Virtual Backgrounds Using Image Matting

Thomas Ramos (taramos), Lindsay Morel (lmmorel), Cheng Qian (chengqia)

Introduction

In the recent past, video calls have become a primary form of communication for work and school due to the global pandemic. With the closing of offices, schools, and studios, virtually everyone has been forced to use their homes as the centerpiece for their video calls. Many people are limited to the spaces with access to a computer and internet within their homes. These spaces can have distracting or unprofessional backgrounds for video calls. To minimize the effects this can have on people working virtually, they would like the ability to change the background of their video calls to an image of their choice that best fits the scenario or environment they are in.

Virtual backgrounds have been implemented in the past using the chroma keying method with green screens. This method replaces the pixels of a select color from an image or video with the desired background of choice [1]. However, this method will not work for most people as they do not have access to a green screen to use for their video calls to create a virtual background.

Another common technique is background subtraction. is a widely used approach for detection moving objects in videos from static cameras. The basic method is to calculate the foreground mask by performing a subtraction between the current frame and a reference frame [2]. However, this technique needs an initial background as a reference, and the reference needs to be updated often if there are some changes in the background.

A better solution to implementing a virtual background is to use image segmentation. Image segmentation is a computer vision process which is a very mature technology. This technique partitions the pixels in an image or video frame into segments that represent a recognizable image object that can be classified. The classified pixels can be used to separate the object or person in the foreground from the background in a video call. Once separated, the background can be changed and composited with the foreground to create a virtual background effect. Our team has implemented this process using a neural network trained to classify objects in images. This technique doesn't need any reference and it enables people to replace the background at home in everyday settings with a fixed or handheld camera, instead of a studio.

Approach

The project started with a search for a neural network pre-trained on classifying objects that can output an image that marks the classified pixels with a selected color. The search led our team to using a Residual Network (Resnet) model as they have proven themselves to be efficient and accurate classification models in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), being the 2015 Winner and 2016 runner up [3]. The ILSVRC is a yearly image classification competition where models are trained, validated, and tested using ImageNet's dataset with the winner being the model with the lowest error rate of mislabeling objects in images. The main idea of Resnet is introducing "identity shortcut connections" that skip one or more layers to overcome vanishing gradient problems [1].

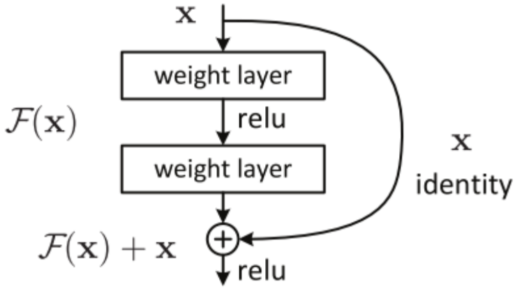
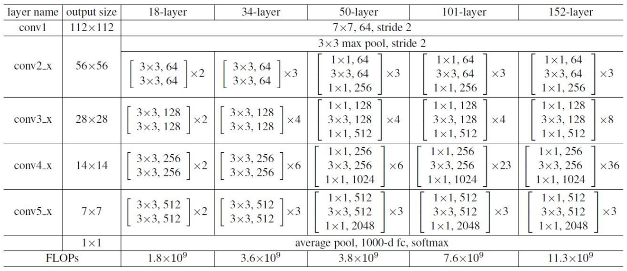

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 1: A simple residual block [1]

Additionally, our team selected the FCN-Resnet101 pretrained variant of the Resnet model as it can output a mask that classifies the pixels in an image.

The architecture of fcn_resnet101 is a fully-convolutional network with the backbone of Resnet 101. Resnet 101 is a neural network model with 101 layers. Table 1 shows the basic structure of layers for different backbones of Resnet. Here, we choose the architecture with 101 layers. It has 100 convolutional layers followed by an average pooling layer and a 1000-class fc layer [3].



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Table 1: Architecture of Resnet Variants [3]

Below, figure 2 shows the residual block of Resnet 101. For each small block, it stacks three convolutional layers of shape 1x1, 3x3, and 1x1, and the shortcut skips these three layers.

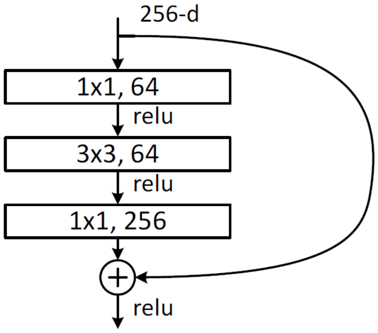


Figure 2: Residual block for 101-layer Resnet [4]

The model is pre-trained using the COCO train2017 dataset with the 20 Pascal VOC dataset categories. The output of the net is an ordered dictionary and the "out" key contains the output tensors. The output is of shape (1, 21, H, W). 21 is the number of classes (including the "background" class). H and W represent the height and width of the original input. At each pixel, there are probabilities corresponding to the prediction of each class. To get the label of each pixel, we can look for the largest probability over 21 classes. On the COCO val2017 dataset, fcn_resnet101 has a mean IOU of 63.7% and a global pixel wise accuracy of 91.9% [5].
Using the output of fcn_resnet101 from an image, our team created a mask that was applied to cut out the person or object in the foreground and that same shape in the new background. Then, the foreground of the image was composited onto the new background to create the virtual background effect.

Experiments

This model was tested on a subset of images and videos from the Matting Human Dataset developed by AISegment [6]. This dataset provides a diverse range of portrait images. For the purposes of image segmentation focused on people as subjects, this dataset was particularly useful.

Using images and videos from the Matting Human Dataset, initial testing of our implementation began with the calculation of a segmentation mask with the use of the model. This segmentation mask served the particular purpose of identifying and classifying the person in the image.

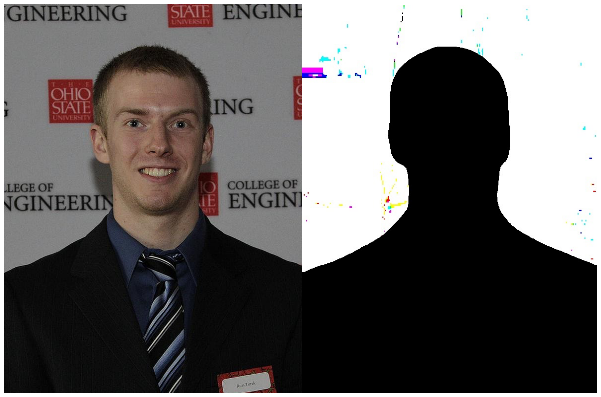


Figure 3: Original Image (left) and Segmentation Mask (right)

Then, once the segmentation mask is obtained, it is applied to the original image to extract the pixels representing the person. The inverse of the segmentation is applied to the background image to extract the background.

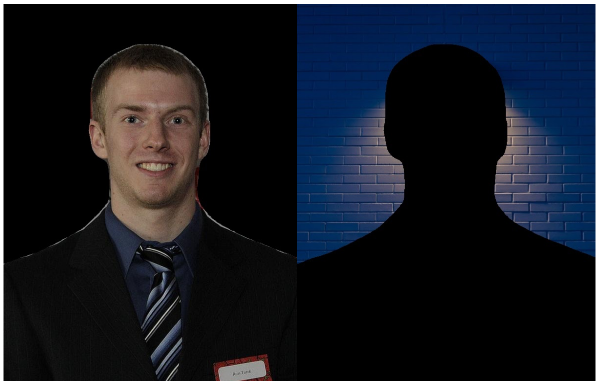


Figure 4: Segmentation mask applied to original image (left), inverse of segmentation mask applied to background image (right)

Finally, once both images shown in Figure 4 are obtained, they are simply added together, therefore compositing the foreground and background as intended. This implementation, as demonstrated on the following image, can also be applied to video data, where the process of segmentation, classification, and composition is done for each frame.

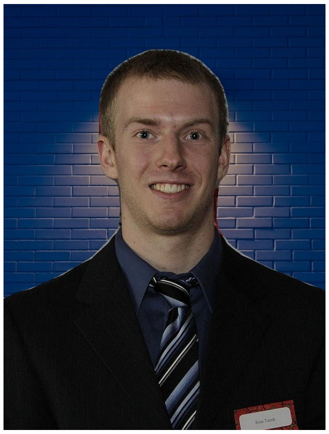


Figure 5: Final Composited Image, obtained from the addition of the two images shown in Figure 4.

In order to further evaluate the effectiveness of our implementation on a well-rounded scale, two primary experiments were designed with the intention of measuring success both qualitatively and quantitatively.

The first of these experiments involved a short survey which was administered to participants after the initial testing of the model. Participants were simply asked to rate the quality of the output video based on their visual evaluation of how well the foreground and background of the video were composited.

After receiving responses from a total of thirteen participants, the average rating given was approximately 4.46 out of 5. The result of our experimental survey for the collection of qualitative data is shown in the figure below.

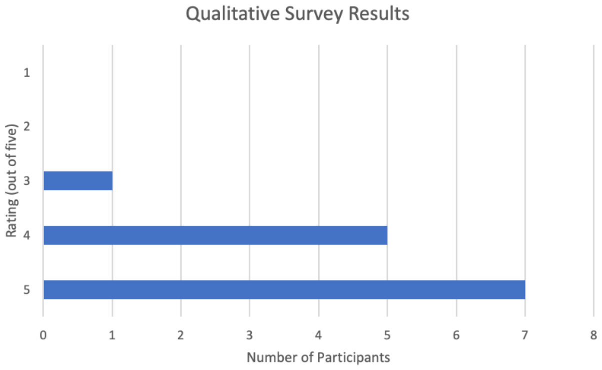


Figure 6: Data collected from experimental survey

The importance of evaluating the effectiveness of this approach qualitatively is seen in the recent and sudden shift toward remote interactions. At the end of the day, the effectiveness of any computer vision process should ultimately be evaluated on the basis of human perception; in the case of virtual backgrounds, where the effectiveness of an approach has arguably the most impact on users, this is particularly relevant.

The second evaluation of this model focuses on the quantitative effectiveness of the model's ability to both segment pixels in an image as well as to classify such segmented pixels. Because this this model is a pretrained variant of the fcn_resNet101 model, there already exists much data on the quantitative evaluation of this model on a number of datasets and for a number of purposes.

From PyTorch documentation [5], the FCN-ResNet101 pretrained models were trained on a subset of the COCO train 2017. Particularly, these variants were tested on the 20 categories present in the Pascal VOC dataset. According to the same published documentation by PyTorch, the evaluation of this pretrained model on the COCO 2017 validation set yielded a mean IOU of 63.7% and a global pixel-wise accuracy of 91.9% [5].

Because this model has been evaluated extensively in the areas of segmentation and classification, it was accepted that further collection of quantitative data was not necessary in order to evaluate the performance of this model. Additionally, the data collected through the experimental survey confirms the positive evaluation of our implementation of virtual backgrounds.

Implementation

Our image segmentation uses the Resnet model as well as its pretrained parameters. This is the only part off the shelf.

The following things are done by us. Our dataset is loaded from Google Drive. For the video, we write a small function to down scale the input video in order to avoid excessive running time. We also write a function to get every frame of the video with JPG format and stack these frames in a large tensor with the size of (N, H, W, 3), where N is the number of frames. The steps for processing the images and videos should be similar. We feed the images or frames into the pretrained model. We get the mask for image segmentation from the output by selecting the pixels with the label of 15. The number 15 is corresponding to the "people" class.

Then, we resize the input background to make it have the same size as the image with foreground. After this, we composite the foreground and background together by applying the mask and its inverse to the foreground and background, respectively, then adding the results. The process of compositing the foreground and background can be seen in the figures below.

The final composited image is output as a JPG to a desired destination. In the case of a video, each composited image is stitched together with the video's original frame rate and output as an MP4 file.

References

[1] V. Feng, “An Overview of ResNet and its Variants,” *Medium*, 17-Jul-2017. [Online]. Available: https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035.

[2] S. Kench, “What is Chroma Keying and How Does it Work?,” *StudioBinder*, 05-Apr-2021. [Online]. Available: https://www.studiobinder.com/blog/what-is-chroma-key-green-screen/.

[3] S.-H. Tsang, “Review: ResNet - Winner of ILSVRC 2015 (Image Classification, Localization, Detection),” *Medium*, 20-Mar-2019. [Online]. Available: https://towardsdatascience.com/review-resnet-winner-of-ilsvrc-2015-image-classification-localization-detection-e39402bfa5d8.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385,2015.

[5] PyTorch. [Online]. Available: https://pytorch.org/hub/pytorch_vision_fcn_resnet101/.

[6] L. H., “AISegment.com - Matting Human Datasets,” 06-Jun-2019. [Online]. Available: https://www.kaggle.com/laurentmih/aisegmentcom-matting-human-datasets. [Accessed: 26-Apr-2021].