Segmentation of human hair

Assignment 3

Image Based Biometrics 2019/20, Faculty of Computer and Information Science, University of Ljubljana

Blaž Česnik

***Abstract*—**

I. Introduction

Computer vision has evolved dramatically in the past years and it is no secret that the reasons are new deep learning methods that revolutionized Computer Science. Today we use it for various tasks, for example different applications for image classification, face recognition, object detection for autonomous vehicles, video analysis and so on.

The goal of pixel wise segmentation is to output pixel-level masks in which regions belonging to certain categories are assigned the same distinct pixel value. We can than distinguish between different objects in pictures. We decided to create a model for segmentation, which segments the area of human hair. We used transfer learning to create our model and test if it does better work than model with random weights. The results were as expected and you can see them in Results section IV.

Figure 1. Semantic segmentation of traffic.

II. Methodology

Our work is based on transfer learning, which is a machine learning method where a model developed for a specific domain and is reused as the starting point for a model on a different domain. This is a popular approach in deep learning where we use pre-trained model, with already calculated weights for a specific domain, as a starting point for calculation of model. This method has become popular in recent years to improve the performance of a neural network trained on a small dataset. The example is seen in Figure 2.

In our case, we decided to focus only on human hair, so we started solving our problem by collecting annotated pictures of hair. First we tried to use subset of CelebA align and cropped dataset, annotated by Borza et al. [1] which consist of 3600 annotated pictures related to CelebA align and cropped dataset. The masks are in bitmap format, and we had big problems implementing it in our work. There were numerous errors when we tried to import it, so we decided to switch to different annotated dataset CelebA-HQ-Mask [2]. For fine-tuning our

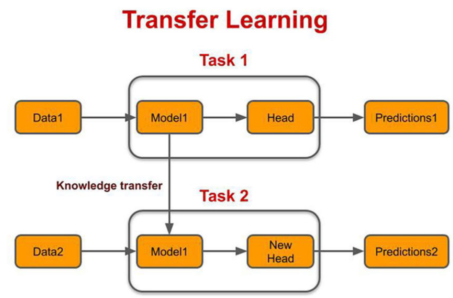

Figure 2. The process of transfer learning.

model, we used 2600 annotated pictures from CelebA-HQ-Mask [2] database and focused only on human hair domain. We extracted only annotated pictures of hair. The images were divided in two categories, train data with approximately 2000 annotated images and test data with 600 annotated images. Our data loader was design to receive images of same height and width and scaling them to 256x256 pixels. After that the values of image pixels were normalized. For implementing our work, we used Python [3] and deep learning library Pytorch [4].

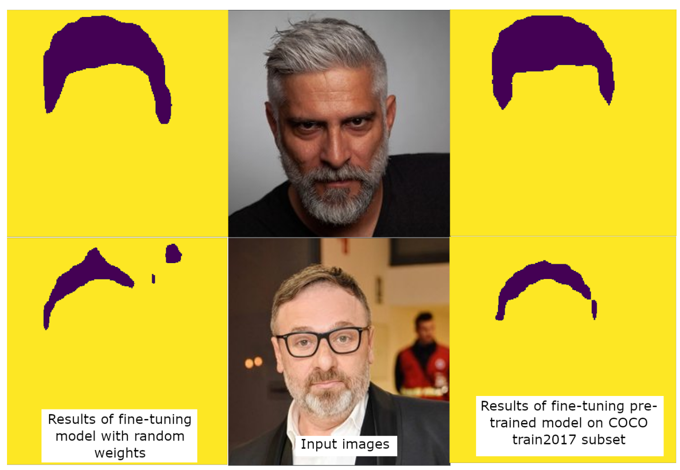

Figure 3. In the middle we see input images, and on the left and right is output of our two models.

DeepLabv3 was made by Google when they tried to rethink and reconstruct his ancestors DeepLabv1 and DeepLabv2. The model We used Deeplabv3-ResNet101 network, which is constructed by a Deeplabv3 [5] model with a ResNet-101 [6] backbone. The pre-trained model has been trained on a subset of COCO train2017 [7], on the 20 categories that are present in the Pascal VOC dataset. We then added a Sigmoid activation after the last convolution layer with one output channel. For computing the loss function on both models, we used Mean Square Error loss function, which functions by computing average squared difference between two pixels.

$$MeanSquareError = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2$$

Model based on pre-trained implementation, was back-propagated using the Adam optimizer and the learning rate of 10^{-4} and the model trained on random weights with learning rate 10^{-2}. This is because when we train already pre-trained model, we don't need to change the weights that much, because the model already has weights set for different domain and we just need smaller tweaks to make him work for our domain. It is different if we use random weights, for that we need bigger tweaks to train the model, cause the weights are randomized and do not suit any domain yet. Adam is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments [8].

For training the models, we used Asus laptop with Nvidia GeForce 1060 6GB graphic card and the processing was made with the help of Nvidia's CUDA API. For training of pre-trained model we used 15 epochs with 6 batches and the computing was finished in 6 hours. Training of model with random weights last for approximately 3 hours with 6 epochs and batch size of 6.

III. Conclusion

With pixel wise image segmentation we can develop many useful applications, which include Autonomous driving, Industrial inspection, classification of terrain visible in satellite imagery medical imaging analysis and many others. With our solution we could implement an application for changing color of human hair. The segmented pixels of hair are showing us the region of human hair, we would just need to use some filters for overlaying the segmented region over the original picture and choose the color for our new hair.

We can see on the graphs in Figure 4 and Figure 5 that we got more accurate results using pre-trained model. This is also visible in pictures in Figure 3, where we can see less false-positive predictions, so the usage of pre-trained model was definitely better option to do.

IV. Results

We fine-tuned two types of models, model with random weights and pre-trained model. We got better results on pre-trained model than on the model with random weights. On the middle in the figure 3, we can see the two input images we putted in our two models, on the left are output results of fine-tuned model on random weights and on the right are results of fine-tuned pre-trained model on COCO train2017 subset [7]. The train and test results of every epoch for pre-trained model are seen on Figure 4. We plotted the loss values, f1 score and AUROC *(Area Under the Receiver Operating Characteristics)*. The testing showed that the best achieved value of AUROC was 0.981631. We are quite happy with the results, although the edges of the segmented masks are quite superficial and the small parts of hair looking over head are not correctly segmented.

The model with random weights achieved AUROC of value 0.968331 which still good. The results of the model are visible on Figure 5.

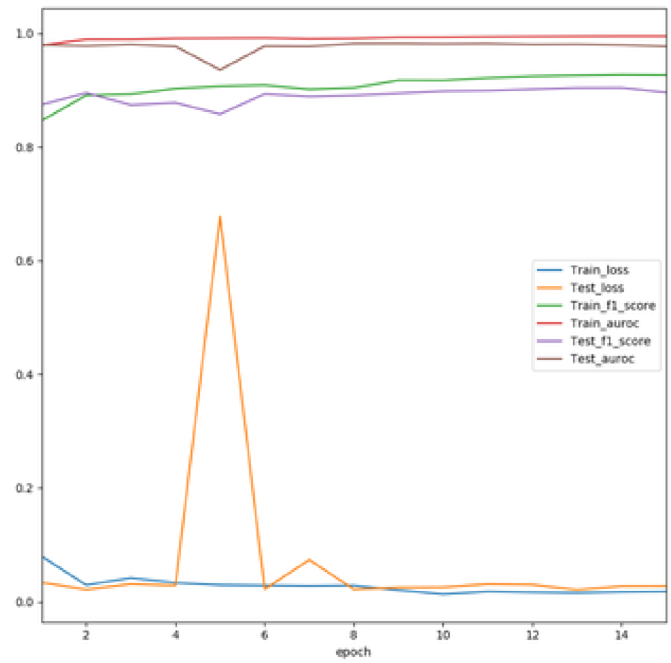

Figure 4. The results of our fine-tuned DeepLabv3-ResNet101 model pre-trained on COCO train 2017.

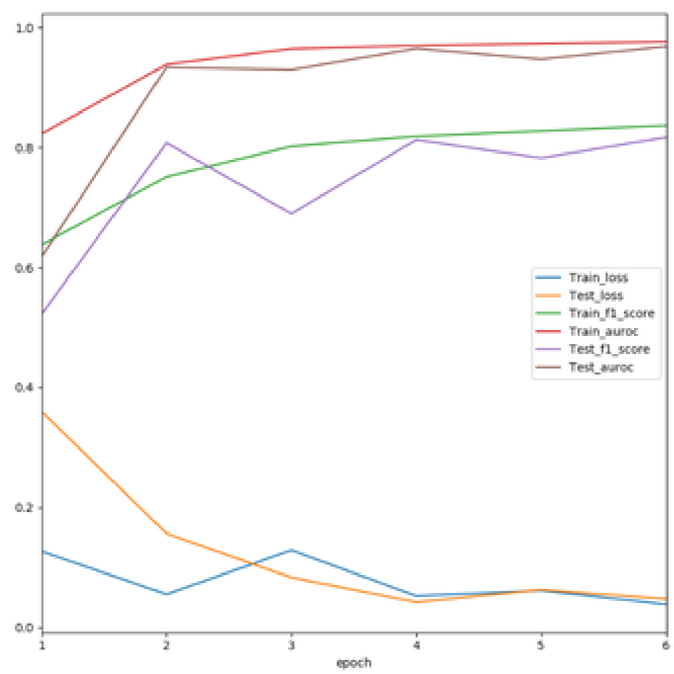

Figure 5. The results of our fine-tuned DeepLabv3-ResNet101 model with random weight initialization.

References

[1] D. Borza, T. Ileni, and A. Darabant, "A deep learning approach to hair segmentation and color extraction from facial images," in *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, 2018, pp. 438–449.

[2] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild." in *ICCV*. IEEE Computer Society,

2015, pp. 3730–3738. [Online]. Available: http://dblp.uni-trier.de/db/conf/iccv/iccv2015.html#LiuLWT15

[3] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2014. [Online]. Available: http://www.R-project.org/

[4] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[5] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *ArXiv*, vol. abs/1706.05587, 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980