Rotoscope Automation With Deep Learning

By Oscar Estrada Torrejon, Nicholas Peretti, and Ricardo Figueroa

Abstract

We present a deep learning-based algorithm that can automatically rotoscope people in a given scene, without any user input. Current approaches to image matting require a significant amount of human input, irrespective of whether it is manually rotoscoped or through a chroma key. This study shows that this algorithm can perform as well as and even surpass the rotoscoping capabilities of the Adobe After Effects' RotoBrush tool, in a variety of scenes comprising different lighting conditions, movements, and subjects. This makes it suitable for integration within a visual effects (VFX) pipeline.

Keywords

Compositing, deep learning, image matting, instance segmentation

Introduction

Image matting is one of the most common techniques used in filmmaking. It consists of extracting elements from different scenes and combining them into a new, composite shot. This extraction is performed through the use of a matte, i.e., a black and white frame, in which the white pixels denote the foreground elements; black pixels, the background elements; and gray pixels, the semitransparent elements.[1] The process of creating a matte is vital for VFX pipelines, since it allows filmmakers to place actors in environments that do not exist in real life.

However, the matting process is a very mechanical and resource-intensive one, since it requires the skills of a professional artist to achieve a clean, polished result. Factors such as the number of elements to extract, amount of detail, transparency, and the length of the shot can make this process even more complicated and, in turn, require the use of multiple digital tools and additional man-hours. Often, matting becomes a bottleneck in the post-production pipeline.

To overcome these issues, there has been a considerable amount of research devoted to developing algorithms that can generate a polished matte, while minimizing user input. For example, the Adobe After Effects' RotoBrush tool predicts the foreground elements' boundaries in a given frame and propagates them to the subsequent ones, reducing the manual labor required.[2] This raises the question as to whether there is a way to automatically extract a matte from an arbitrary scene, without any user input.

The computer vision field is well known for having deep learning algorithms that can automatically detect and segment objects within an image. Therefore, it is worth evaluating whether these object detection algorithms can be utilized in the VFX industry to produce mattes that are, with minimal labor, ready for implementation.

> **The computer vision field is well known for having deep learning algorithms that can automatically detect and segment objects within an image. Therefore, it is worth evaluating whether these object detection algorithms can be utilized in the VFX industry to produce mattes that are, with minimal labor, ready for implementation.**

Background

Image Matting

To properly understand the complexity of the matting process, it is useful to frame it from a mathematical perspective. The equation below describes the input image as a linear combination of its foreground and background elements[3]:

$$I_p = \alpha_p F_p + (1 - \alpha_p) * B_p \quad \alpha_p \in [0,1] \tag{1}$$

where $p(x, y)$ corresponds to a specific pixel in an image, I_p is the intensity value for that pixel, F_p and B_p are the intensity values for the foreground and background elements, respectively, and α_p is the opacity of the foreground. The opacity value is important for semitransparent objects such as glass, whose final intensity value is a combination of the glass itself and the background behind it. Usually, only I_p, the intensity of the final image is known, while F_p, B_p, and α_p are unknown.

Digital Object Identifier 10.5594/JMI.2019.2959967
Date of publication: 6 March 2020

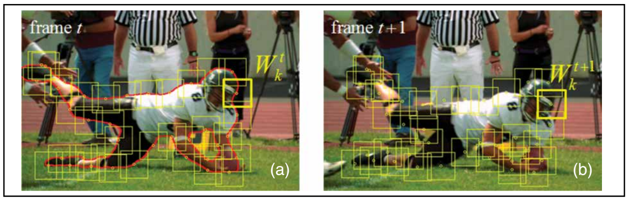


FIGURE 1. After Effects' RotoBrush tool. (a) Given an input boundary around the player, a series of overlapping windows are created, which compute local segmentation. (b) Then, this matte is propagated to the next frame.

For a three-channel RGB image, the matting problem is severely underconstrained

$$\begin{pmatrix} I_R \\ I_G \\ I_B \end{pmatrix}_p = \alpha \begin{pmatrix} F_R \\ F_G \\ F_B \end{pmatrix}_p + (1-\alpha) \begin{pmatrix} B_R \\ B_G \\ B_B \end{pmatrix}_p . \qquad (2)$$

It has seven unknowns, but only three equations to solve them. Therefore, user input that provides additional constraints is required, such that the equation can be properly solved.

In most productions, these constraints are defined by two main methods: chroma keying[4] and rotoscoping.[5] Chroma keying consists of placing the actors in front of a green background that is evenly lit and has a constant hue. Then, in post-production, algorithms take advantage of this constraint to isolate the actors. However, the green screen setup may not fit all productions or scenes, and significant expertise is required to correct for any issues during capture, such as a misexposure.[6]

Because chroma keying has a very specific set of constraints and there are algorithms specifically tuned to that approach, it is not as interesting as the rotoscoping approach, which is more flexible but more labor-intensive.

Manual Rotoscoping

In contrast to chroma keying, manual rotoscoping does not rely upon a well-defined background to perform the foreground/background separation but rather on manual input from an artist. Even though there are multiple techniques to approach and make this process more efficient, it still requires the artist to outline the subjects to extract, frame by frame. It is important that the artist pays attention to fine details of the actor, such as their hair, and partial transparencies that may occur, such as glasses or motion blur. These features need to be carefully processed since they could carry background pixels with them and affect both the quality of the composite and the authenticity of the story. In addition, it is important that the matte is consistent over time, as any unnatural temporal deviations may attract the attention of the observer.

Due to the high amount of work that is required to create a successful rotoscope, software developers have included tools that are aimed to ease this process in their VFX programs. Tools such as keyframe interpolation or planar tracking can aid the process by propagating the mask over time, thus minimizing the amount of refinement required from the user.

There are also more advanced algorithms dedicated to predicting the location and shape of the foreground objects, based on an initial input from the artist, effectively reducing the time spent in the process.

Adobe After Effects' RotoBrush Tool

One of the most prominent tools available to ease the rotoscoping process is the Adobe After Effects's RotoBrush tool.[2] This function is based on the paper Video SnapCut from SIGGRAPH'09, by Bai et al. It works by receiving scribble inputs from the user defining the foreground and background areas of the image. Then it proceeds to predict the boundary of the elements, which can be adjusted through further user input. This prediction also extends into the temporal domain, since it propagates the mask throughout the sequence of images (**Fig. 1**).

To accurately predict the matte, this algorithm takes advantage of local segmentation. Given the initial input outline, it divides it into multiple overlapping windows (boxes) along the boundary. Then, a local classifier is run on each window to separate the foreground and background pixels. This classifier is based on color and shape models, since it assumes that foreground elements will have a distinct color and shape profile than their background.[7] The color and shape models operate independently and are assigned a confidence value, which determines their weight when they are combined into a single matte. In that way, if the foreground and background elements have similar color tonalities, the shape model can ensure an accurate segmentation. The output of each local box is then combined to create a single matte for the given frame.

For propagation to a next frame, the algorithm takes advantage of the Scale Invariant Feature Transform (SIFT).[8] This is an algorithm that finds and matches key features from two subsequent frames. Thus, SIFT is used to align the frames and move all boxes together. Then, local optical flow is computed to account for the deformation of the subject between frames, adjusting each segmentation window independently.

User input is accepted throughout this stage to ensure that the predicted mask is correct and that the propagation does not add any errors.

Given that the purpose of these algorithms is to reduce user input as much as possible, it is therefore interesting to test an approach that would require minimal to no input, based on deep learning.

Deep Learning Approach

In the past few years, there have been considerable advances in the field of computer vision, that include utilizing deep learning algorithms to classify images based on their content and segment them into different object instances.[9–12]

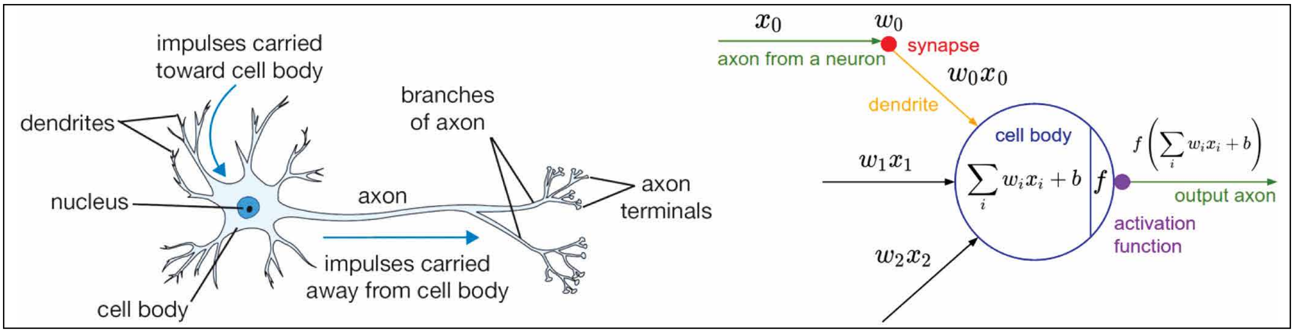


FIGURE 2. Illustration of a real neuron structure (left) versus an artificial neuron (right). Note the similarity in their information processing.

Although image matting and object classification are different problems, for the case in which there is no partial object transparency [α_p is either 0 or 1 in (1)], the matting problem can be reduced to a simple classification one, in which the task is to discern whether a given pixel p belongs to the foreground or the background class. Therefore, it is of special interest to analyze the performance and potential of these algorithms to automate the matting process.

Most object classification algorithms are based on neural network architectures. These networks extract image features at different scale and semantic levels and output a vector with the probabilities of the image belonging to the given classes. A neuron in these networks functions similarly to the ones in the human brain (**Fig. 2**). Each neuron within a layer is connected to multiple neurons from its previous layer. It takes the signals from those neurons and weight averages them. Then it computes an activation function (usually a nonlinear one) before sending the signal to the neurons in the next layer.[13]

Although there are different types of network architectures, the most common ones for image classification are called convolutional neural networks (CNNs).[13] They are formed by convolutional layers, where the neurons are organized in 3D volumes, representing the three dimensions of an image (height, width, and channel depth), rather than single columns (**Fig. 3**). Each neuron in a CNN is connected to a local 3D region on the previous layer rather than to every single neuron as in a traditional architecture. The size of this local 3D region is the same for all neurons in the layer. This allows the same set of parameters to be applied at each neuron in the spatial dimension, effectively performing a filtering operation. This parameter sharing reduces the total number of parameters to be trained, making the process more efficient. Multiple filters can be stacked within each convolution layer to extract different features at once, giving the layer its depth dimension.

Other layers that form the CNN architecture include rectified linear unit (ReLu), which compute the activation (nonlinear) function, allowing the network to model more complex functions, and Pool layers, which downsample the features from previous layers to analyze it at multiple scales.

For this experiment, we used a CNN developed by Google, called Xception,[9] and adapted it to identify human subjects in a variety of situations.

Our Approach

Xception

Xception[9] is a network developed by François Chollet at Google which performed extremely well in the ImageNet Large Scale Visual Recognition Competition (ILSVRC), an object classification challenge.[14]

It is based on the architecture of Inception,[12] another network developed by Google, which was the first to propose the separation of cross-channel and spatial correlations as a way to improve computational efficiency.

As previously mentioned, the parameters of a regular convolutional layer correspond to a set of learnable filters. These filters are small in the spatial dimension, but extend through the full depth of the

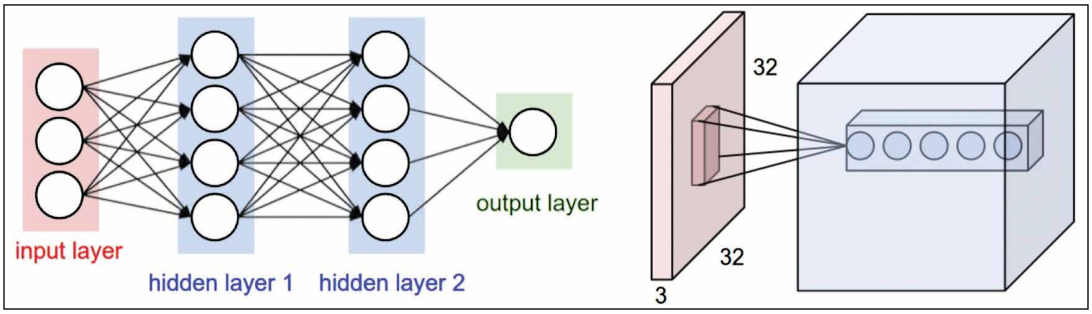


FIGURE 3. Neural network architectures. Traditional neural network (left). CNN (right). Note the structural difference. In the CNN, each new depth layer corresponds to a stacking of filters.

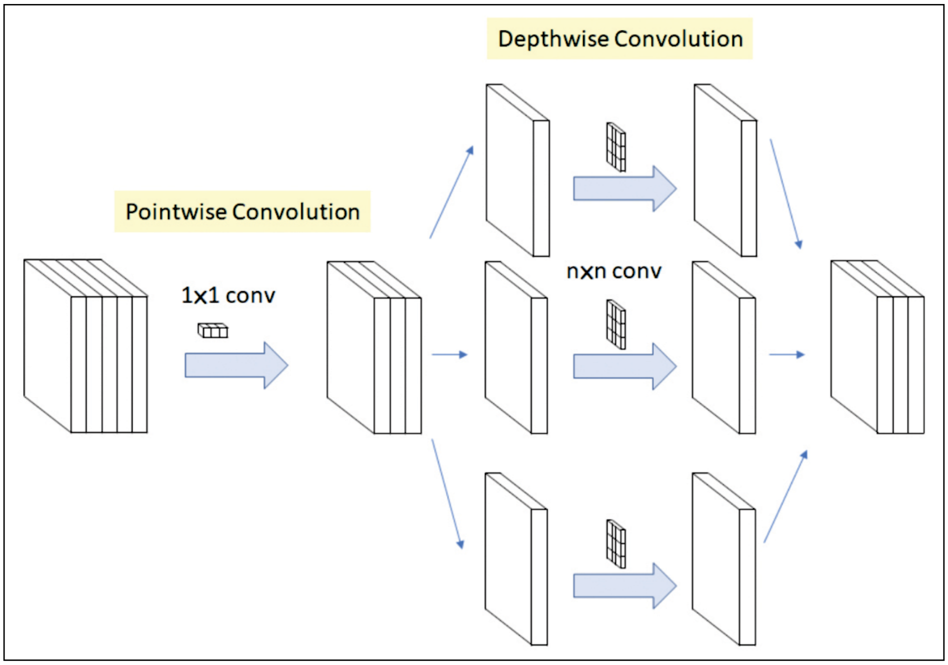


FIGURE 4. Xception's depth wise separable convolution. First a set of 1 × 1 convolutions calculate channel cross correlations, and then 2D filters are used for spatial cross correlations, improving the computation's performance.

input volume.[13] Therefore, they compute both channel and spatial correlations simultaneously. Inception initially proposed a way to separate these operations since their correlations can be assumed to be independent of one another. Xception took this assumption further by completely separating them and making them fully independent (**Fig. 4**).

At its core, Xception is comprised of an "Xception module," which is a series of depthwise separable convolutions stacked together. A depthwise separable convolution first computes a set 1×1 convolution across the entire image depth, to compute channel cross correlations, and then applies spatial 2D filters to each of the output channels. This procedure has proven to be more efficient than traditional 3D filters and even outperformed the Inception network.

Our Approach

Our network utilizes an Xception backbone implemented in a pyramidal array.[15] After each Xception block, the extracted features are downsampled spatially by a factor of 2. After six Xception blocks, upsampling blocks scale back the feature maps by 2, until the original resolution is achieved. This is because low-resolution features have more semantic meaning, i.e., they provide more information about the content of the image, whereas high-resolution features provide better spatial information. At each downsampling operation, residual connections that link to upsampling layers deeper in the network are created. This allows high-resolution features to be combined with semantic details from deeper in the network.[11] In **Fig. 5**, yellow layers are the added upsampling blocks that perform bilinear upsampling, convolution, batch normalization, and then a ReLu activation.

The use of downsampling after every Xception block allows the network to analyze features at multiple scales. This corresponds to the extraction of semantically significant information, such as the presence of a human subject or not. Then, the upsampling layers along with the residual connections provide spatial information to accurately predict where the person is located.

Experimental Procedure

The experiment compared our algorithm's output to mattes generated by manual rotoscoping (which we call ground truth), as well as to the output of After Effect's RotoBrush tool. The test footage for this evaluation contained subjects with no partial transparency or motion blur, to reduce (1) to a binary classification problem and to remove the need for the user input.

Equipment

The algorithm was trained and tested on a system using two NVIDIA GTX 1080ti GPUs, with the data stored locally in the system.

The details of the cameras and settings used to capture the test footage are as follows:

- Sony FS5 Mark 2. Recorded at 1080p, 24 frames/sec, 2.2 Gamma, Rec. 709 color space and ProRes 422 HQ.
- Arri Alexa Mini. Recorded at 1080p, 24 frames/sec, 2.2 Gamma, Arri Look Classic 709, ProRes 4444.

The ground truth was created using the Roto node in NukeX 11.

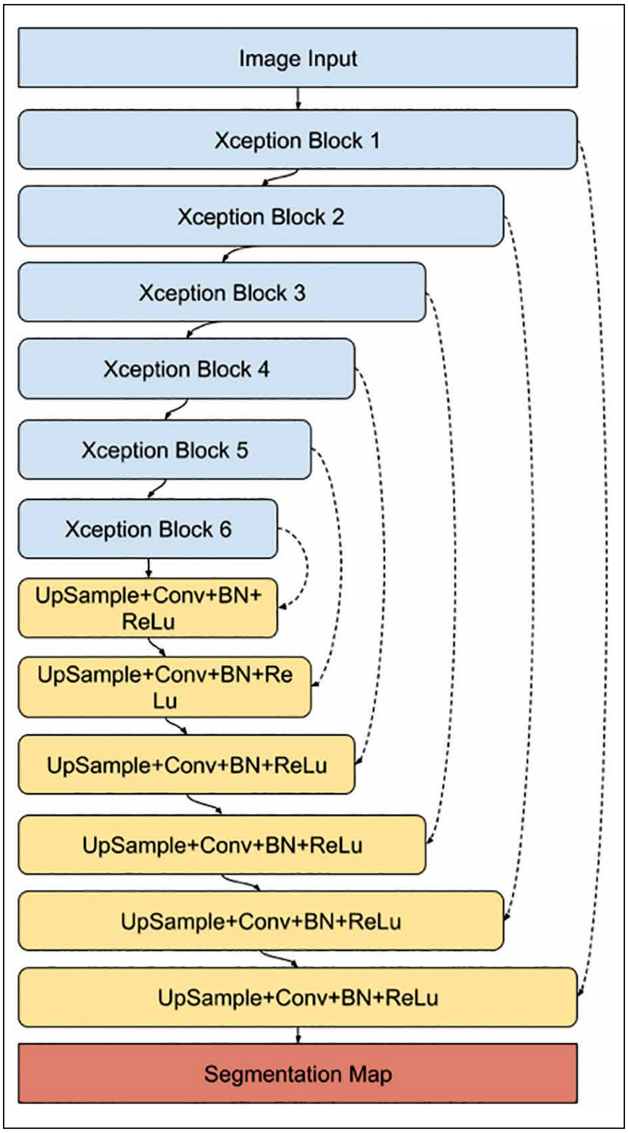


FIGURE 5. Our network architecture. The blue blocks correspond to Xception modules described previously. The yellow blocks correspond to the layers we added for locating the people within the image. Note how they are arranged in a pyramid structure, to take advantage of both semantic and high-resolution location features.

The RotoBrush matte was computed on After Effects CC 2018.

Data Set

The Xception backbone of the network was pretrained on the ImageNet data set, which contains more than 14 million images that have been hand annotated for object recognition across 20,000 categories.[14] The full data set had been fed to the backbone ten times during training (ten epochs). The significant amount of data used during pretraining allowed us to simply fine-tune the network to our specifications using transfer learning,[13] rather than to train it from scratch.

The second half of the network was trained on Common Objects in Context (COCO),[16] on the person category only. This data set contains roughly 46,000 images with segmentation masks for people. These images were separated randomly in 41,000 for training and 5,000 for validation. The network was trained using the Adam optimizer, with an initial learning rate of 0.001 and a batch size of 4.

It was important to ensure that the training data reflected the multiple scenes in which the algorithm may be utilized, and accounted for any problems during shooting since not all scenes will be perfect straight out of the camera. With this in mind, augmentation operations were applied to the COCO images. Some of these augmentations included modifications in rotation, blur, saturation, noise, exposure, and warping.

Test Images

Just like the training batch, the test images needed to account for different shooting conditions and content relevant to professional productions. Therefore, a test batch was captured using the equipment listed in the section titled "Equipment," to test the network under different input conditions:

- outdoor versus indoor scenes
- type of shot (wide, medium, and close up)
- number of people in the shot
- improperly exposed shots
- people movement (walking sideways, walking to/from camera, and rotating)
- amount of detail (edge quality).

The shooting took place at Magic Spell Studios, Rochester, NY. Once the footage was shot, one second (24 frames) of each scene was extracted and transcoded to TIFF images to feed them into the network.

Ground Truth and RotoBrush Mattes

The creation of the ground truth mattes was outsourced to students in the School of Film and Animation, Rochester Institute of Technology. These students were recommended by faculty members based on their rotoscoping skills. The mattes were created manually using the roto node in NukeX 11. The ground truth was delivered as black-and-white TIFF files, where the white pixels correspond to the foreground and the black ones to the background.

The creation of the RotoBrush matte was performed in After Effects 2018. For this, a key or reference frame was chosen among the 24, and a matte was created and refined by the user for that single frame. Then, the algorithm propagated this initial matte to the adjacent frames, without further input from the user. Just as with the ground truth, the matte was exported as TIFF files.

Metrics

The metric chosen to evaluate the matte quality was Intersection over Union (IOU). IOU is utilized in competitions such as the PASCAL Visual Object Classification (VOC) challenge,[17] which is a benchmark in visual object recognition.

Given a ground truth and test boundaries, the IOU is defined as:

$$\text{IOU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \qquad (3)$$

where the area of overlap is the number of pixels that are common to both the foreground boundaries, and the area of union is the number of pixels that both boundaries encompass when placed one on top of the other. The IOU is therefore a metric that calculates the degree of overlap between both the ground truth and the test areas. If they overlap perfectly, then the IOU will have a value of 1, whereas if they are totally different, the IOU will have a value of 0. Since we compared the overlap between the ground truth and the test mattes, it was an appropriate metric to be used.[18]

Results

The results from the network are shown below for different tests. The IOU metric is also included for both the network and the RotoBrush approaches. It is worth noting that the artists who generated the ground truth mattes spent two to ten hours on average per 24 frames. On the other hand, the algorithm processed each frame in 0.38 seconds on average, or each shot of 24 frames in 9.12 seconds. This significant amount of time saved could be spent by artists for refining the matte from the network rather than having to create it from scratch.

Person Wide Walking

This shot was used to test the performance of the network in an indoor setting, which is usually characterized by a lower amount of light available. In addition, the framing was chosen to include the whole actor in the shot and make him move across the entire scene. **Figure 6** shows one of the input frames to the network, with its respective ground truth, and **Fig. 7** shows the output mattes from the network and RotoBrush. As can be seen, the network detects the person very well, but struggles with the shoes, which are very dark, and therefore may not have enough detail to be classified as part of a person.

Below are the IOU computations per frame, for the 24 frames extracted for this scene (**Fig. 8**). It can be seen that the network outperformed the RotoBrush tool in After Effects in most of the frames. The RotoBrush's peak performance occurred at the frame in which the refined matte was created and decayed as the frames got farther away from the refined matte.

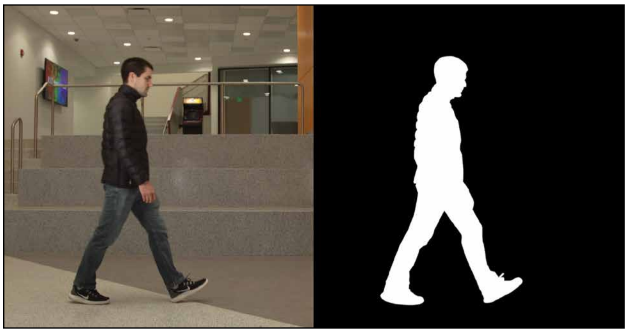


FIGURE 6. Input TIFF to the network (left). Ground truth matte (right).

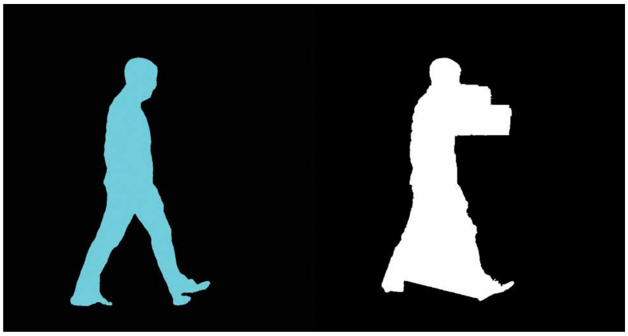


FIGURE 7. Matte from the network (left). Matte from RotoBrush (right).

Person Medium Rotation

This shot was also indoors, and the goal was to present the network with a differently scaled actor. This shot included a medium close up, as opposed to the previous wide one. In addition, the actor was asked to incline his body sideways, to deviate his head and body from the vertical axis (**Fig. 9**). It can be seen that the network did a good job in outlining the edges of the subject, such as his jacket's neck and ears, but struggled with very dark areas such as the bottom part of his jacket (**Fig. 10**).

Looking at the IOU metrics per frame (**Fig. 11**), it can be seen that the network had a similar performance to the RotoBrush tool, achieving over 0.9 IOU for the whole scene. Its performance decayed toward the end of the shot, in which the actor was not vertical. However, it can be seen that the actor's edges were preserved consistently, so the main cause of the decreased performance was the underexposed areas of his jacket and not the deviation from the vertical axis.

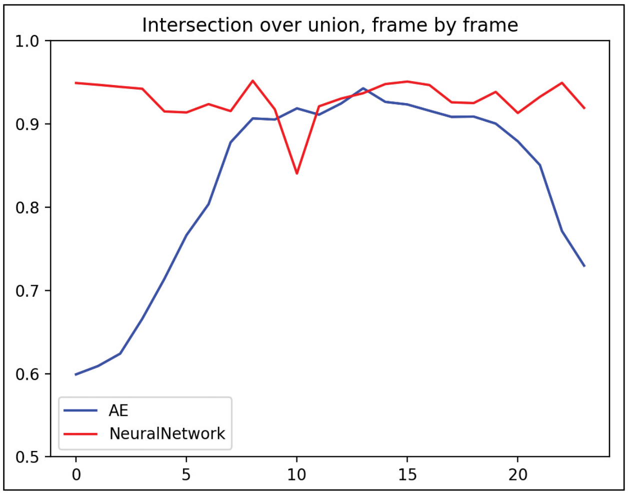


FIGURE 8. IOU metrics for the person wide walking shot. As can be seen the network remains fairly consistent at high accuracy, whereas the RotoBrush struggles with propagation.

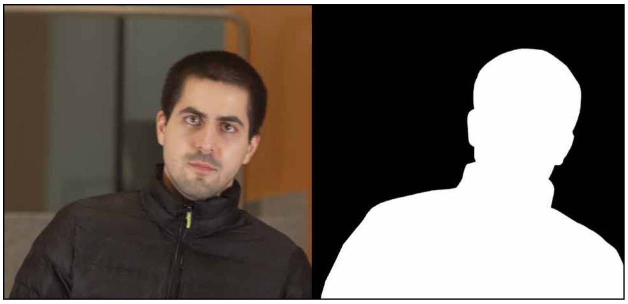


FIGURE 9. Input TIFF to the network (left). Ground truth matte (right).

Person Medium Yaw Overexposed

This shot presented the actor indoors in a medium shot. This time the actor was asked to rotate along his vertical axis and the right side of his face was overexposed by half a stop, to create a higher dynamic range across the person (**Fig. 12**). Once again, it can be seen that the network struggled with the darker parts of the jacket, while the outlines and edges were very similar to the ground truth, even if the actor was not lit uniformly (**Fig. 13**).

By looking at the IOU metric, it can be determined that the network very closely followed the RotoBrush's performance, achieving almost 1.0 across the shot. The network has a small dip toward the end of the sequence of frames, but this is mainly due to the darker regions of the jacket and not a problem with the edges, as seen in **Fig. 14**.

Person Medium to Wide

This shot presented a progression from a medium to a wide shot and tested the network's consistency with a change in the actor's scale. In addition, the subject was facing away from the camera, to evaluate the network's ability to detect people when facial features are not present (**Fig. 15**).

It can be seen that the algorithm accurately outlined the subject at the distance, even with the presence of dark elements such as the jacket. However, it had some trouble discerning the fingers on the right side, taking the whole area as a solid hand (**Fig. 16**).

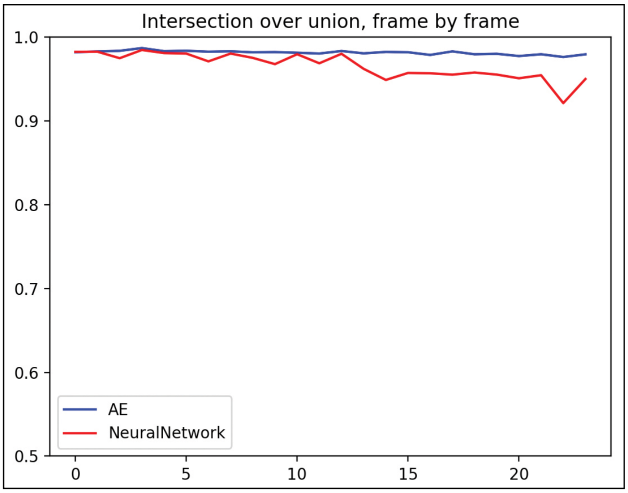


FIGURE 11. IOU metrics for the person medium rotation shot. Note how the network performs similar to RotoBrush, but decays in performance due to dark regions in the jacket.

The IOU plot reflects the high accuracy of the algorithm, which again is placed above 0.9 across the entire duration of the shot. It also seems to outperform the RotoBrush tool at several frames (**Fig. 17**).

Two People Outdoor Wide

This shot presented two people of different skin tones on camera in an outdoor setting. These new variables were chosen to see how the algorithm would perform in more complex scenes (**Fig. 18**).

It can be seen that the overexposed and cluttered background had an impact on the performance of the algorithm since it incorrectly classified some of the trees as people. However, it was able to identify both actresses on the scene and outline them without any problems. Once again, it struggled on the darkest areas, such as the dark pants of the actress on the left (**Fig. 19**).

The IOU heavily penalized this mislabeling of the background elements, reducing the network's performance to merely over 0.7 (**Fig. 20**). This is because the extra area in the background does not intersect with the ground truth data. However, it can be seen that it was pretty consistent across frames.

If we zoom in to the areas of interest, disregarding the background clutter, we can better appreciate the algorithm's

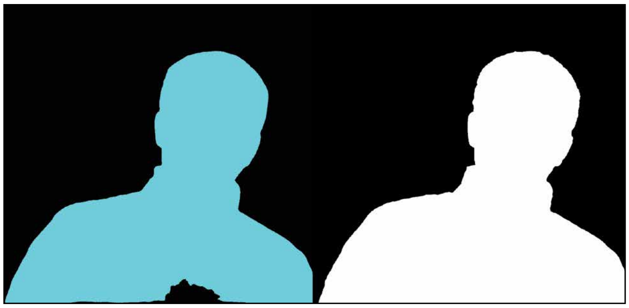


FIGURE 10. Matte from the network (left). Matte from RotoBrush (right).

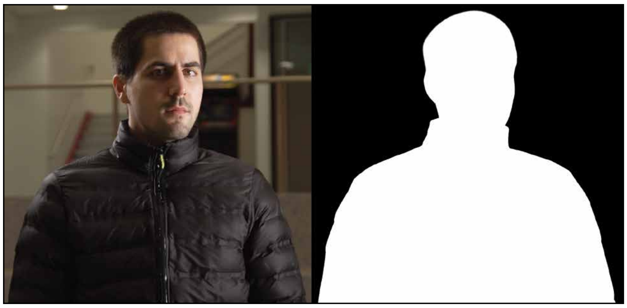


FIGURE 12. Input to the network (left). Ground truth matte (right).

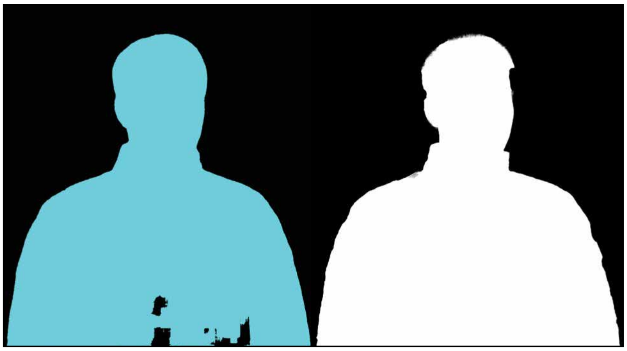


FIGURE 13. Matte from the network (left). Matte from RotoBrush (right).

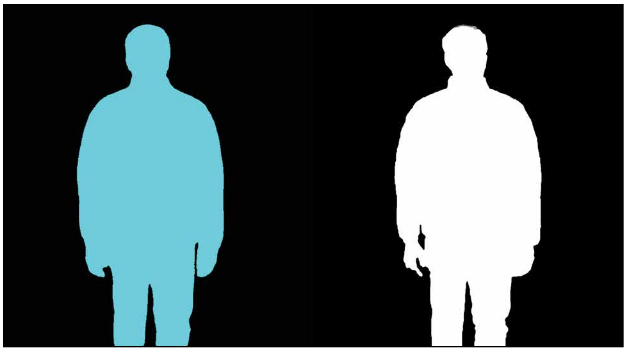


FIGURE 16. Network output matte (left). RotoBrush output matte (right). Note the detail on the hands, as well as on the neck.

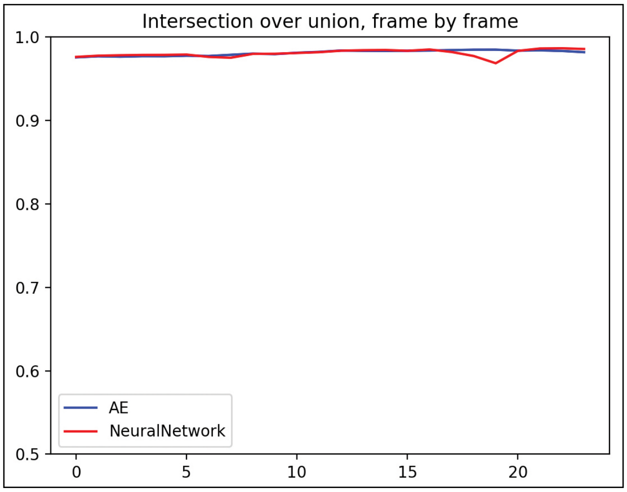


FIGURE 14. IOU metrics for the person medium yaw overexposed shot.

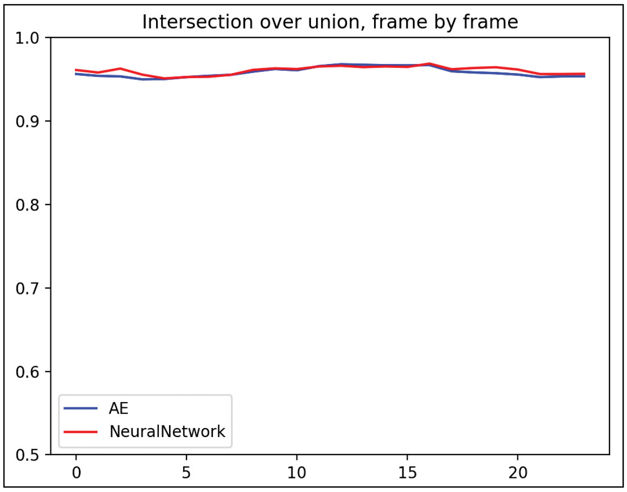


FIGURE 17. IOU metrics for the person medium to wide shot.

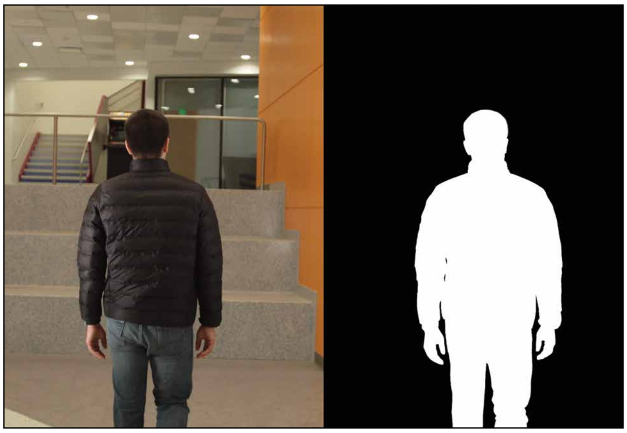


FIGURE 15. Input image to the network (left). Ground truth matte (right).

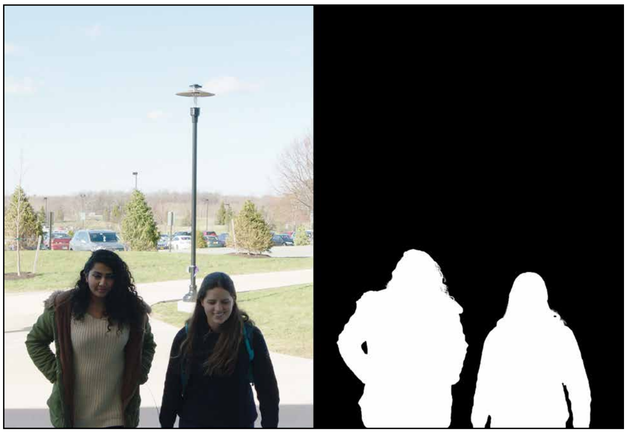


FIGURE 18. Input image to the network (left). Ground truth matte (right).

job on the silhouette of the person (**Fig. 21**). Since the IOU metric was not determined by a bad performance on the subjects' outline, then a refinement algorithm could be implemented to reduce the number of detections to the number of subjects identified in the scene.

Composite Example

As a practical way of testing the algorithm's feasibility to generate a matte for VFX, one of the shots was taken and composited onto an arbitrary background, to better evaluate the edge quality (**Fig. 22**).

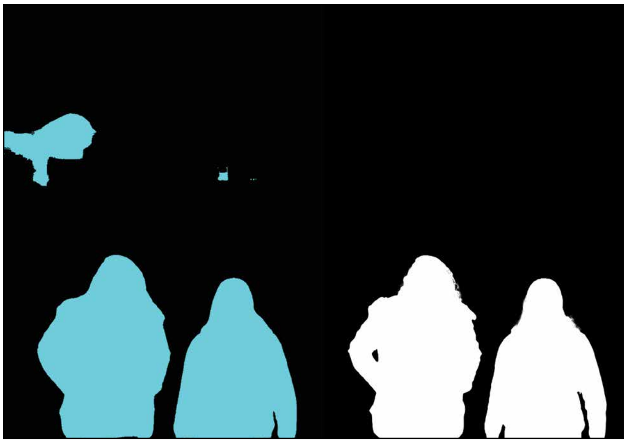


FIGURE 19. Output matte from the network (left). Matte from the RotoBrush (right).

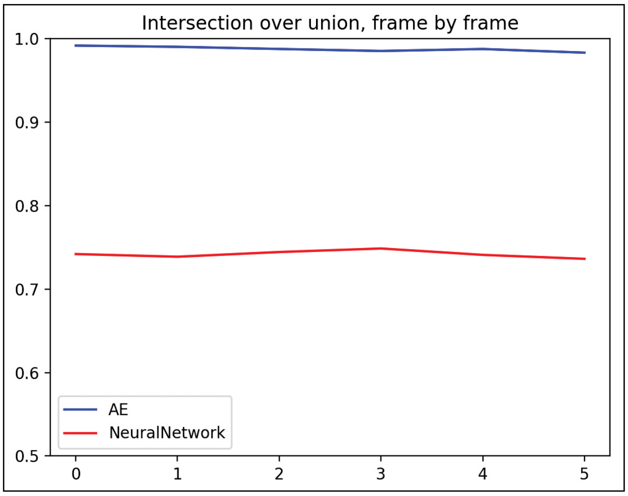


FIGURE 20. IOU Metrics for the two people outdoor wide shot. Note how the incorrect classification of the background elements heavily penalized the network.

It can be seen that the outlines are very similar between both the ground truth and the algorithm. The main difference arises with small details such as the fingers on his left hand, where the algorithm incorrectly classified background pixels as pertaining to the person.

Conclusion

We presented an automated way to rotoscope people, given an arbitrary sequence of images. For the different tests conducted, our algorithm performed to the standard, matching or even exceeding the performance of Adobe After Effect's RotoBrush tool in some shots. Even though in other shots it missed parts of the actor due to dark regions, and included elements from the background, the overall performance on the edges was quite satisfactory. The algorithm seemed to perform better than RotoBrush when there was significant movement from the actor across frames, and more tests will need to be performed in order to validate this observation. In terms of labor hours, the algorithm outperformed the traditional rotoscoping method, from an average of five hours to ten seconds approximately per shot. This huge time saved would give a head start to artists and allow them to focus on refining the matte output rather than creating it from scratch, which would improve the productivity of the whole VFX pipeline.

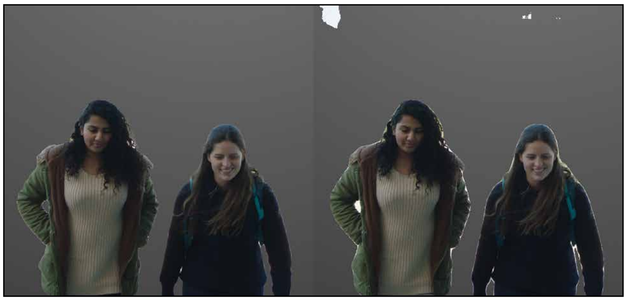


FIGURE 21. Actresses extracted using the ground-truth matte (left). Actresses extracted using the network matte (right). Note how their edges are well preserved, despite the low IOU performance.

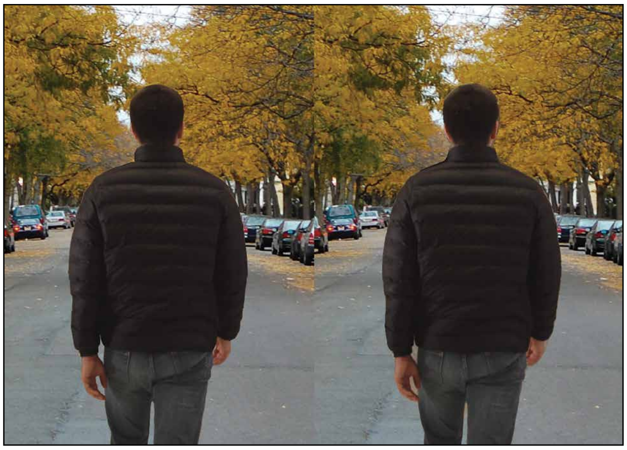


FIGURE 22. Composite created with the ground truth matte (left). Composite created with the output of the network (right). Note the difference in the fingers, as well as in the neck.

Next Steps

The scope of this project was to provide insights into whether deep learning architectures designed for object detection could be used in the Visual Effects world, so most of the time was spent on testing its performance with basic settings and conditions, which showed promising results. Some areas for further improvement are outlined below.

Hyperparameter Tuning

Due to time constraints and resources, there was not much time available to try out different types of hyperparameters for the network, including the size of each convolution layer, the strides, and the depth of the overall network, as well as the learning rate. By dedicating time to tuning these parameters on the validation set defined above, improvements in the results could be obtained.

Temporal Dependency

Another avenue for improvement would be to consider adding a correlation in the temporal domain. Currently, the network takes and processes each frame independently from each other. On the other hand, the RotoBrush tool utilizes SIFT and optical flow cues to propagate the masks across the different frames and calculate its predictions. There are segmentation data sets specifically tuned for videos, such as Densely Annotated Video Segmentation (DAVIS), which could represent a good area for improvement. Also, adding a gate recurrent unit to leverage features present in sequential frames could be carried out.

Edge Refinement

One last option would be to implement an edge refining technique, such as feathering, which would help compositing be more visually smooth, or a hole-filling algorithm that could compensate for the inside areas that were deemed underexposed.

References

1. N. Xu, B. Price, S. Cohen, and T. Huang, "Deep Image Matting," *Proc. 2017 IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, 2017. Accessed: 2 June 2019. doi: 10.1109/cvpr.2017.41

2. X. Bai, J. Wang, D. Simons, and G. Sapiro, "Video SnapCut," *ACM SIGGRAPH 2009 Papers*, 2009. Accessed: 2 June 2019. doi: 10.1145/1576246.1531376

3. Q. Zhu, P. Heng, L. Shao, and X. Li, "What's the Role of Image Matting in Image Segmentation?," *Proc. 2013 IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, 2013. Accessed: 2 June 2019. doi: 10.1109/robio.2013.6739711

4. A. Smith and J. Blinn, "Blue Screen Matting," *Proc. 23rd Annu. Conf. Comput. Graphics Interactive Techn.—SIGGRAPH'96*, 1996. Accessed: 2 June 2019. doi: 10.1145/237170.237263

5. M. Seymour, "The Art of Roto: 2011," *fxguide*, 2019. Accessed: 2 June 2019. [Online]. Available: https://www.fxguide.com/featured/the-art-of-roto-2011/

6. Y. Aksoy, T. Aydin, M. Pollefeys, and A. Smolić, "Interactive High-Quality Green-Screen Keying via Color Unmixing," *ACM Trans. Graphics*, 36(4):1, 2016. Accessed: 2 June 2019. doi: 10.1145/3072959.2907940

7. C. Singh, "Rotobrush," Cmsc426.github.io, 2019. Accessed: 2 June 2019. [Online]. Available: https://cmsc426.github.io/rotobrush/

8. D. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proc. 7th IEEE Int. Conf. Comput. Vision*, 1999. Accessed: 2 June 2019. doi: 10.1109/iccv.1999.790410

9. F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *Proc. 2017 IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, 2017. Accessed: 2 June 2019. doi: 10.1109/cvpr.2017.195

10. S. Tsang, "Review: Xception—With Depthwise Separable Convolution, Better than Inception-v3 (Image Classification)," Towards Data Science, 2019. Accessed: 2 June 2019. [Online]. Available: https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568

11. J. Olafenwa, "Review of Deep Residual Learning for Image Recognition," Medium, 2019. Accessed: 2 June 2019. [Online]. Available: https://medium.com/deepreview/review-of-deep-residual-learning-for-image-recognition-a92955acf3aa

12. C. Szegedy et al., "Going Deeper with Convolutions," *Proc. 2015 IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, 2015. Accessed: 2 June 2019. doi: 10.1109/cvpr.2015.7298594

13. A. Karpathy, "CS231n Convolutional Neural Networks for Visual Recognition," Cs231n.github.io, 2019. Accessed: 2 June 2019. [Online]. Available: http://cs231n.github.io

14. J. Hui, "Understanding Feature Pyramid Networks for Object Detection (FPN)," Medium, 2019. Accessed: 2 June 2019. [Online]. Available: https://medium.com/@jonathan_hui/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c

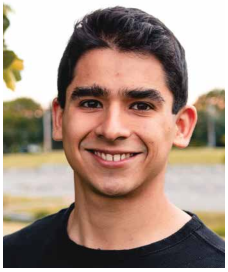
15. J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," *Proc. 2009 IEEE Conf. Comput. Vision Pattern Recognition*, 2009. Accessed: 2 June 2019. doi: 10.1109/cvpr.2009.5206848

16. T. Lin et al., "Microsoft COCO: Common Objects in Context," *Computer Vision—ECCV 2014*, pp. 740–755, 2014. Accessed: 2 June 2019. doi: 10.1007/978-3-319-10602-1_48

17. M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *Int. J. Comput. Vis.*, 88(2):303–338, 2009. Accessed: 2 June 2019. doi: 10.1007/s11263-009-0275-4

18. A. Rosebrock, "Intersection over Union (IoU) for Object Detection—PyImageSearch," *PyImageSearch*, 2019. Accessed: 2 June 2019. [Online]. Available: https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

About the Authors



Oscar Estrada Torrejon is a recent graduate from the Motion Picture Science Program at the Rochester Institute of Technology (RIT), Rochester, NY. His interests include content development for digital platforms, post-production technology, and visual effects. His experience includes creating content marketing videos for his university, as well as managing broadcast equipment and post-production workflows on board oceanographic research vessels. Torrejon graduated *Summa Cum Laude* in 2019 and received the Outstanding Undergraduate Scholar Award and the Outstanding International Student Service Award for his service to his alma matter.

Nicholas Peretti recently graduated from the Rochester Institute of Technology (RIT), Rochester, NY, with a major in software engineering and a minor in imaging science. While at RIT, he worked for one year at Apple Inc., Cupertino, CA, beginning with the Photography Group and then spent six months with the Computer Vision and Machine Learning Team. He is currently the head of machine learning at Scythe Robotics based in Boulder, CO, which is creating a self-driving lawn mower for golf courses and municipalities. Peretti enjoys basketball and snowboarding, when not interacting with a computer.

Ricardo Figueroa is currently an associate professor, the program director for the Motion Picture Science Program, and an interim co-director of the School of Film and Animation, Rochester Institute of Technology (RIT), Rochester, NY. Figueroa joined RIT after working at Eastman Kodak, Rochester, for ten years, where he held positions as film and digital lab manager, digital/hybrid technologies regional director, digital imaging research engineer, and Kodak operating system manager. Figueroa holds a BSEE and an MSEE from the University of Puerto Rico, Mayagüez, Puerto Rico, and is currently pursuing a PhD in computing and information sciences at RIT. His research interests are in the areas of machine learning and deep learning for image processing applications. Outside of work, Figueroa is an avid triathlete, soccer player, and golfer. He is also active in various professional and civic organizations.

This article was the winner of the SMPTE 2019 Student Paper Award competition and was also presented at the SMPTE 2019 Annual Technical Conference & Exhibition, Los Angeles, CA, 21–24 October 2019.