EUROGRAPHICS 2016 / J. Jorge and M. Lin
(Guest Editors)

Volume 35 (2016), Number 2

Automatic Portrait Segmentation for Image Stylization

Xiaoyong Shen[1†], Aaron Hertzmann[2], Jiaya Jia[1], Sylvain Paris[2], Brian Price[2], Eli Shechtman[2] and Ian Sachs[2]

[1]The Chinese University of Hong Kong [2]Adobe Research

(a) Input Image (b) Segmentation (c) Stylization (d) Depth-of-field (e) Cartoon

Figure 1: *Our highly accurate automatic portrait segmentation method allows many portrait processing tools to be fully automatic. (a) is the input image and (b) is our automatic segmentation result. (c-e) show different automatic image stylization applications based on the segmentation result. The image is from the Flickr user "Olaf Trubel".*

Abstract

Portraiture is a major art form in both photography and painting. In most instances, artists seek to make the subject stand out from its surrounding, for instance, by making it brighter or sharper. In the digital world, similar effects can be achieved by processing a portrait image with photographic or painterly filters that adapt to the semantics of the image. While many successful user-guided methods exist to delineate the subject, fully automatic techniques are lacking and yield unsatisfactory results. Our paper first addresses this problem by introducing a new automatic segmentation algorithm dedicated to portraits. We then build upon this result and describe several portrait filters that exploit our automatic segmentation algorithm to generate high-quality portraits.

1. Introduction

With the rapid adoption of camera smartphones, the self portrait image has become conspicuously abundant in digital photography. A study by Samsung UK estimated that about 30% of smart phone photos taken were self portraits [Hal], and more recently, HTC's imaging specialist Symon Whitehorn reported that in some markets, self portraits make up 90% of smartphone photos [Mic].

The bulk of these portraits are captured by casual photographers who often lack the necessary skills to consistently take great portraits, or to successfully post-process them. Even with the plethora of easy-to-use automatic image filters that are amenable to novice photographers, good portrait post-processing requires treating the subject separately from the background in order to make the subject stand out. There are many good user-guided tools for creating masks for selectively treating portrait subjects, but these tools can still be tedious and difficult to use, and remain an obstacle for casual photographers who want their portraits to look good. While many image filtering operations can be used when selectively processing portrait photos, a few that are particularly applicable to portraits include background replacement, portrait style transfer [SPB*14], color and tone enhancement [HSGL11], and local feature editing [LCDL08]. While these can all be used to great effect with little to no user interaction, they remain inaccessible to casual photographers due to their reliance on a good selection.

A fully automatic portrait segmentation method is required to

† This work was done when Xiaoyong was an intern at Adobe Research.

make these techniques accessible to the masses. Unfortunately, designing such an automatic portrait segmentation system is nontrivial. Even with access to robust facial feature detectors and smart selection techniques such as graph cuts, complicated backgrounds and backgrounds whose color statistics are similar to those of the subject readily lead to poor results.

In this paper, we propose a fully automatic portrait segmentation technique that takes a portrait image and produces a score map of equal resolution. This score map indicates the probability that a given pixel belongs to the subject, and can be used directly as a soft mask, or thresholded to a binary mask or trimap for use with image matting techniques. To accomplish this, we take advantage of recent advances in deep convolutional neural networks (CNNs) which have set new performance standards for sematic segmentation tasks such as Pascal VOC [EGW*10] and Microsoft COCO [LMB*14]. We augment one such network with portrait-specific knowledge to achieve extremely high accuracy that is more than sufficient for most automatic portrait post-processing techniques, unlocking a range of portrait editing operations previously unavailable to the novice photographer, while simultaneously reducing the amount of work required to generate these selections for intermediate and advanced users.

To our knowledge, our method is the first one designed for automatic portrait segmentation. The main contributions of our approach are:

- We extend the FCN-8s framework [LSD14] to leverage domain specific knowledge by introducing new portrait position and shape input channels.
- We build a portrait image segmentation dataset and benchmark for our model training and testing.
- We augment several interactive portrait editing methods with our method to make them fully automatic.

2. Related Work

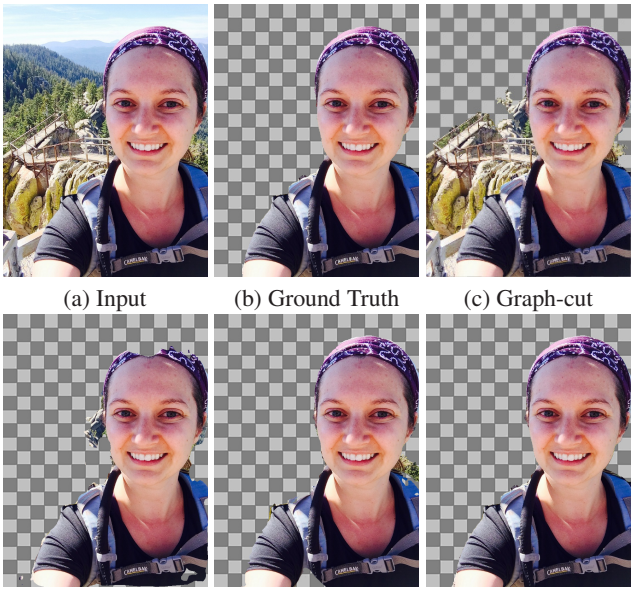
Our work is related to work in both image segmentation and image stylization. The following sections provide a brief overview on several main segmentation methodologies (interactive, learning based, and image matting), as well as some background on various portrait-specific stylization algorithms.

2.1. Interactive Image Selection

We divide interactive image segmentation methods into scribble-based, painting-based and boundary-based methods. In the scribble-based methods, the user specifies a number of foreground and background scribbles as boundary conditions for a variety of different optimizations including graph cut methods [BJ01, LSTS04, RKB04], geodesic distance scheme [BS07], random walks framework [Gra06] and the dense CRF method [KK11].

Compared with scribble-based methods, the painting based method only needs to paint over the object the user wants to select. Popular methods and implementations include painting image selection [LSS09], and Adobe Photoshop quick selection [ADO].

The object can also be selected by tracing along the boundary.



(a) Input (b) Ground Truth (c) Graph-cut

(d) FCN-8s (Person) (e) PortraitFCN (f) Our PortraitFCN+

Figure 2: *Different automatic portrait segmentation results. (a) and (b) are the input and ground truth respectively. (c) is the result of applying graph-cut initialized with facial feature detector data. (d) is the result of the FCN-8s (person class). (e) is the FCN-8s network fine-tuned with our portrait dataset and reduced to two output channels which we named as PortraitFCN. (f) is our new PortraitFCN model which augments (e) with portrait-specific knowledge.*

For example, Snakes [KWT88] and Intelligent Scissors [MB95] compute the object boundary by tracking the user's input rough boundaries. However, this requires accurate user interactions which can be very difficult, especially in the face of complicated boundaries.

Although the interactive selection methods are prevalent in image processing software, their tedious and complicated interaction limits many potentially automatic image processing applications.

2.2. CNNs for Image segmentation

A number of approaches based on deep convolutional neural networks (CNNs) have been proposed to tackle image segmentation tasks. They apply CNNs in two main ways. The first one is to learn the meaningful features and then apply classification methods to infer the pixel label. Representative methods include [AHG*12, MYS14, FCNL13], but they are optimized to work for a lot of different classes, rather than focusing specifically on portraits. As with our FCN-8s tests, one can use their "person class" for segmentation, but the results are not accurate enough on portraits to be used for stylization.

The second way is to directly learn a nonlinear model from the images to the label map. Long et al. [LSD14] introduce fully convolutional networks in which several well-known classification networks are "convolutionalized". In their work, they also introduce a skip architecture in which connections from early layers to lat-

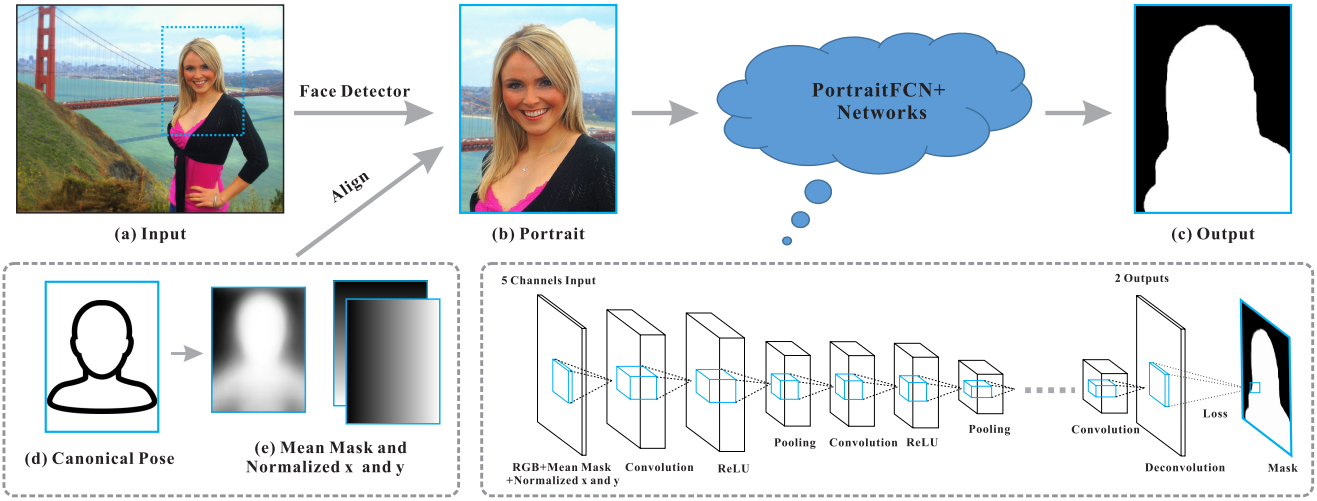


Figure 3: *Pipeline of our automatic portrait segmentation framework. (a) is the input image and (b) is the corresponding cropped portrait image by face detector. (d) is a template portrait image. (e) is the mean mask and normalized x- and y- coordinate. (c) shows the output with the PortraitFCN+ regression. The input to the PortraitFCN+ is the aligned mean mask , normailized x- and y-, and the portrait RGB channels.*

er layers were used to combine low-level and high-level feature cues. Following this framework, DeepLab [CPK*14] and CRFas-RNN [ZJR*15] apply dense CRF optimization to refine the CNNs predicted label map. Because deep CNNs need large-scale training data to achieve good performance, Dai et al. [DHS15] proposed the BoxSup which only requires easily obtained bounding box annotations instead of the pixel labeled data. It produced comparable results compared with the pixel labeled training data under the same CNNs settings.

These CNNs were designed for image segmentation tasks and the state-of-the-art accuracy for Pascal VOC is around 70%. Although they outperform other methods, the accuracy is still insufficient for inclusion in an automatic portrait processing system.

2.3. Image Matting

Image matting is the other important technique for image selection. For natural image matting, a thorough survey can be found in [WC07]. Here we review some popular works relevant to our technique. The matting problem is ill-posed and severely under-constrained. These methods generally require initial user defined foreground and background annotations, or alternatively a trimap which encodes the foreground, background and unknown matte values. According to different formulations, the matte's unknown pixels can be estimated by Bayesian matting [CCSS01], Poisson matting [SJTS04], Closed-form matting [LLW08], KNN matting [CLT13], etc. To evaluate the different methods, Rhemann et al. [RRW*09] proposed a quantitative online benchmarks. For our purposes, the disadvantages of these methods is their reliance on the user to specify the trimap.

2.4. Semantic Stylization

Our portrait segmentation technique incorporates high level semantic understanding of portrait images to help it achieve state of the art segmentation results which can then be used for subject-aware portrait stylization. Here we highlight a sampling of other works which also take advantage of portrait-specific semantics for image processing and stylization. [SPB*14] uses facial feature locations and sift flow to create robust dense mappings between user input portraits, and professional examples to allow for facial feature-accurate transfer of image style. In [LCODL08], a database of inter-facial-feature distance vectors and user attractiveness ratings is used to compute 2D warp fields which can take an input portrait, and automatically remap it to a more attractive pose and expression. And finally [CLR*04] is able to generate high-quality non-photorelistic drawings by leveraging a semantic decomposition of the main face features and hair for generating artistic strokes.

3. Our Motivation and Approach

Deep learning achieves state-of-the-art performance on semantic image segmentation tasks. Our automatic portrait segmentation method also applies deep learning to the problem of semantic segmentation, while leveraging portrait-specific features. Our framework is shown in Figure 3 and is detailed in Section 3.3. We start with a brief description of the fully convolutional neural network (FCN) [LSD14] upon which our technique is built.

3.1. Fully Convolutional Neutral Networks

As mentioned in the previous section, many modern semantic image segmentation frameworks are based on the fully convolutional neutral network (FCN) [LSD14] which replaces the fully connected layers of a classification network with convolutional layers. The

FCN uses a spatial loss function and is formulated as a pixel regression problem against the ground-truth labeled mask. The objective function can be written as,

$$\varepsilon(\theta) = \sum_{p} e(X_\theta(p), \ell(p)), \tag{1}$$

where p is the pixel index of an image. $X_\theta(p)$ is the FCN regression function in pixel p with parameter θ. The loss function $e(.,.)$ measures the error between the regression output and the ground truth $\ell(p)$. FCNs are typically composed of the following types of layers:

Convolution Layers This layer applies a number of convolution kernels to the previous layer. The convolution kernels are trained to extract important features from the images such as edges, corners or other informative region representations.

ReLU Layers The ReLU is a nonlinear activation to the input. The function is $f(x) = \max(0,x)$. This nonlinearity helps the network compute nontrivial solutions on the training data.

Pooling Layers These layers compute the max or average value of a particular feature over a region in order to reduce the feature's spatial variance.

Deconvolution Layers Deconvolution layers learn kernels to upsample the previous layers. This layer is central in making the output of the network match the size of the input image after previous pooling layers have downsampled the layer size.

Loss Layer This layer is used during training to measure the error (Equation 1) between the output of the network and the ground truth. For a segmentation labeling task, the loss layer is computed by the softmax function.

Weights for these layers are learned by backpropagation using stochastic gradient descent (SGD) solver.

(a) Input Image (b) FCN-8s (Person Class)

(c) PortraitFCN (d) PortraitFCN+

Figure 4: *Comparisons of FCN-8s applying to portrait data. (a) is the input image. (b) is the person class output of FCN-8s and (c) is the output of PortraitFCN. (d) is our PortraitFCN+.*

3.2. Understandings for Our Task

The fully convolutional network (FCN) used for this work is originally trained for semantic object segmentation on the Pascal VOC dataset for twenty class object segmentation. Although the dataset includes a person class, it still suffers from poor segmentation accuracy on our portrait image dataset as shown in Figure 4 (b). The reasons are mainly: 1) The low resolution of people in the Pascal VOC constrains the effectiveness of inference on our high resolution portrait image dataset. 2) The original model outputs multiple labels to indicate different object classes which introduces ambiguities in our task which only needs two labels. We address these two issues by labeling a new portrait segmentation dataset for fine-tuning the model and changing the label outputs to only the background and the foreground. We show results of this approach and refer to it in the paper as PortraitFCN.

Although PortraitFCN improves the accuracy for our task as shown in Figure 4 (c), it is still has issues with clothing and background regions. A big reason for this is the translational invariance that is inherent in CNNs. Subsequent convolution and pooling layers incrementally trade spatial information for semantic information. While this is desirable for tasks such as classification, it means that we lose information that allows the network to learn, for example, the pixels that are far above and to the right of the face in 4 (c) are likely background.

To mitigate this, we propose the PortraitFCN+ model, described next, which injects spatial information extracted from the portrait, back into the FCN.

3.3. Our Approach

Our approach incorporates portrait-specific knowledge into the model learned by our CNNs. To accomplish this, we leverage robust facial feature detectors [SLC09] to generate auxiliary position and shape channels. These channels are then included as inputs along with the portrait color information into the first convolutional layer of our network.

Position Channels The objective of these channels is to encode the pixel positions relative to the face. The input image pixel position only gives limited information about the portrait because the subjects are framed differently in each picture. This motivates us to provide two additional channels to the network, the *normalized x* and *y* channels where x and y are the pixel coordinates. We define them by first detecting facial feature points [SLC09] and estimating

a homography transform $\mathcal{T}$ between the fitted features and a canonical pose as shown in Figure 3 (d). We defined the normalized x channel as $\mathcal{T}(x_{\text{img}})$ where x_{img} is the x coordinate of the pixels with its zero in face center in the image. We define the normalized y channel similarly. Intuitively, this procedure expresses the position of each pixel in a coordinate system centered on the face and scaled according to the face size.

Shape Channel In addition to the position channel, we found that adding a shape channel further improves segmentation. A typical portrait includes the subject's head and some amount of the shoulders, arms, and upper body. By including a channel in which a subject-shaped region is aligned with the actual portrait subject, we are explicitly providing a feature to the network which should be a reasonable initial estimate of the final solution. To generate this channel, we first compute an aligned average mask from our training dataset. For each training portrait-mask pair $\{P_i, M_i\}$, we transform M_i using a homography $\mathcal{T}_i$ which is estimated from the facial feature points of P_i and a canonical pose. We compute the mean of these transformed masks as:

$$M = \frac{\sum_i w_i \circ \mathcal{T}_i(M_i)}{\sum_i w_i}, \tag{2}$$

where w_i is a matrix indicating whether the pixel in M_i is outside the image after the transform $\mathcal{T}_i$. The value will be 1 if the pixel is inside the image, otherwise, it is set as 0. The operator $\circ$ denotes element-wise multiplication. This mean mask M which has been aligned to a canonical pose can then be similarly transformed to align with the facial feature points of the input portrait.

Figure 3 shows our PortraitFCN+ automatic portrait segmentation system including the additional position and shape input channels. As shown in Figure 4, our method outperforms all other tested approaches. We will quantify the importance of the position and shape channels in Section 5.1.

4. Data and Model Training

Since there is no portrait image dataset for segmentation, we labeled a new one for our model training and testing. In this section we detail the data preparation and training schemes.

Data Preparation We collected 1800 portrait images from Flickr and manually labeled them with Photoshop quick selection. We captured a range of portrait types but biased the Flickr searches toward natural self portraits that were captured with mobile front-facing cameras. These are challenging images that represent the typical cases that we would like to handle. We then ran a face detector on each image, and automatically scaled and cropped the image to 600×800 according the bounding box of the face detection result as shown in Figure 3(a) and (b). This process excludes images for which the face detector failed. Some of the portrait images in our dataset are shown in Figure 5 and display large variations in age, color, background, clothing, accessories, head position, hair style, etc. We include such large variations in our dataset to make our model more robust to challenging inputs. We split the 1800 labeled images into a 1500 image training dataset and a 300 image testing/validation dataset. Because more data tends to produce better results, we augmented our training dataset by perturbing the rotations and scales of our original training images. We

Figure 5: *Some example portrait images with different variations in our dataset.*

synthesize four new scales $\{0.6, 0.8, 1.2, 1.5\}$ and four new rotations $\{-45°, -22°, 22°, 45°\}$. We also apply four different gamma transforms to get more color variation. The gamma values are $\{0.5, 0.8, 1.2, 1.5\}$. With these transforms, we generate more than 19,000 training images.

Model Training We setup our model training and testing experiment in Caffe [JSD*14]. With the model illustrated in Figure 3, we use a stochastic gradient descent (SGD) solver with softmax loss function. We start with a FCN-8s model which pre-trained on the PASCAL VOC 2010 20-class object segmentation dataset. While it is preferable to incrementally fine-tune, starting with the topmost layer and working backward, we have to fine-tune the entire network since our pre-trained model does not contain weights for the aligned mean mask and x and y channels in the first convolutional layer. We initialize these unknown weights with random values and fine-tune with a learning rate of 10^{-4}. As is common practice in fine-tuning neural networks, we select this learning rate by trying several rates and visually inspecting the loss as shown in Figure 6. We found that too small and too large learning rate did not successfully converge or over fitting.

Running Time for Training and Testing We conduct training and testing on a single Nvidia Titan X GPU. Our model training requires about one day to learn a good model with about 40,000 Caffe SGD iterations. For the testing phase, the running time on a 600×800 color image is only 0.2 second on the same GPU. We also run our experiment on the Intel Core i7-5930K CPU which takes 4 seconds using the MKL-optimized build of Caffe.

5. Results and Applications

Our method achieved substantial performance improvements over other methods for the task of automatic portrait segmentation. We provide a detailed comparison to other approaches. A number of applications are also conducted because of the high performance segmentation accuracy.

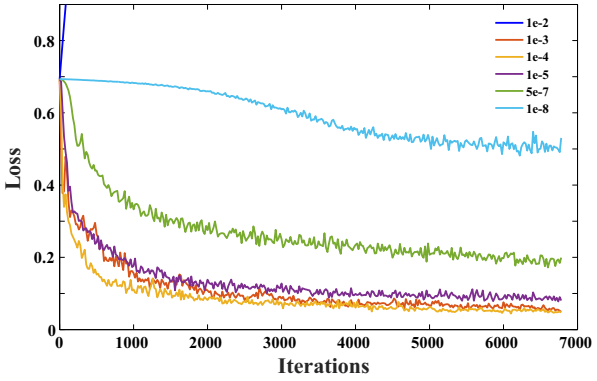


Figure 6: *The loss according to different learning rate. This helps us choosing the best learning rate to get the best model.*

Methods	Mean IoU
Graph-cut	80.02%
FCN (Person Class)	73.09%
PortraitFCN	94.20%
PortraitFCN+ (Only with Mean Mask)	94.89%
PortraitFCN+ (Only with Normalized x and y)	94.61%
PortraitFCN+	**95.91%**

Table 1: *Quantitative Comparison results of different automatic portrait segmentation method.*

5.1. Quantitative and Visual Analysis

Based on our labeled 300 testing images, we quantitatively compare our method with previous methods. The segmentation error is measured by the standard metric Interaction-over-Union (IoU) accuracy which is computed as the area of intersection of the output with the ground truth, divided by the union of their areas as,

$$\text{IoU} = \frac{\text{area}(\text{output} \cap \text{ground truth})}{\text{area}(\text{output} \cup \text{ground truth})}.$$

We first compare our method with a standard graph-cut method [BJ01]. In our graph-cut method, we start with an aligned mean mask similar to the one in our shape channel. This soft mask is aligned with the detected facial features [SLC09] of the input image and is used to set the unary term in our graph cut optimization.

We also run the result of fully convolutional network (FCN-8s) from [LSD14]. We look only at the results of the person class and ignore the remaining 19 class object labels. As reported in Table 1, the mean IoU accuracy in our testing dataset of graph-cut is 80.02% while the FCN-8s (Person Class) is only 73.09%. In our testing data, the graph-cut fails for examples whose background and foreground color distribution is similar, or in which the content (texture, details, etc) is complex. As for the FCN-8s, it fails because it has no consideration of the portrait data knowledge. PortraitFCN+, on the other hand, achieves 95.91% IoU accuracy which is a significant improvement.

In order to verify the effectiveness of the proposed position and shape channels, we setup our PortraitFCN+ model only with the mean mask channel and only with the normalized x and y channels. As shown in Table 1, considering the shape and position channels together achieves the best performance. The reason is that the position channels help to reduce the errors which are far from the face region while the shape channel helps to get the right label in the foreground portrait region. Figure 4 is an example of the type of highly distracting error corrected by the position and shape channels of PortraitFCN+. In fact, running PortraitFCN on our test set produces 60 segmentations which exhibit this type of error, while that number drops to 6 when using PortraitFCN+.

(a) Scale (b) Rotation (c) Color (d) Occlusion

Figure 8: *Our PortraitFCN+ model is robust to scale, rotation, color and occlusion variations. The top row is the input and the bottom one is the output.*

Besides the quantitative analysis, we also show some visual comparison results in Figure 7 from our testing dataset which reinforces the quantitative analysis. More results are provided in our supplementary file.

Our automatic portrait segmentation system is robust to the portrait scale, rotation, color and occlusion variations. This is because our proposed position and shape channels allow the network to take these variations into account. Further, the consideration of these variations in our training dataset also helps. We show some examples in Figure 8.

5.2. Post-processing

Our estimated segmentation result provides a good trimap initialization for image matting. As shown in Figure 9, we generate a trimap by setting the pixels within a 10-pixel radius of the segmentation boundary as the "unknown". KNN matting [CLT13] is performed with the trimap shown in (b) and the result is shown in (c). The matting performs very well in part because our segmentation provides an accurate initial segmentation boundary.

5.3. User Study of Our Method

The average 95.9% IoU portrait segmentation accuracy means that most of our results are close to the ground truth. In the cases where

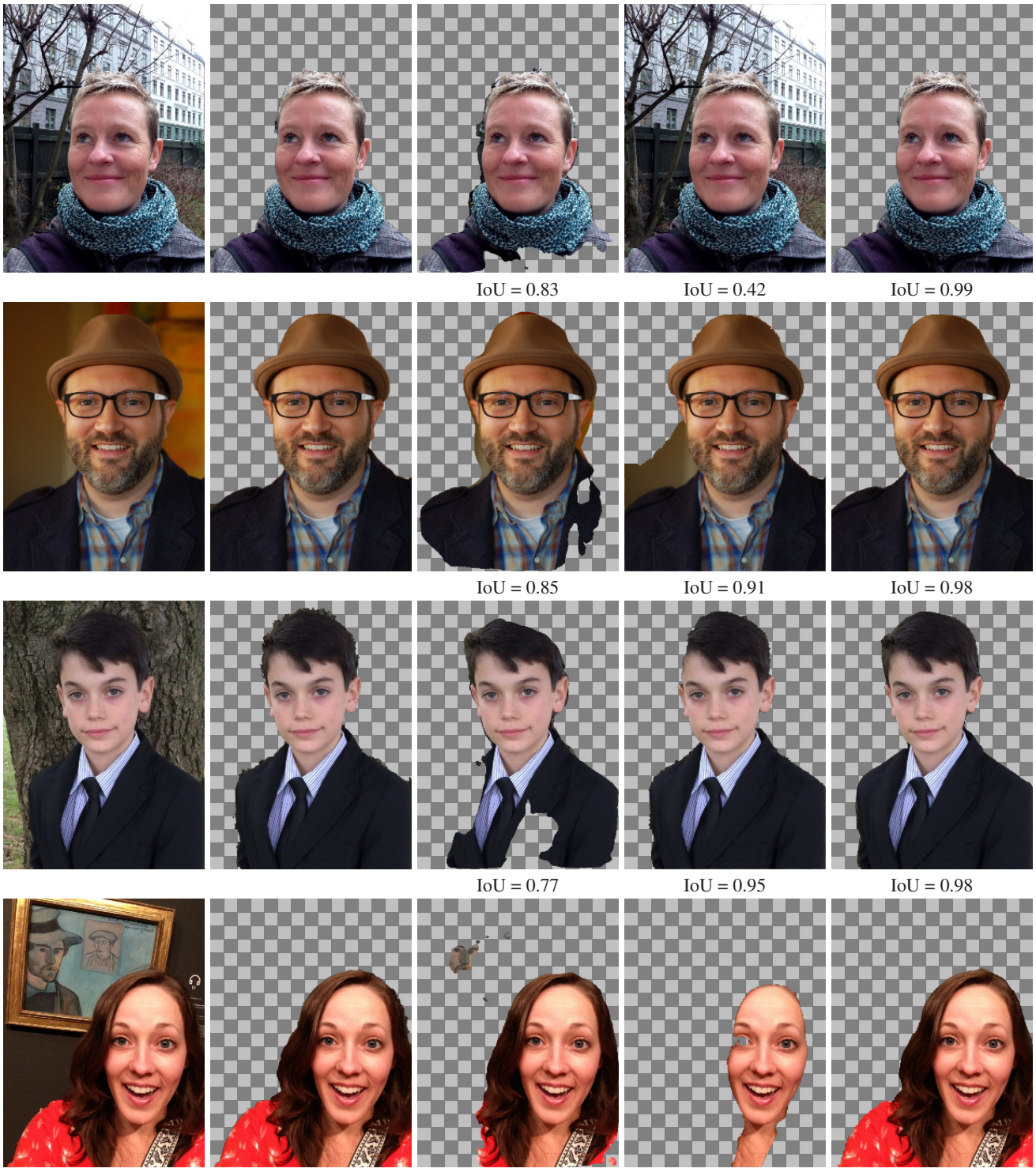


Figure 7: *Comparisons of different automatic portrait segmentation methods. (a) and (b) are the inputs and ground truth respectively. (c) is the results of FCN-8s (Person Class) and (d) is the graph-cut results. (e) is our PortraitFCN+. We will show all the results of our testing dataset in our supplementary file.*

(a) Input Image (b) Trimap (c) KNN Matting

Figure 9: *Segmentation as matting initialization. (a) is the input image and (b) is the trimap directly from our segmentation result. (c) is the KNN matting result from our trimap.*

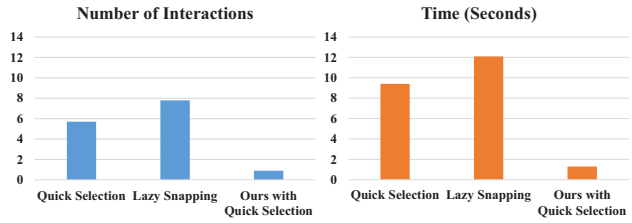


Figure 10: *User study of different interactive image selection system.*

there are small errors, they can be quickly corrected using interactive methods that are initialized with our method's result. The corrections are very fast when compared with starting from scratch. In order to verify this, we collected the number of interactions and time taken for a user to get the foreground selection with different interactive selection methods. 40 users with different backgrounds conducted selections for the 50 images from our testing dataset. We ask them to do the same thing using Photoshop quick selection [ADO] and lazy snapping [LSTS04] for each image. We also let the users do the quick selection initialized with our segmentation results. As shown in Figure 10, the number of interactions and time cost is largely reduced when compared with the quick selection and lazy snapping starting from only the original image.

5.4. Automatic Segmentation for Image Stylization

Due to the high performance of our automatic portrait segmentations, automatic portrait stylization schemes can be implemented in which the subject is considered independently of the background. Such approaches provide increased flexibility in letting the subject stand out while minimizing potentially distracting elements in the background.

We show a number of examples in Figure 12, using the stylization methods of [SPB*14] and [Win11], as well as several several Photoshop [ADO] filters with varying effects such as Palette Knife, Glass Smudge Stick and Fresco. After applying the filters to the portrait subject, we either perform background replacement, or we reapply the method that was used on the subject, but with different settings to weaken the background's detail and draw the viewer's

(a) Input (b) BG Replacement (c) BW one color

Figure 12: *Our automatic portrait segmentation also benefits for background editing. (a) is the input. (b) and (c) are the background replacement and black-and-white with one color [WSL12] result respectively.*

attention to the subject. Because of our segmentation accuracy, our results have no artifacts across the segmentation boundaries and allow for precise control of the relative amount of focus on the subject with minimal user interaction.

5.5. Other Applications

In addition to allowing for selective processing of foreground and background pixels, our approach also make background replacement trivial. As shown in Figure 12 (b), we automatically replace the portrait background. In (c), a black-and-white with one color [WSL12] is automatically generated.

The ability to eliminate the background can also help with other computer graphics and vision tasks, for example by limiting distracting background information in applications such as 3D face reconstruction, face view synthesis, and portrait style transfer [SPB*14].

6. Conclusions and Future Work

In this paper we propose a high performance automatic portrait segmentation method. The system is built on deep convolutional neural network which is able to leverage portrait specific cues. We construct a large portrait image dataset with enough portrait segmentation and ground-truth data to enable effective training and testing of our model. Based on the efficient segmentation, a number of automatic portrait applications are demonstrated. Our system could fail when the background and foreground have very small contrast. We treat this as the limitation of our method. In the future, we will improve our model for higher accuracy and extend the framework to the portrait video segmentation.

Acknowledgements

We thank the anonymous reviewers for their suggestions. We also thank Flickr users "Olaf Trubel", "Woodleywonderworks", "RD Glamour Photography" and "Justin Law" for the pictures used in the paper.

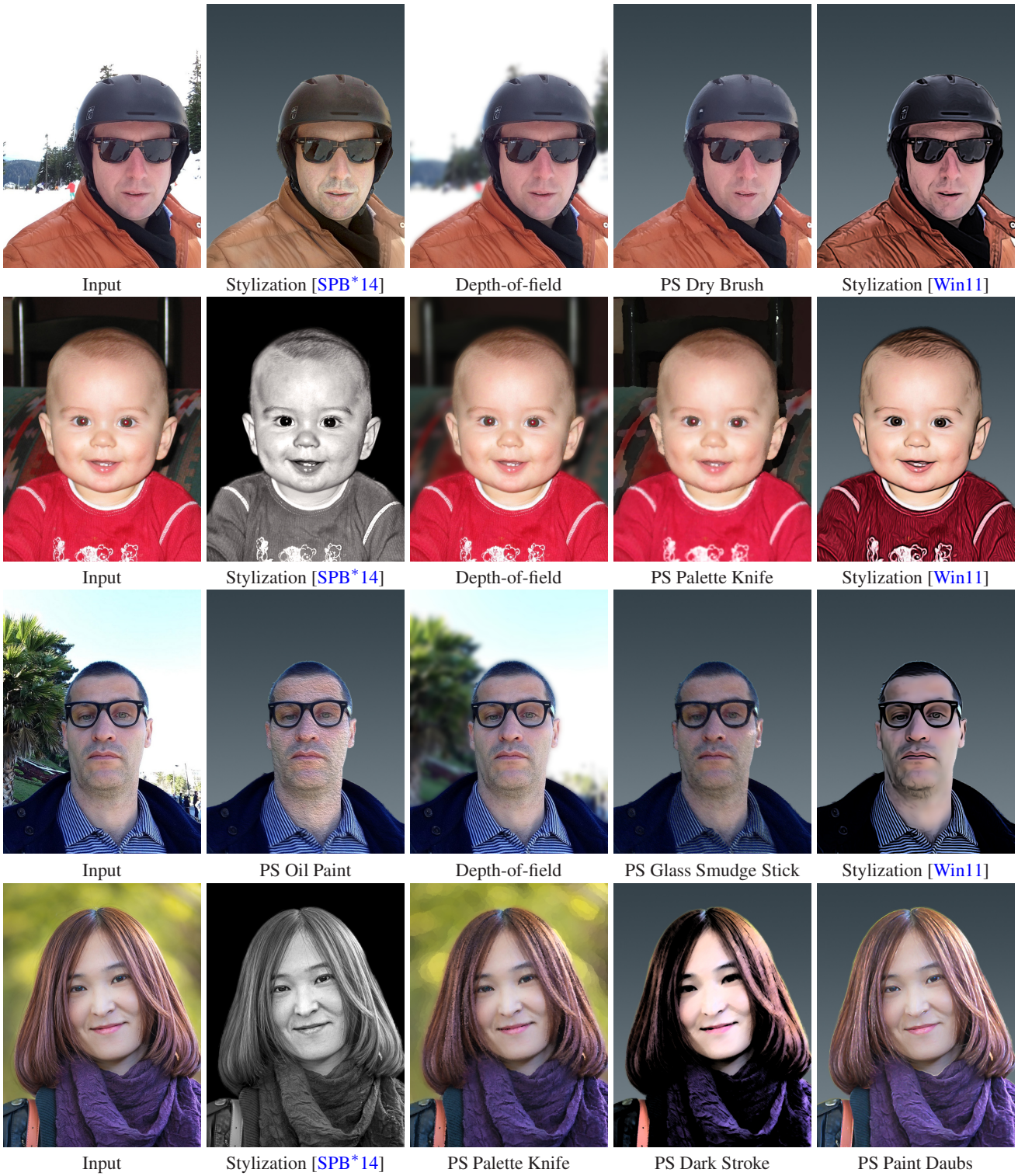


Figure 11: *A few examples of semantic portrait filters that differentiate the subject from the background. A typical effect is background replacement to deal with cases where the input background is cluttered. Another useful possibility enabled by our approach is applying a coarse-scale effect on the background and a finer-scale filter on the subject to make it stand out. We build our filters upon the stylization techniques of Shih et al. [SPB*14] and Winnemoeller et al. [Win11], and on some Photoshop filters (prefixed with PS). We encourage the reader to look at these results in the electronic document and to zoom in to better appreciate the details.*

References

[ADO] ADOBE SYSTEMS: Adobe photoshop cc 2015 tutorial. 2, 8

[AHG*12] ARBELAEZ P., HARIHARAN B., GU C., GUPTA S., BOURDEV L. D., MALIK J.: Semantic segmentation using regions and parts. In *CVPR* (2012), pp. 3378–3385. 2

[BJ01] BOYKOV Y. Y., JOLLY M.-P.: Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *ICCV* (2001), vol. 1, pp. 105–112. 2, 6

[BS07] BAI X., SAPIRO G.: A geodesic framework for fast interactive image and video segmentation and matting. In *ICCV* (2007), pp. 1–8. 2

[CCSS01] CHUANG Y., CURLESS B., SALESIN D., SZELISKI R.: A bayesian approach to digital matting. In *CVPR* (2001), pp. 264–271. 3

[CLR*04] CHEN H., LIU Z., ROSE C., XU Y., SHUM H.-Y., SALESIN D.: Example-based composite sketching of human portraits. In *NPAR* (2004), pp. 95–153. 3

[CLT13] CHEN Q., LI D., TANG C.: KNN matting. *IEEE Trans. Pattern Anal. Mach. Intell. 35*, 9 (2013), 2175–2188. 3, 6

[CPK*14] CHEN L., PAPANDREOU G., KOKKINOS I., MURPHY K., YUILLE A. L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. *ICLR* (2014). 3

[DHS15] DAI J., HE K., SUN J.: Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. *CVPR* (2015). 3

[EGW*10] EVERINGHAM M., GOOL L. J. V., WILLIAMS C. K. I., WINN J. M., ZISSERMAN A.: The pascal visual object classes (VOC) challenge. *International Journal on Computer Vision 88*, 2 (2010), 303–338. 2

[FCNL13] FARABET C., COUPRIE C., NAJMAN L., LECUN Y.: Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell. 35*, 8 (2013), 1915–1929. 2

[Gra06] GRADY L.: Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. 28*, 11 (2006), 1768–1783. 2

[Hal] HALL M.: Family albums fade as the young put only themselves in picture. 1

[HSGL11] HACOHEN Y., SHECHTMAN E., GOLDMAN D. B., LISCHINSKI D.: Non-rigid dense correspondence with applications for image enhancement. *ACM Trans. Graph. 30*, 4 (2011), 70. 1

[JSD*14] JIA Y., SHELHAMER E., DONAHUE J., KARAYEV S., LONG J., GIRSHICK R., GUADARRAMA S., DARRELL T.: Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093* (2014). 5

[KK11] KRÄHENBÜHL P., KOLTUN V.: Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS* (2011), pp. 109–117. 2

[KWT88] KASS M., WITKIN A. P., TERZOPOULOS D.: Snakes: Active contour models. *International Journal on Computer Vision 1*, 4 (1988), 321–331. 2

[LCDL08] LEYVAND T., COHEN-OR D., DROR G., LISCHINSKI D.: Data-driven enhancement of facial attractiveness. *ACM Trans. Graph. 27*, 3 (2008). 1

[LCODL08] LEYVAND T., COHEN-OR D., DROR G., LISCHINSKI D.: Data-driven enhancement of facial attractiveness. *ACM Trans. Graph. 27*, 3 (2008). 3

[LLW08] LEVIN A., LISCHINSKI D., WEISS Y.: A closed-form solution to natural image matting. *IEEE Trans. Pattern Anal. Mach. Intell. 30*, 2 (2008), 228–242. 3

[LMB*14] LIN T., MAIRE M., BELONGIE S., HAYS J., PERONA P., RAMANAN D., DOLLÁR P., ZITNICK C. L.: Microsoft COCO: common objects in context. In *ECCV* (2014), pp. 740–755. 2

[LSD14] LONG J., SHELHAMER E., DARRELL T.: Fully convolutional networks for semantic segmentation. *CVPR* (2014). 2, 3, 6

[LSS09] LIU J., SUN J., SHUM H.: Paint selection. *ACM Trans. Graph. 28*, 3 (2009). 2

[LSTS04] LI Y., SUN J., TANG C., SHUM H.: Lazy snapping. *ACM Trans. Graph. 23*, 3 (2004), 303–308. 2, 8

[MB95] MORTENSEN E. N., BARRETT W. A.: Intelligent scissors for image composition. In *Proceedings of ACM SIGGRAPH* (1995), pp. 191–198. 2

[Mic] MICK J.: HTC: 90% of phone photos are selfies, we want to own the selfie market. 1

[MYS14] MOSTAJABI M., YADOLLAHPOUR P., SHAKHNAROVICH G.: Feedforward semantic segmentation with zoom-out features. In *CVPR* (2014). 2

[RKB04] ROTHER C., KOLMOGOROV V., BLAKE A.: "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. 23*, 3 (2004), 309–314. 2

[RRW*09] RHEMANN C., ROTHER C., WANG J., GELAUTZ M., KOHLI P., ROTT P.: A perceptually motivated online benchmark for image matting. In *CVPR* (2009), pp. 1826–1833. 3

[SJTS04] SUN J., JIA J., TANG C., SHUM H.: Poisson matting. *ACM Trans. Graph. 23*, 3 (2004), 315–321. 3

[SLC09] SARAGIH J. M., LUCEY S., COHN J. F.: Face alignment through subspace constrained mean-shifts. In *ICCV* (2009), pp. 1034–1041. 4, 6

[SPB*14] SHIH Y., PARIS S., BARNES C., FREEMAN W. T., DURAND F.: Style transfer for headshot portraits. *ACM Trans. Graph. 33*, 4 (2014), 148:1–148:14. 1, 3, 8, 9

[WC07] WANG J., COHEN M. F.: Image and video matting: A survey. *Foundations and Trends in Computer Graphics and Vision 3*, 2 (2007), 97–175. 3

[Win11] WINNEMÖLLER H.: Xdog: advanced image stylization with extended difference-of-gaussians. In *NPAR* (2011), pp. 147–156. 8, 9

[WSL12] WU J., SHEN X., LIU L.: Interactive two-scale color-to-gray. *The Visual Computer 28*, 6-8 (2012), 723–731. 8

[ZJR*15] ZHENG S., JAYASUMANA S., ROMERA-PAREDES B., VINEET V., SU Z., DU D., HUANG C., TORR P. H. S.: Conditional random fields as recurrent neural networks. *ICCV* (2015). 3