Exploring Image Matting: When GAN Meets U-Net

Amine El Hattami, Étienne Pierre-Doray, Youri Barsalou

Department of Computer Engineering and Software Engineering, Polytechnique Montréal
Email:{amine.elhattami, etienne.pierre-doray, youri.barsalou}@polymtl.ca

Abstract

Image matting has many applications in image editing and video production. We propose in this paper a deep learning architecture based on GAN and U-Net for image matting that estimates image foreground and outputs a realistic image. Our model generates a brand new image containing the foreground object to be extracted. In addition, we create a large-scale dataset including 6529 synthetic images for training and validation of our architecture.

1 Introduction

In this article, we tackle on the problem of background removal through image matting. It consists of predicting the foreground of an image or a video frame. However, unlike basic background / foreground segmentation, matting takes into account the transparency of an object. Indeed, objects seen on images are not always present at full opacity. Think for instance of a tinted glass box. Ideal image segmentation would give a mask telling which pixel belongs to the box and which to the rest of the image. However, ideal image matting would return a transparency mask for the box's coordinates, such that applying a mask to the box's original image and then onto a completely different background would allow us to see this new background through the box.

As presented in (Xu, Price, Cohen and Huang 2016), an image's matting can be formulated as :

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i \quad \alpha_i \in [0, 1]$$

where i indicates which pixel is concerned, α_i is the pixel's matte estimation, F_i is the foreground color, B_i is the background color and I_i is the pixel's color. The goal is therefore to estimate not only α_i, but also F_i and B_i.

This concept, once applied to color images, means we have to handle three color channels (for instance RGB color space) for I_i, F_i and B_i. This means the matting equation above aims to find seven unknowns through three known values. This is quite challenging to solve. Hence, additional information is commonly given in image matting situations, through what is known as a trimap. Such trimaps identify in white the sure foreground pixels, in black the sure background pixels and in gray the unsure values. Unsure values could for instance be the result of a partially transparent foreground, or the mixing of foreground and background colors on the foreground's borders.

Figure 1: Example of resulting background removal. From left to right : input image (generated by composition), associated input trimap and resulting extracted foreground

In this project, we made a script to generate a dataset of images and trimaps from results of Google Image searches for both transparent foreground images and some patterned background images. Image compositions between the two types of results have then been done in a similar way suggested by Xu et al. (2016). The generated dataset could be of virtually any size. However we restricted ourselves to a reasonable amount of images considering the limited amount of time and resources available to us.

We suggest continuing on the same path as Xu, Price, Cohen and Huang (2016) in order to explore deep learning solutions to image matting. We chose to do this by using U-Nets after their success in the Carvana Image Masking Challenge (Iglovikov and Shvets 2018). The use of this type of convolutional network allows us to identify both local features and global features in the image and use these in addition to the pixel's color in order to estimate said pixel's matte. We learn this neural network end-to-end by giving it both the image and its associated trimap for the object we aim to estimate the matting for.

Following the results of Iizuka, Simo-Serra and Ishikawa (2017) at their adaptation of a Generative Adversarial Networks (GAN)(Goodfellow et al. 2014) to image

reconstruction, we add a discriminator network tasked to identify if a matting mask was generated by the U-Net. This allowed us to generate more realistic matting masks. We then tried to add the application of the matte on the image inside the network and therefore obtained a GAN returning a complete image of the extracted foreground object.

We perform experiments on our dataset in order to evaluate our suggested solution. Preliminary results seem to indicate that our method can successfully generate the alpha matte mask. However the image's RGB predictions seem to be a bit off target.

1.1 Related Works

At first, before choosing to work with trimaps, we were looking into a way to identify a foreground object in an image. We therefore looked into using the contour detection technique suggested in Yang, Price, Cohen, Lee, and Yang (2016). They suggest using a fully convolutional encoder-decoder in order to detect the higher level contours of an image. This could therefore have been used in our project in order to be able to calculate the image matting on any object detected by the contour detection. However, it would require training another network, which we lacked time and resources for. Nevertheless, in future works it could be interesting to include this into the project in order to make an interactive user interface allowing to choose what object to infer the matting mask for.

The main idea behind our project came from the deep learning approach taken for image matting in Xu, Price, Cohen and Huang (2017). They present a model for image matting that makes matting mask predictions using an encoder-decoder network, which they connect to a refinement network that improves the alpha matting mask. They give the network both the image and its associated trimap. They also generate a dataset by extracting foregrounds using Photoshop, making an alpha matte for it and composing the resulting on new backgrounds. They conclude that such a composite dataset doesn't hurt the trained network's performance to evaluate on natural images, which justify why we chose to similarly create a dataset from composite images.

Other techniques have been suggested in order to perform image matting. Chuang, Curless, Salesin and Szelski (2001) use Bayesian distributions to represent both foreground and background and then estimate the matte according to a maximum likelihood criterion. Sun, Jia, Tang and Shum (2004) represent the matting problem in a way that the matte is obtained through the resolution of a Poisson equation. Levin, Lischinski and Weiss (2007) derive cost functions for both foreground and background, combine them into a quadratic cost function giving the alpha, and then solve the linear system of equations to obtain the matte. He, Sun, and Tang (2012) develop a new image filter called the guided filter, and test its applications in computer graphics. One of the experiments builds upon the closed form solution for matting (Levin, Lischinski and Weiss 2007) to apply the guided filter and shows promising results. None of these four popular matting techniques go in a similar direction as what is explored in Xu, Price, Cohen and Huang (2017), which we aim to explore further.

Our work is done using a U-Net architecture. This network was suggested by Ronneberger, Fischer and Brox (2015). They perform image segmentation using a U shaped fully convolutional architecture. The down slope of the U is composed of regular convolutions and max-pooling layers, while the upward slope is made of up sampling and convolutions. Information from the down slope convolutions are sent forward to some layers of the up slope in order to pass along more global information to the up-sampling steps.

Recently, U-Nets are still showing good performance in various applications. An example of this was made by Iglovikov and Shvets (2018). They illustrate that initializing the weights on a U-Net (especially its encoder weights) with those of a pre-trained network can improve its performance. One of their experiments was to apply their pre-trained network on the Carvana Image Masking Challenge, which they won with their U-Net architecture based on a VGG11. We noticed that the task of the Carvana dataset isn't that far from the task of image matting : both are related to background removal. We therefore made the hypothesis that applying a U-Net to the image matting problem could give good results.

One could say that a fully convolutional network applied to a task such as image matting is a generative model, as it generates a new image of its own. Such generative models can be improved to reach greater performance by using a concept known as adversarial training. In particular, the Generative Adversarial Nets suggested by GoodFellow et al. (2014), seem to yield interesting results. Such GAN use another model known as the discriminative model that tries to identify which inputs come from the generative model and which are real inputs. The generative model's task therefore becomes to fool the discriminative model. This "competition" between both models leads to increasing performance in both. A good example of how such networks could be used is shown by Iizuka, Simo-Serra and Ishikawa (2017). They adapted a GAN to add a second discriminative model, one for local information and one for global information. That way they were able to train a generative model to fill holes in image realistically based on both local and global information.

2 Method

Our approach is based on generative adversarial networks (GAN) applied for the image matting task. We employ a U-Net architecture as a generative network to estimate image foreground and a deep convolutional network is used as a discriminator during training.

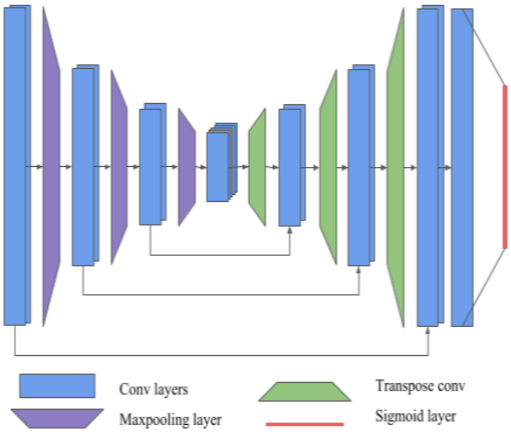

Figure 2: Architecture used for the U-Net acting as a generative model, and its legend.

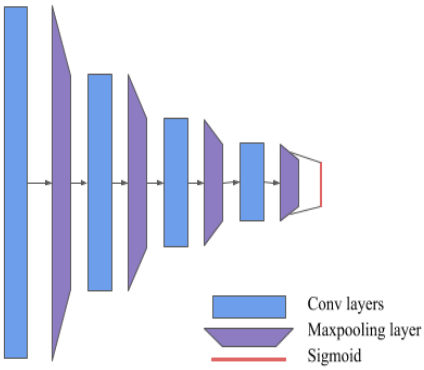

Figure 3 : Architecture used for the ConvNet acting as the discriminative model, and its legend

2.1 Generative Network

The U-Net architecture has gained popularity for various applications, including segmentation problems. We propose a deep convolutional architecture for the matting network that reuses the idea of concatenating lower level representations that are sent forward through layers of the network in order to improve accuracy and to better propagate local information through the network. The proposed network takes 4 channels as input, RGB and the trimap as a 4th channel, and outputs a 4 channels (RGBA) image that includes opacity to represent the foreground object. Layers of the network are detailed in Table 1 and illustrated on Figure 2.

2.2 Discriminator Network

The discriminator is a deep convolutional network terminated with a fully connected layer to output a single value. The goal of the discriminator is to tell if the input image containing the foreground object is the ground truth or was synthesized using the matting generative network. Layers of the network are detailed in Table 2 and illustrated in Figure 3.

2.3 Training

Let E(x, M_T) denote the matting network in a functional form, with x the input image and M_T the trimap that is the same size as the input image. Similarly, D(x) denotes the discriminator in a functional form.

Following previous work on Generative Adversarial Network (GAN) (Iizuka et al. 2017), two loss functions are jointly used: a weighted Mean Squared Error (MSE) loss for training stability, and a Generative Adversarial Network (GAN) loss to improve the realism of the results.

The generative network is trained alone for T_m iterations with a MSE loss defined as:

$$L(x, M_T) = \|E(x, M_T)\|^2$$

The MSE loss can be interpreted as the foreground reconstruction error.

Table 1. Architecture of the image matting network. The network is a U-Net with skip links that goes from the output of each max pooling layer and concatenated with the output of the corresponding deconvolution layer.

Type	Kernel	Stride	Activation	Outputs
Double conv.	3x3	1x1	ReLu	16
Double conv.	3x3	2x2	ReLu	32
Double conv.	3x3	2x2	ReLu	64
Double conv.	3x3	2x2	ReLu	128
conv.	3x3	1x1	ReLu	128
conv.	3x3	1x1	ReLu	128
conv.	3x3	1x1	ReLu	128
Deconv.	3x3	1/2x1/2	ReLu	64
Double conv.	3x3	1x1	ReLu	64
Deconv.	3x3	1/2x1/2	ReLu	32
Double conv.	3x3	1x1	ReLu	32
Deconv.	3x3	1/2x1/2	ReLu	16
Double conv.	3x3	1x1	ReLu	16
conv.	1x1	1x1	Sigmoid	4

Table 2. Architectures of the discriminators used in our network model.

Type	Kernel Size	Stride	Activation	Outputs
conv.	3x3	2x2	ReLu	16
conv.	3x3	2x2	ReLu	32
conv.	3x3	2x2	ReLu	64
conv.	3x3	2x2	ReLu	128
FC	-	-	Sigmoid	1

The discriminator outputs the confidence of an image being authentic or else, the outcome of the matting operation. The GAN loss is the cross-entropy that compares the confidence for both images. As a result, training both the generative and discriminator networks becomes a min-max optimization.

$$\min_E \max_D \mathbb{E}[\log(D(x, M_T)) + \log(1 - D(E(x), M_T)]$$

We combine both loss functions to obtain:

$$\min_E \max_D \mathbb{E}[L(x, M_T) + \alpha \log(D(x)) + \alpha \log(1 - D(E(x, M_T))]$$

In practice, the discriminator network is trained alone for T_D iterations and the generative network is trained again with the combined loss function for T_{train} iterations.

In optimization, we use the ADADELTA algorithm (Zeiler 2012), which sets a learning rate for each weight in the network automatically.

3 Experiments

We first experimented with a PyTorch implementation of a U-Net neural network on the Carvana Image Masking Challenge dataset, which contains a large number of high-resolution car images (1918×1280). Each car has multiple images taken at different angles. The goal was to be able to filter the studio background by predicting a mask that can be applied to the original image. The first issue we faced was the large computing power requirement needed during training, due to the large size of the input images which required more than 17 Gb of GPU memory. A workaround for this issue was to either to reduce the size of the input images or the depth of U-Net. The second issue was related to dataset images. Since all images of the challenge have the same background the results of the network on the validation set were good, but it performed poorly on images with a slight change in the background as we can see on figures 4 and 5.

Figure 4:Result on validation set. From left to right: the input image and the resulting mask.

Figure 5: Result on a random car image. From left to right: the input image and the resulting mask.

We evaluated the approach presented in this article using a custom dataset. The dataset was created by crawling Google Image for foreground and background images separately to create 960x720 images. For the foreground images, "portrait transparent background" was used as a search criteria, with an additional filter to select RGBA images. The resulting images were then manually filtered to select the ones with proper contours, to obtain a total of 329 foreground images. As for the background images, "textured background" was used as a search criteria. Then each foreground image was combined with 20 randomly selected background images using the image matting formula :

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i \quad \alpha_i \in [0, 1]$$

Finally, we generated a trimap for each resulting image. The inner contour was obtained by applying a small threshold on the opacity of the image and then eroding the resulting contour. The outer contour was obtained by applying a large threshold and then dilating the contour. Both the inner contour and outer contour were combined in a single image, coloring with grey pixels in between, to obtain the trimap.

The model presented in this article was implemented using TensorFlow and can be found at https://github.com/eti-p-doray/unet-gan-matting. We first experimented with a model that did not include a GAN. But in cases where the background color was close to the one of the foreground, the model was not able to filter the background properly and also the transition from foreground to background was not seamless. The extracted foregrounds obtained through that version of the model had luminance values that differed from their original versions, as could be noted in the 3rd column of Figure 6.

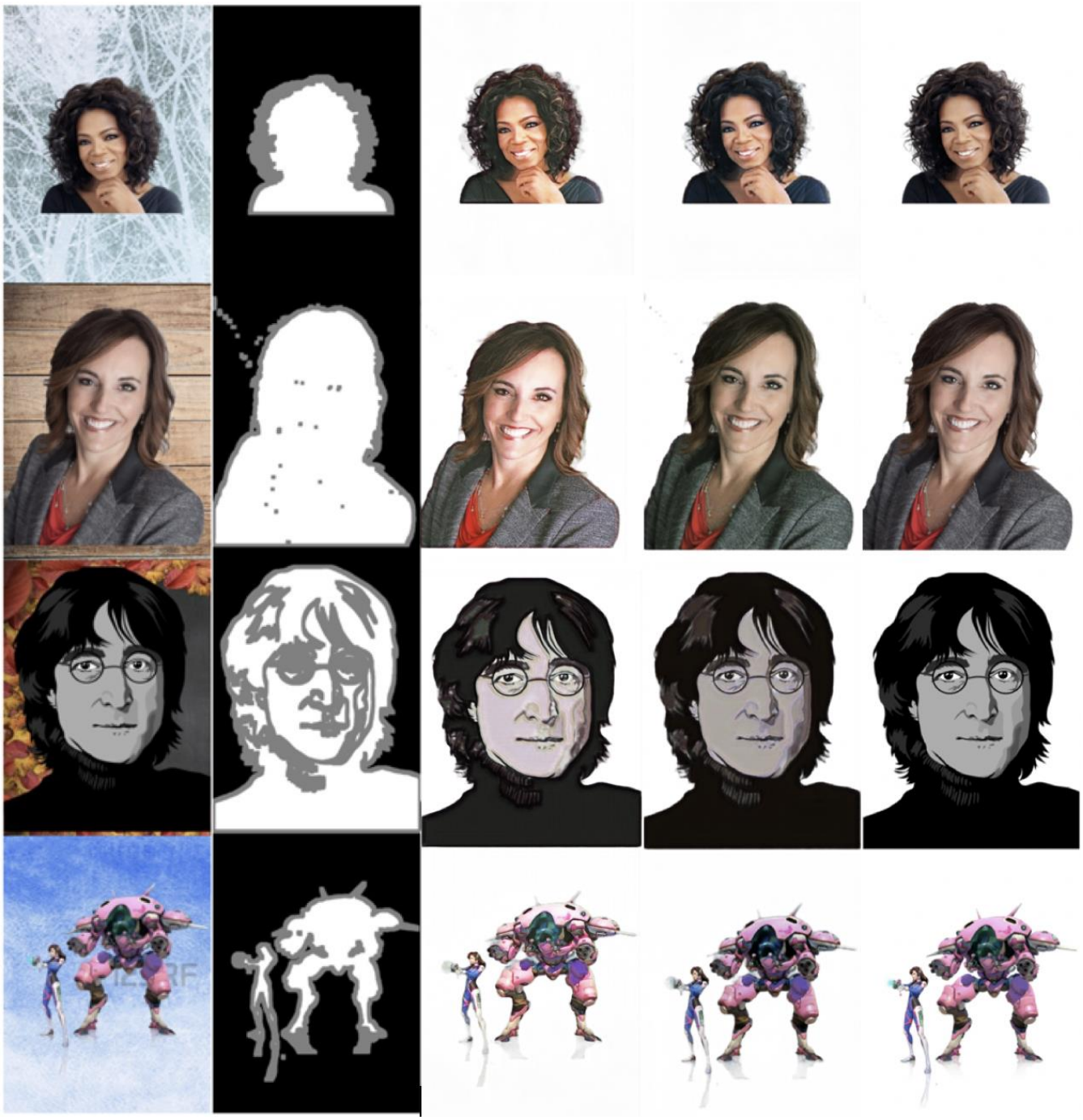

Figure 6: Final results. From left to right: the input image, the associated input trimap, resulting extracted foreground using only a U-Net in (a), resulting extracted foreground using a GAN in (b) and the ground truth. One can notice the variation in luminance in (a) and the variation in color hue in (b) when compared to the ground truth

Then we added a GAN to our model in order to add a more realistic and authentic feel to the output images as shown in Figure 4. We also experimented with input images in which our first U-Net implementation performed poorly. We observed better results despite the discoloration and even if our training dataset did not include car images, which can be seen by comparing the results from Figure 5 and Figure 7.

4 Discussion

We have yet to evaluate our method on any benchmarks, and must therefore limit ourselves to a qualitative evaluation of the resulting images. Our method seems to produce images with pretty accurate alpha channels values. This means that the image matting is generally well estimated. However, as it can be observed on Figure 6, the RGB values of the resulting image differ from the original image. On that figure, the difference is not that drastic, but is still easily noticeable when compared with the original image. We can attribute this color change to the use of a GAN. Indeed, when we look at the output obtained without a GAN, the color hue seems to be more accurate than what is obtained with a GAN. The use of a GAN is more accurate when it comes to luminance values and the precision of the matte in specific situations (in most situation both perform equally). We therefore think the application of a GAN to this model is an improvement.

A more drastic error in the resulting RGB channel can be seen in Figure 7. That last figure was made using a foreground image that wasn't in the generated dataset, and we can see that the model had trouble handling it correctly. This hints towards potential overfitting to our dataset. This could be explained by the small number of foreground objects in the training set (329).

In Figure 6, it is interesting to notice that if the output of the image and the ground truth are presented side by side without label, it is often hard for the human eye to judge which is real, despite the changes in coloration. This is most likely the result of the application of a GAN to our model.

5 Conclusion

U-Nets and Generative Adversarial Networks deliver promising results in the field of image matting. When used to generate the foreground F instead of the matte mask α, there are some coloration issues, but this stays in the realm of realistic colors. We do seem to have some overfitting issues due to the limited number of foreground objects in the training set we generated. We think this could be partly solved by exploring data augmentation. Future works could use the architecture to only generated alpha matte and try to refine it as it was done in Xu, Price, Cohen and Huang (2017), or could try to adjust the coloring by training over a bigger dataset, possibly made of real images.

Figure 7: Example of a drastic coloration error. In order, foreground, foreground with background and resulting image.

References

Carvana. 2017. Carvana Image Masking Challenge. Kaggle: https://www.kaggle.com/c/carvana-image-masking-challenge.

Chuang., Y.-Y., Curless, B., Salesin, D. H., and Szeliski, R. 2001. A Bayesian Approach to Digital Matting. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2001*, 264-271. Piscataway, NJ, USA: IEEE Press.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. 2014. Generative Adversarial Networks. In *Proceedings of the 27th Advances in Neural Information Processing Systems (NIPS 2014), 2672–2680.*

He, K., Sun, J., and Tang, X. 2012. Guided Image Filtering. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(6): 1397-1409.

Iglovikov, V., and Shvets, A. 2018. TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation. arXiv:1801.05746.

Iizuka, S., Simo-Serra, E., and Ishikawa, H. 2017. Globally and Locally Consistent Image Completion. In *Proceedings of the 44th ACM Transactions on Graphics (SIGGRAPH 2017)*, article no. 107. New York, NY, USA: ACM Press.

Levin, A., Lischinski, D., and Weiss, Y. 2007. A Closed-Form Solution to Natural Image Matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(2): 228-242.

Ronneberger, O., Fischer, P., and Brox. T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, eds. Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., 234-241. Switzerland: Springer, Cham.

Sun, J., Jia, J., Tang, C.-K., and Shum, H.-Y. 2004. Poisson Matting. In *Proceedings of the 31st ACM Transactions on Graphics (SIGGRAPH 2004)*, 315-321. New York, NY, USA: ACM Press.

Yang, J., Price, B., Cohen, S., Lee, H., and Yang, M. 2016. Object Contour Detection with a Fully Convolutional Encoder-Decoder Network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016*, 193-202. Piscataway, NJ, USA: IEEE Press.

Xu, N., Price, B., Cohen, S., and Huang, T. 2017. Deep Image Matting. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*, 311-320. Piscataway, NJ, USA: IEEE Press.

Zeiler, M. D. 2012. ADADELTA: An Adaptive Learning Rate Method. arXiv:1212.5701.