Removing Background from Portrait Images using U2Net trained with Cyclical Learning

Gurjot Singh **Rajdeep Dutta**

Abstract

This project presents a novel approach to remove the background from portrait images using the U^2-Net model. It is trained using cyclical learning and generates the alpha channel transparency mask of the image. We use a custom root mean squared error-based loss function that allows the model to focus primarily on the edges of the human portrait, which is critical since those are the areas where the foreground (in our case, the human) slowly changes into the background. Since we treat this task as image Matting and predict continuous pixel values instead of discrete labels, our approach remains agnostic to the user input image resolution. The proposed method is trained and evaluated in the EasyPortrait dataset. The generated results show that our approach does indeed work and can extract the human from an image, with fine boundaries and proper transparency mask, especially near the hair and clothing.

The source code and pre-trained models are available on GitHub here: https://github.com/rdutta1999/bg-removal-human-portrait

1. Introduction and Background

Background removal in portrait images stands as a pivotal challenge in diverse creative fields, from photography to digital media. The need to precisely extricate subjects from their backgrounds is fundamental for creating visually compelling and adaptable content. Traditional methods, like manual selection or chroma keying (Raditya et al., 2021), often prove inadequate when confronted with the intricacies of real-world images.

Our work's primary goal is to harness the capabilities of the U^2-Net (Qin et al., 2020) model for background removal in human portrait images. Specifically, we aim to augment U^2-Net's proficiency by integrating Cyclical Learning (Smith, 2017) (Smith, 2022) (Smith & Topin, 2018) into the training process. Cyclical Learning, with its ability to enhance model generalization and prevent convergence to sub-optimal solutions, becomes a key component in our pursuit of superior performance compared to traditional training methods.

Moreover, we tackle this problem similar to image matting (Li et al., 2023) instead of image segmentation (Minaee et al., 2020). This is because image segmentation is, inherently, a classification task where we predict discrete labels for every individual pixel. In image matting, however, we predict the alpha values for each pixel which can be a continuous set of values within a particular range, e.g. $0 - 1$ or $0 - 255$.

The authors of the U^2-Net architecture use an Adam optimizer with an initial learning rate set to 10^{-3}. In our case, we plan to oscillate the learning rate cyclically between a base learning rate and a maximum learning rate, inspired by (Smith, 2017) (Smith, 2022) (Smith & Topin, 2018).

We also diverted from the loss function used by the authors of U^2-Net architecture. Focusing on image segmentation, they used cross entropy (Mao et al., 2023) loss in all the side output saliency probability maps and the final fused saliency probability map after upsampling. We, however, focussing on image matting, used the root mean squared error (Hodson, 2022) loss function for all the side and fused saliency maps. Additionally, we added a weighted edge-based loss, inspired by (Ke et al., 2022), to force the model to focus more on the transition region between the foreground and background.

The benefit of using root mean squared error instead of cross entropy is that it allows us to nicely deal with the interpolation of the pixels that take place when images are resized. Additionally, this helps our approach to be agnostic to the resolution of the original input images. For example, let us consider that our model is trained on batches of 320×320 resolution images. This resolution is a good middle point to have an optimal number of images in a single batch for proper gradient descent (J. Zhang, 2019). However, this means that all the images and their corresponding ground truth mask in the dataset would need to be resized to this common 320×320 resolution. Suppose, the original mask had two unique values - 1 for denoting the foreground (in our case, the human) and 0 for denoting the background. After resizing, the mask will now have interpolated values between 0 and 1 and not just the binary values - 0 and 1. If

we apply a threshold on this resized mask, it will result in hard boundaries and sharp edges similar to a step function, since any intermediate values will be rounded off to 0 or 1. Training on such masks will result in predictions with similar hard edges when using cross entropy since the loss function pushes the probabilities to the minimum (i.e. 0) and maximum (i.e. 1) values. It should be noted that this can be somewhat tackled using focal loss (Lin et al., 2018), but that alone is not sufficient. Further, resizing them from 320×320 to the original high resolution of the image (and applying a threshold again over the resized mask) would result in these rough edges being further enlarged and pronounced, leading to unacceptable results.

A better idea is to use regression-based losses such as minimum absolute error (Qi et al., 2020) or root mean squared error since these loss functions will then predict continuous values within the $0 - 255$ range. That way, when we resize the predicted mask from 320×320 to 1920×1080, the gradual transition at the edges of the mask will be maintained since during resizing, the interpolation will also result in a continuous set of values. This will lead to a nice smooth alpha transparency mask.

2. Related Work

The task of removing backgrounds from portrait images has gained significant attention in computer vision and image processing. Various approaches have been proposed to address this challenging problem, with a focus on achieving high-quality results and real-time performance. In this section, we review the related work in the context of background removal, emphasizing the advancements in deep learning techniques and the use of U^2-Net trained with cyclical learning.

Early attempts to remove backgrounds from images often relied on handcrafted features and conventional image processing techniques. Methods such as color-based segmentation, edge detection, entropy filtering (C.-C. Cheng, 2021) and region growing were prevalent. While these approaches showed some success, they struggled with complex scenes, varied lighting conditions, and intricate object boundaries.

Traditional techniques like chroma keying, which involves replacing a specific color in an image with a different background, have been widely used in the film industry. However, their effectiveness is contingent upon a uniform background color, limiting their applicability to portrait images with diverse backgrounds.

Deep learning-based approaches, using model architectures such as U-Net (Ronneberger et al., 2015) and Mask R-CNN (He et al., 2018), have shown promise in segmenting out the salient object in an image. The authors of (H. K. Cheng et al., 2020) present a novel approach that refines a low-resolution segmentation to high resolution. Their approach is plugged after an existing segmentation model in the pipeline and takes as input the existing model's output. Using a global step and a series of local steps, the input segmentation map is refined to a higher resolution.

In (Ke et al., 2022), the authors have used a multi-branch architecture to simultaneously optimize three objectives - semantic estimation, detail prediction, and semantic-detail fusion. This allows them to extract the salient object in the foreground without using a trimap (Gupta & Raman, 2017). Taking inspiration from their detail prediction optimization step, we also implement a transition region-based loss that helps the model preserve details near the edges or boundaries of the salient object, in our case, the human.

3. Methodology

We started with the original pre-trained U^2-Net model and fine-tuned it on the EasyPortrait dataset. For the learning rate scheduler, we used Cyclical Learning to oscillate the learning rate between two bounds - the base learning rate and the maximum learning rate.

At the end of the training, for the last few epochs, we 'annihilate' the learning rate i.e. push the learning rate much further down compared to the base learning rate. For the loss function, we used root mean squared error between the saliency probability maps and the resized ground truth mask. Moreover, we added a weighted term to focus more on the transition region between the foreground and background.

3.1. Data Selection and Preprocessing

We used the EasyPortrait dataset. There are $40,000$ images, out of which approximately $38.3K$ images have a resolution of 1920×1080 or higher. The dataset contains images of $13,705$ unique persons and takes up a size of 91.78 GB of space.

The dataset contains masks with discrete pixel values corresponding to each part of the human face. The pixel value and class mapping have been shown in Table 1.

For our task, however, we only need to extract the human face from the image. Thus, we have no need for individual masks for each part of the human face. Instead, we apply a simple pre-processing step to extract all non-zero pixels and label them as 1 to denote the foreground. Thus, all the pixels that represent these classes - *PERSON*, *SKIN*, *LEFT_BROW*, *RIGHT_BROW*, *LEFT_EYE*, *RIGHT_EYE*, *LIPS*, and *TEETH* are simply labeled as 1.

The U^2-Net model was trained on **DUTS-TR**, which is a part of the DUTS dataset (Wang et al., 2017). As a result, the input images were normalized using the channel-wise mean and standard deviation of the ImageNet (Deng et al.,

Figure 1. Image Segmentation

Figure 2. Image Matting

Index	Class
0	BACKGROUND
1	PERSON
2	SKIN
3	LEFT_BROW
4	RIGHT_BROW
5	LEFT_EYE
6	RIGHT_EYE
7	LIPS
8	TEETH

Table 1. Pixel value VS class mapping

Figure 3. Human portrait image on the left, and the output image after background removal on the right

2009) dataset. For the red channel, the mean is 0.485 and standard deviation is 0.229. For the green channel, the mean is 0.456, and the standard deviation is 0.224. Finally, for the blue channel, the mean is 0.406 and the standard deviation is 0.225. Since we are using the pre-trained U^2-Net model, we also applied the same normalization technique.

We applied several data augmentation (Yang et al., 2023) techniques to increase the variety of the images seen by the model. Broadly speaking, three types of augmentation have been applied - cropping, geometric transformation, and visual transformation. For cropping, we randomly cropped a subset of the image and used padding if necessary. For geometric transformations, we applied horizontal flipping, And finally, visual transformations contain the majority of augmentation techniques. We randomly applied channel shuffling, shifting the RGB channels, and modifying the hue, contrast, and saturation values. We also changed the brightness and contrast of the images, used noise filters such as Gaussian noise and ISO noise, randomly changed the

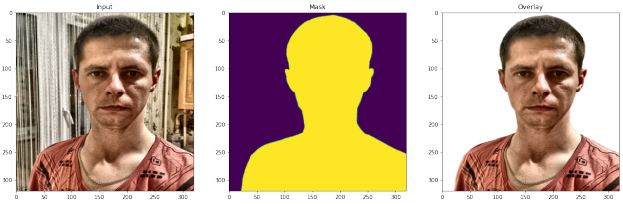

Figure 4. Left: Training image, Middle: Ground truth mask, Right: Mask overlaid on image

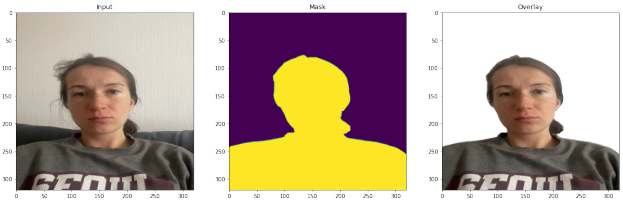

Figure 5. Left: Validation image, Middle: Ground truth mask, Right: Mask overlaid on image

gamma, and used histogram equalization techniques such as CLAHE (Mishra, 2021). Further, we applied multiple blur filters at random such as Gaussian blur, glass blur, median blur, and motion blur. Finally, we transformed the image and mask arrays to the (C, H, W) format that PyTorch expects, where C,H and W refers to the channel dimension, height, and width of the image respectively.

After the augmentation, each image and mask pair has been resized to 320×320. Again, it is to be noted that, keeping in line with the idea of image matting, we did not apply thresholding to the resized mask to convert it into a binary mask. Resizing the mask causes pixels near the border to be interpolated and achieve continuous values between 0 and 1 (where 0 denotes the background and 1 denotes the foreground). We maintained these continuous values and considered our task to be predicting continuous values between 0 and 1 for each pixel rather than a $0/1$ binary value. In short, we treat this as a pixel-wise regression task (image matting) rather than pixel-wise classification (image segmentation). In short, we aim to predict the alpha-channel transparency mask for the salient object in the image, in our case, the human itself.

3.2. Model Selection and Transfer Learning

We start with a pre-trained U^2-Net model (Fig. 8) and fine-tune it on the EasyPortrait dataset using the data preprocessing techniques mentioned before. The model has Residual U-blocks (Fig. 9), similar to residual blocks in ResNets (He et al., 2016), which allow it to capture intra-stage multi-scale features. We continue to use the sigmoid activation function for the fused saliency output and the intermediate saliency output maps. As a result, our predictions are continuous pixel values between 0 and 1.

3.3. Loss Function

Since we are focused on image matting and predicting continuous values for the pixels in the image, we use root mean squared error instead of cross entropy. Thus, our loss function, initially, looks like this:-

$$L = \sum_{m=1}^{M} w_{side}^{(m)} l_{side}^{(m)} + w_{fuse} l_{fuse}$$

where $l_{side}^{(m)}$ ($M = 6$, as the Sup1, Sup2, $\cdots$, Sup6 in Fig. 8) is the loss of the side output saliency map $S_{side}^{(m)}$ and l_{fuse} (Sup7 in Fig. 8) is the loss of the final fusion output saliency map S_{fuse}. $w_{side}^{(m)}$ and w_{fuse} are the weights of each loss term.

For each term l. we calculate the standard root mean squared error to calculate the loss:-

$$l_{side/fuse} = \sqrt{\frac{1}{HxW} \sum_{(r,c)}^{(H,W)} (P_{G(r,c)} - P_{S(r,c)})^2}$$

where (r, c) is the pixel coordinates and (H, W) is image size: height and width. $P_{G(r,c)}$ and $P_{S(r,c)}$ denote the pixel probability values of the ground truth and the predicted saliency probability map, respectively. The training process tries to minimize the overall loss L. In the testing process, we choose the fusion output l_{fuse} as our final saliency map.

Additionally, we generate a mask for the transition region - the region between the foreground and the background. Using a weight parameter λ, we control the amount of attention the model gives to the edge regions of the mask during learning. Thus, our loss function gets modified and its final form is as follows:-

$$L_{final} = \sum_{m=1}^{M} w_{side}^{(m)} l_{side}^{(m)} + w_{fuse} l_{fuse} + \lambda l_{transition}$$

The loss in the transition region, $l_{transition}$ is defined as:-

$$l_{transition} = \sqrt{\frac{1}{HxW} \sum_{(r,c)}^{(H,W)} m_d * (P_{G(r,c)} - P_{S(r,c)})^2}$$

where (r, c), (H, W), $P_{G(r,c)}$ and $P_{S(r,c)}$ are defined as above. m_d is the mask of the transition region as shown

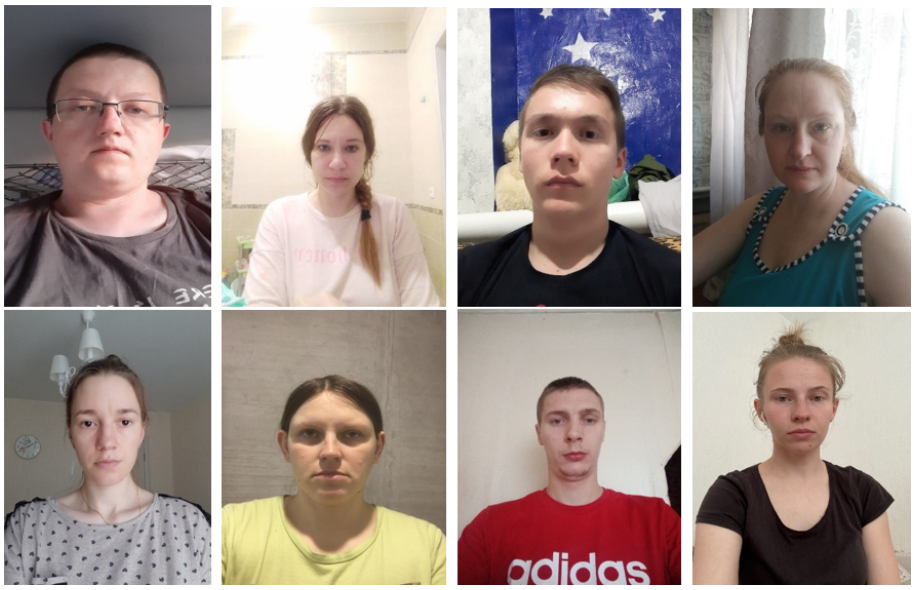

Figure 6. Some images from the EasyPotrait dataset

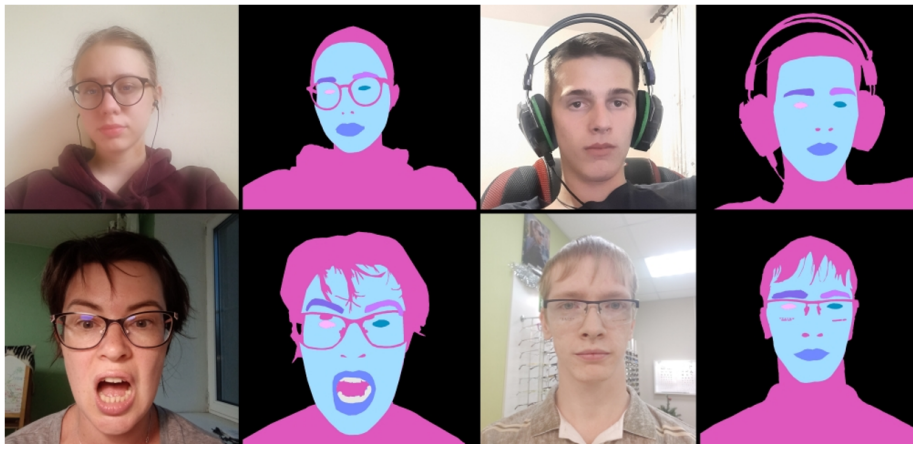

Figure 7. Each part of the face are mapped to different pixels for the task of segmentation

in Fig. 6. m_d is generated through dilation and erosion operations on $P_{G(r,c)}$. Its values are 1 if the pixels are inside the transition region, and 0 otherwise. $*$ denotes element-wise multiplication.

In our implementation, we set the loss weights $w_{side}^{(m)}$ and w_{fuse} to 1. We also set the weight parameter for the edge loss λ to 1.

3.4. Training Procedure

Before training, we defined the U^2-Net architecture and loaded the pre-trained weights (from the **DUTS-TR** dataset). We used the Adam optimizer with a learning rate of 10^{-3}. The coefficients used for computing running averages of gradient and its square i.e. the betas argument were set to its default value of $(0.9, 0.999)$. We used a weight decay of 0. For the learning rate scheduler, we used OneCycleLR to implement cyclical learning. The maximum learning rate was set to the same value of 10^{-3}.

The loss function is our custom boundary-based loss that we have defined previously. We trained for 15 epochs and saved checkpoints every 3 epochs. The checkpoints contain the current epoch number and the state dictionaries of the model, the optimizer, and the learning rate scheduler.

We trained the model on an NVIDIA RTX 2060 Super and AMD Ryzen 3600. Each epoch took around 2 hours 35 mins.

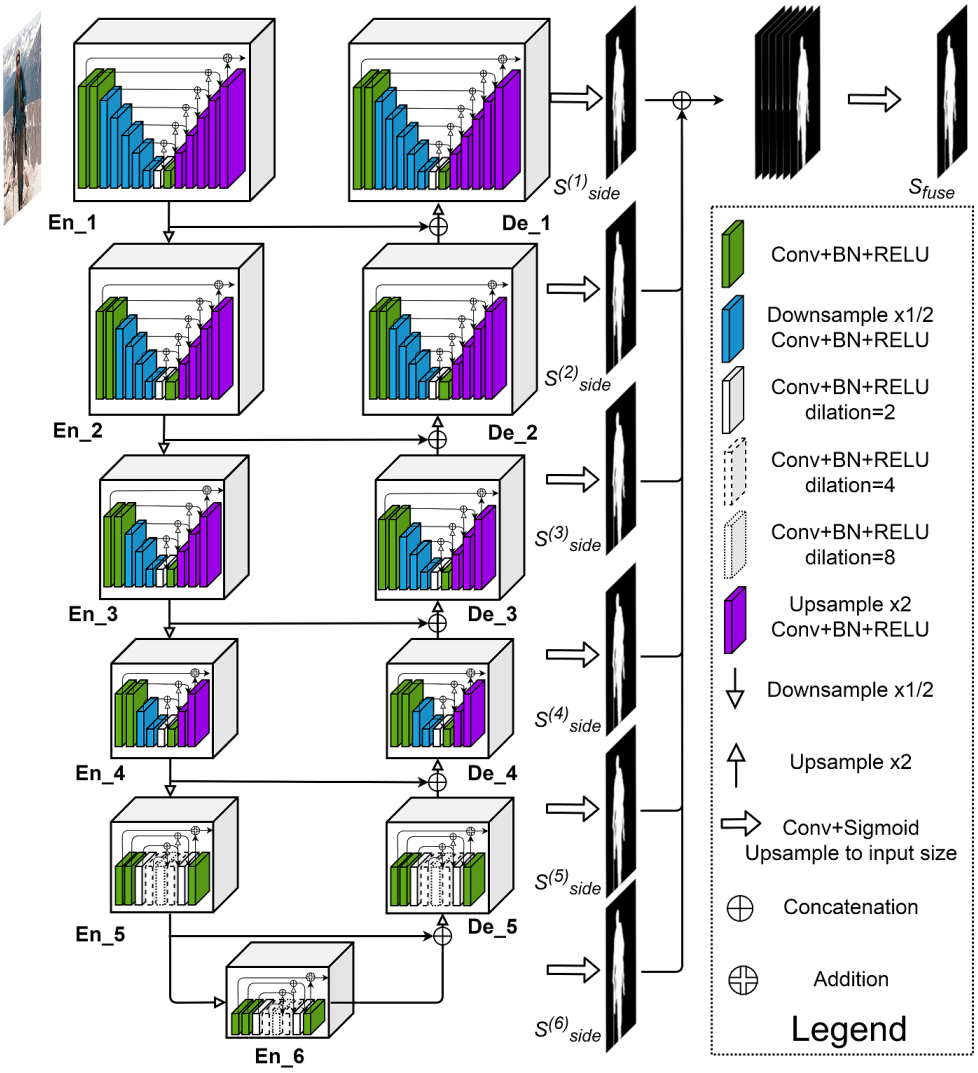

Figure 8. U^2-Net model architecture

4. Experiment and Results

Fig. 10 and 11 shows the training loss curve and validation loss curve against the number of iterations. Since we started from a pre-trained model, the initial loss was not that much high and as we fine-tuned on the EasyPortrait dataset, the loss kept decreasing. The training loss is comparatively higher than the validation loss. This is because, we applied a series of complex augmentations to the training images, which made it difficult for the model to properly predict the alpha transparency mask of the image. We did not apply any augmentations to the validation image, other than just transforming the image to (C, H, W) format from (H, W, C) as described previously.

Fig. 12 shows the results of our trained model. As you can see, it can deal with busy backgrounds and can remove them from a variety of human portrait images. It is not perfect, however. For example, it faces trouble in properly separating the thin edges such as the hair. We hope to tackle this by assigning a higher value to the λ parameter and tweaking the loss function to better focus on the edges of the mask.

5. Discussion and Conclusion

Our proposed approach is able to remove a wide variety of backgrounds and properly extract the human portrait, as shown in Figure 10. There are some artifacts especially, near the contours of the human body, where the background is dark and the model is unable to easily separate the foreground from the background. We expect to tackle this by increasing the weightage of the edge-based loss i.e. the λ parameter in the loss function, as defined above. This will

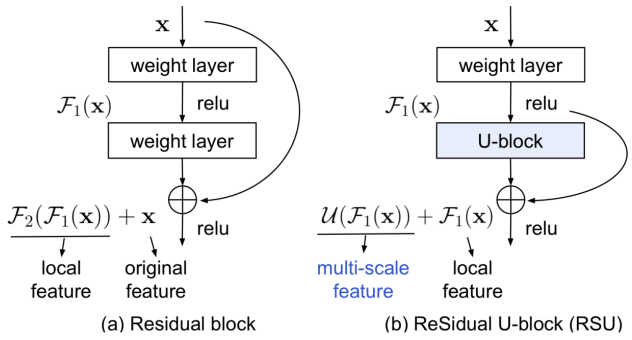

Figure 9. Residual block in ResNet vs Residual U-Block in U^2-Net

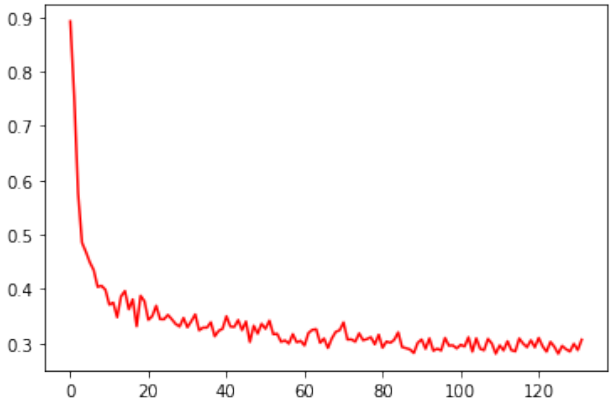

Figure 10. Training loss values every 200 iterations

force the model to give additional attention to the edges where the foreground slowly blends with the background.

We also hope to explore other recent architectures such as K-Net (W. Zhang et al., 2021). Such heavy models may have a high turnaround time, but it may be compensated by the increase in prediction quality. Additionally, we plan to use quantization-aware training in the future, such as the Automatic Mixed Precision package in PyTorch (PyTorch, n.d.). There are multiple benefits - faster training for the same input size and a possible increase in the batch size during the training process. Moreover, since using mixed precision lowers the VRAM requirements, we may be able to increase the training size to (720×720) and generate better results.

Another possible scope of improvement is to use a learning rate finder (Lightning, n.d.) to find the optimal maximum learning rate for the cyclical learning scheduler. This will enable faster convergence and allow us to find the global minima of the optimization function.

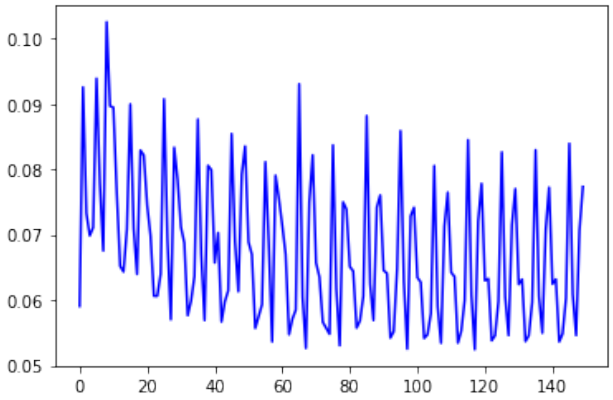

Figure 11. Validation loss values every 200 iterations

References

Cheng, C.-C. (2021). Single-image background removal with entropy filtering. In *Proceedings of the 16th international joint conference on computer vision, imaging and computer graphics theory and applications (visigrapp 2021) - volume 4: Visapp* (p. 431-438). SciTePress. doi: 10.5220/0010301204310438

Cheng, H. K., Chung, J., Tai, Y.-W., & Tang, C.-K. (2020). *Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement.*

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 ieee conference on computer vision and pattern recognition* (p. 248-255). doi: 10.1109/CVPR.2009.5206848

Gupta, V., & Raman, S. (2017). *Automatic trimap generation for image matting.*

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2018). *Mask r-cnn.*

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 ieee conference on computer vision and pattern recognition (cvpr)* (p. 770-778). doi: 10.1109/CVPR.2016.90

Hodson, T. O. (2022). Root-mean-square error (rmse) or mean absolute error (mae): when to use them or not. *Geoscientific Model Development*, *15*(14), 5481–5487. Retrieved from https://gmd.copernicus.org/articles/15/5481/2022/ doi: 10.5194/gmd-15-5481-2022

Figure 12. Top: Test images of human portraits, Bottom: Our model's predictions

Ke, Z., Sun, J., Li, K., Yan, Q., & Lau, R. W. H. (2022). *Modnet: Real-time trimap-free portrait matting via objective decomposition.*

Li, J., Zhang, J., & Tao, D. (2023). *Deep image matting: A comprehensive survey.*

Lightning, P. (n.d.). *Learning rate finder — pytorch lightning 1.4.9 documentation.* https://pytorch-lightning.readthedocs.io/en/1.4.9/advanced/lr_finder.html. (Accessed 22-12-2023)

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018). *Focal loss for dense object detection.*

Mao, A., Mohri, M., & Zhong, Y. (2023). *Cross-entropy loss functions: Theoretical analysis and applications.*

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2020). *Image segmentation using deep learning: A survey.*

Mishra, A. (2021). *Contrast limited adaptive histogram equalization (clahe) approach for enhancement of the microstructures of friction stir welded joints.*

PyTorch. (n.d.). *Automatic mixed precision package - torch.amp — pytorch 2.1 documentation.* https://pytorch.org/docs/stable/amp.html. (Accessed 22-12-2023)

Qi, J., Du, J., Siniscalchi, S. M., Ma, X., & Lee, C.-H. (2020). On mean absolute error for deep neural network based vector-to-vector regression. *IEEE Signal Processing Letters*, *27*, 1485–1489. Retrieved from http://dx.doi.org/10.1109/LSP.2020.3016837 doi: 10.1109/lsp.2020.3016837

Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O. R., & Jagersand, M. (2020, October). U2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, *106*, 107404. Retrieved from http://dx.doi.org/10.1016/j.patcog.2020.107404 doi: 10.1016/j.patcog.2020.107404

Raditya, C., Rizky, M., Mayranio, S., & Soewito, B. (2021). The effectivity of color for chroma-key techniques. *Procedia Computer Science*, *179*, 281-288. Retrieved from https://www.sciencedirect.com/science/article/pii/S1877050921000089 (5th International Conference on Computer Science and Computational Intelligence 2020) doi: 10.1016/j.procs.2021.01.007

Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-net: Convolutional networks for biomedical image segmentation.*

Smith, L. N. (2017). *Cyclical learning rates for training neural networks.*

Smith, L. N. (2022). *General cyclical training of neural networks.*

Smith, L. N., & Topin, N. (2018). *Super-convergence: Very fast training of neural networks using large learning rates.*

Wang, L., Lu, H., Wang, Y., Feng, M., Wang, D., Yin, B., & Ruan, X. (2017). Learning to detect salient objects with image-level supervision. In *Cvpr*.

Yang, S., Xiao, W., Zhang, M., Guo, S., Zhao, J., & Shen, F. (2023). *Image data augmentation for deep learning: A survey.*

Zhang, J. (2019). *Gradient descent based optimization algorithms for deep learning models training.*

Zhang, W., Pang, J., Chen, K., & Loy, C. C. (2021). *K-net: Towards unified image segmentation.*