Google AI Blog

The latest news from Google AI

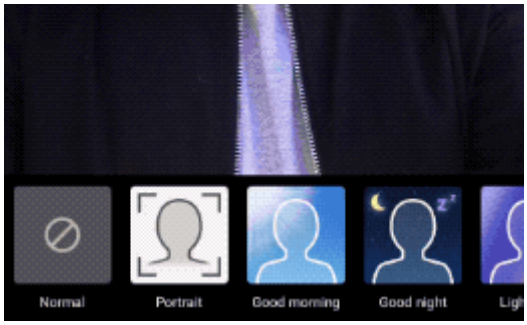
# Mobile Real-time Video Segmentation

Thursday, March 1, 2018

Valentin Bazarevsky and Andrei Tkachenka, Software Engineers, Google Research

Video segmentation is a widely used technique that enables movie directors and video content creators to separate the foreground of a scene from the background, and treat them as two different visual layers. By modifying or replacing the background, creators can convey a particular mood, transport themselves to a fun location or enhance the impact of the message. However, this operation has traditionally been performed as a time-consuming manual process (e.g. an artist rotoscoping every frame) or requires a studio environment with a green screen for real-time background removal (a technique referred to as chroma keying). In order to enable users to create this effect live in the viewfinder, we designed a new technique that is suitable for mobile phones.

Today, we are excited to bring precise, real-time, on-device mobile video segmentation to the YouTube app by integrating this technology into stories. Currently in limited beta, stories is YouTube's new lightweight video format, designed specifically for YouTube creators. Our new segmentation technology allows creators to replace and modify the background, effortlessly increasing videos' production value without specialized equipment.




Neural network video segmentation in YouTube stories.

To achieve this, we leverage machine learning to solve a semantic segmentation task using convolutional neural networks. In particular, we designed a network architecture and training procedure suitable for mobile phones focusing on the following requirements and constraints:

- A mobile solution should be lightweight and run at least 10-30 times faster than existing state-of-the-art photo segmentation models. For real time inference, such a model needs to provide results at 30 frames per second.
- A video model should leverage temporal redundancy (neighboring frames look similar) and exhibit temporal consistency (neighboring results should be similar)
- High quality segmentation results require high quality annotations.

### The Dataset

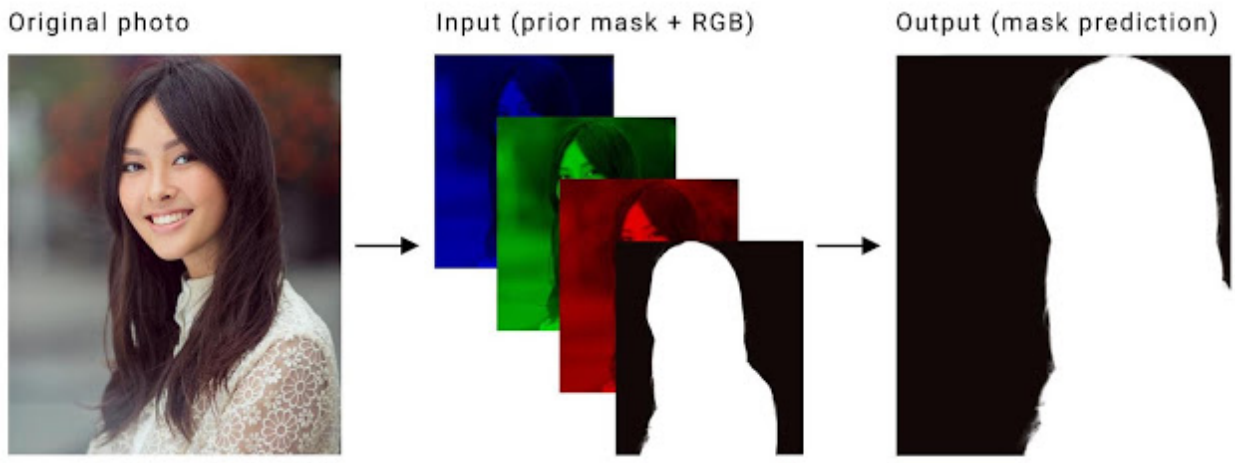
To provide high quality data for our machine learning pipeline, we annotated tens of thousands of images that captured a wide spectrum of foreground poses and background settings. Annotations consisted of pixel-accurate locations of foreground elements such as hair, glasses, neck, skin, lips, etc. and a general background label achieving a cross-validation result of 98% Intersection-Over-Union (IOU) of human annotator quality.


An example image from our dataset carefully annotated with nine labels - foreground elements are overlaid over the image.

### Network Input

Our specific segmentation task is to compute a binary mask separating foreground from background for every input frame (three channels, RGB) of the video. Achieving temporal

background for every input frame (three channels, RGB) of the video. Achieving temporal consistency of the computed masks across frames is key. Current methods that utilize LSTMs or GRUs to realize this are too computationally expensive for real-time applications on mobile phones. Instead we first pass the computed mask from the previous frame as a prior by concatenating it as a fourth channel to the current RGB input frame to achieve temporal consistency, as shown below:




The original frame (left) is separated in its three color channels and concatenated with the previous mask (middle). This is used as input to our neural network to predict the mask for the current frame (right).

### Training Procedure

In video segmentation we need to achieve frame-to-frame temporal continuity, while also accounting for temporal discontinuities such as people suddenly appearing in the field of view of the camera. To train our model to robustly handle those use cases, we transform the annotated ground truth of each photo in several ways and use it as a previous frame mask:

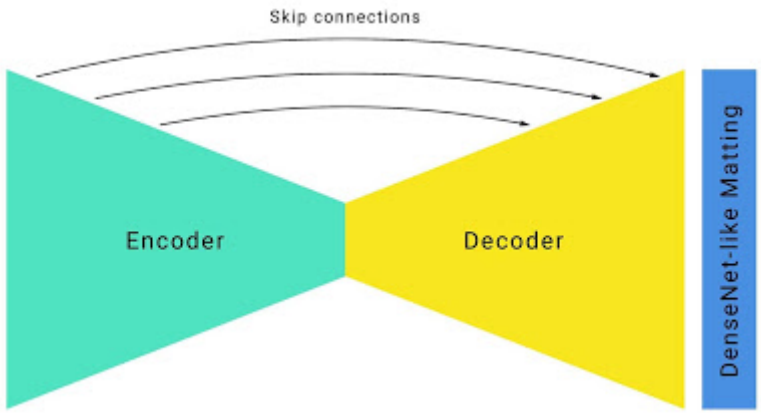
- **Empty previous mask** - Trains the network to work correctly for the first frame and new objects in scene. This emulates the case of someone appearing in the camera's frame.
- **Affine transformed ground truth mask** - Minor transformations train the network to propagate and adjust to the previous frame mask. Major transformations train the network to understand inadequate masks and discard them.
- **Transformed image** - We implement thin plate spline smoothing of the original image to emulate fast camera movements and rotations.

Our real-time video segmentation in action.

**Network Architecture**

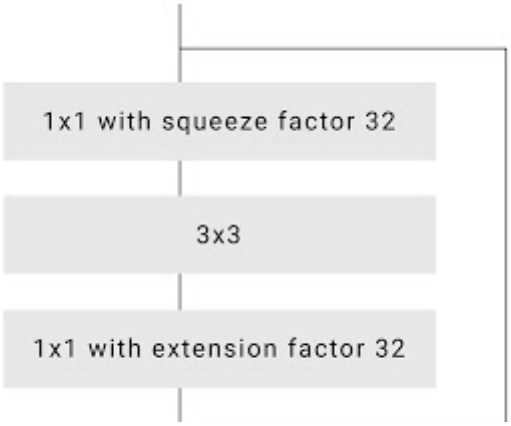
With that modified input/output, we build on a standard hourglass segmentation network architecture by adding the following improvements:

- We use big convolution kernels with large strides of four and above to detect object features on the high-resolution RGB input frame. Convolutions for layers with a small number of channels (as it is the case for the RGB input) are comparably cheap, so using big kernels here has almost no effect on the computational costs.
- For speed gains, we aggressively downsample using large strides combined with skip connections like U-Net to restore low-level features during upsampling. For our segmentation model this technique results in a significant improvement of 5% IOU compared to using no skip connections.




Hourglass segmentation network w/ skip connections.

- For even further speed gains, we optimized default ResNet bottlenecks. In the literature authors tend to squeeze channels in the middle of the network by a factor of four (e.g. reducing 256 channels to 64 by using 64 different convolution kernels). However, we noticed that one can squeeze much more aggressively by a factor of 16 or 32 without significant quality degradation.



ResNet bottleneck with large squeeze factor.

- To refine and improve the accuracy of edges, we add several DenseNet layers on top of our network in full resolution similar to neural matting. This technique improves overall model quality by a slight 0.5% IOU, however perceptual quality of segmentation improves significantly.

The end result of these modifications is that our network runs remarkably fast on mobile devices, achieving 100+ FPS on iPhone 7 and 40+ FPS on Pixel 2 with high accuracy (realizing 94.8% IOU on our validation dataset), delivering a variety of smooth running and responsive effects in YouTube stories.


Our immediate goal is to use the limited rollout in YouTube stories to test our technology on this first set of effects. As we improve and expand our segmentation technology to more labels, we plan to integrate it into Google's broader Augmented Reality services.

**Acknowledgements**
*A thank you to our team members who worked on the tech and this launch with us: Andrey Vakunov, Yury Kartynnik, Artsiom Ablavatski, Ivan Grishchenko, Matsvei Zhdanovich, Andrei Kulik, Camillo Lugaresi, John Kim, Ryan Bolyard, Wendy Huang, Michael Chang, Aaron La Lau, Willi Geiger, Tomer Margolin, John Nack and Matthias Grundmann.*