method on our challenging dataset of egocentric images. The figure also shows a magnification of the top-left corner.

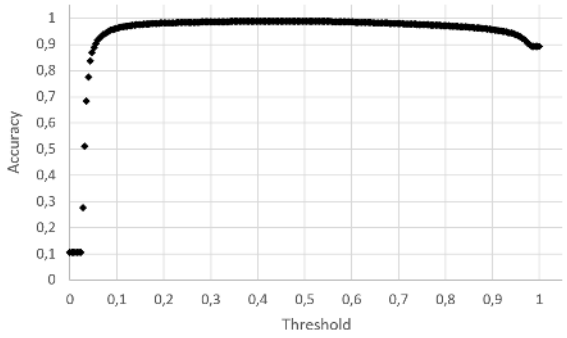

Figure 7: Accuracy of the classifier depending on probability threshold. Best accuracy can be obtained for a large range of thresholding values.

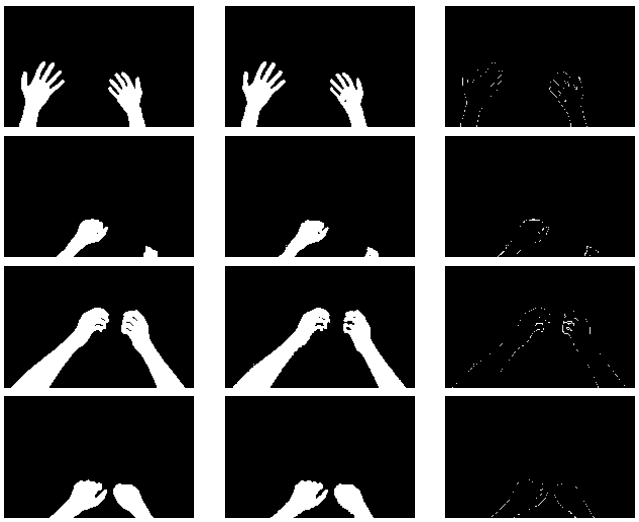

Figure 8: Comparison between the ground truth segmentations and the predicted ones. (a) Ground truth, (b) prediction, (c) differences. Errors are typically very small, and 1-pixel thin.

were bigger at later layers. The reduced resolution segmentation estimation already provided very good results, but it still produced some false positives. Because of the lower resolution the edges were not as smooth as desired. The full resolution final classifier was in most cases able to improve both the false positives and produce smoother edges.

4.4 Evaluation of the Different Aspects of the Method

4.4.1 Convolution on Full Resolution Without Pooling and Upscaling

To verify that splitting the classifier into two parts performs better than a more standard classifier, we trained a classifier to perform segmentation on full resolution images without first calculating the reduced resolution segmentation estimation.

We used the same structure as the second part of

our classifier and modified it to only use the original image. To compensate the absence of input from the first part, we tried using more feature maps. The best trade-off we found was using 16 feature maps and filter size of 5×5 pixels on each of the three layers—instead of 3×3 and 8, 4, and 1 feature maps. Because of memory size limit on the used GPU, we were not able to train a more complex classifier, which may produce better results. Nevertheless, processing time per image was 185 milliseconds with accuracy of 94.0%, significantly worse than the proposed architecture.

4.4.2 Upscale Without the Original Image

To verify that the second part of the classifier benefited from re-introducing the original image compared to only having results of the first part, we trained a classifier like the one suggested in this work, but this time we provided the second part of the classifier with only results of the first part. In this experiment processing time was 36.7 milliseconds, compared to 39.2 milliseconds in the suggested classifier and the accuracy fell from 99.3% to 98.6%. Processing was therefore faster, but the second part of the classifier was not able to improve the accuracy much further. The second part of the classifier was able to correct some false positives from the first part, but unable to improve accuracy along the edges between foreground and background.

4.5 Comparison to a Color-based Classification

As discussed in the introduction, segmentation based on skin color is prone to fail as other parts of the image can have similar colors. To give a comparison we applied the method described in [16] to our test set and obtained an accuracy of 81%, which is significantly worse than any other approach we tried.

5 Conclusion

Occlusions are crucial for understanding the position of objects. In Augmented Realist applications, their exact detection and correct rendering contribute to the feeling that an object is a part of the world around the user. We showed that starting with a low resolution processing of the image helps capturing the context of the image, and using the input image a second time helps capturing the fine details of the foreground.

References

[1] Z. Al-Taira, R. Rahmat, M. Saripan, and P. Sulaiman. Skin Segmentation Using YUV and RGB Color Spaces. *Journal of Information Processing Systems*, 10(2):283–299, 2014.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *arXiv Preprint*, 2015.

[3] L. Baraldi, F. Paci, G. Serra, L. Benini, and R. Cucchiara. Gesture recognition in ego-centric videos using dense trajectories and hand segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 688–693, 2014.

[4] Y. Dauphin, H. de Vries, J. Chung, and Y. Bengio. RMSProp and Equilibrated Adaptive Learning Rates for Non-Convex Optimization. In *arXiv*, 2015.

[5] P. Debevec. Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography. In *ACM SIGGRAPH*, July 1998.

[6] Y.-J. Huang, M. Bolas, and E. Suma. Fusing Depth, Color, and Skeleton Data for Enhanced Real-Time Hand Segmentation. In *Proceedings of the First Symposium on Spatial User Interaction*, pages 85–85, 2013.

[7] M. Kawulok, J. Nalepa, and J. Kawulok. Skin Detection and Segmentation in Color Images, 2015.

[8] G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *ISMAR*, 2007.

[9] V. Lepetit and M. Berger. A Semi Automatic Method for Resolving Occlusions in Augmented Reality. In *Conference on Computer Vision and Pattern Recognition*, June 2000.

[10] Y. Lew, R. A. Rhaman, K. S. Yeong, A. Roslizah, and P. Veeraraghavan. A Hand Segmentation Scheme using Clustering Technique in Homogeneous Background. In *Student Conference on Research and Development*, 2002.

[11] J. Lon, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[12] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2015.

[13] A. Maas, A. Hannun, and A. Ng. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *Proceedings of the International Conference on Machine Learning Workshop on Deep Learning for Audio, Speech, and Language Processing*, 2013.

[14] M. Meilland, C. Barat, and A. Comport. 3D high dynamic range dense visual slam and its application to real-time object re-lighting. In *International Symposium on Mixed and Augmented Reality*, pages 143–152, 2013.

[15] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *International Symposium on Mixed and Augmented Reality*, 2011.

[16] S. Phung, A. Bouzerdoum, and D. Chai. Skin Segmentation using Color Pixel Classification: Analysis and Comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):148–154, 2005.

[17] J. Pilet, V. Lepetit, and P. Fua. Retexturing in the Presence of Complex Illuminations and Occlusions. In *International Symposium on Mixed and Augmented Reality*, 2007.

[18] Z. Tu and X. Bai. Auto-Context and Its Applications to High-Level Vision Tasks and 3D Brain Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[19] L. Vacchetti, V. Lepetit, and P. Fua. Stable Real-Time 3D Tracking Using Online and Offline Information. *PAMI*, 26(10), October 2004.

[20] H. Zheng, J. Li, C. Weng, and C. Lee. Beyond Cross-Entropy: Towards Better Frame-Level Objective Functions for Deep Neural Network Training in Automatic Speech Recognition. In *Proceedings of the InterSpeech Conference*, 2014.

[21] S. Zollmann, D. Kalkofen, E. Mendez, and G. Reitmayr. Image-Based Ghostings for Single Layer Occlusions in Augmented Reality. In *International Symposium on Mixed and Augmented Reality*, 2010.