Face Parsing for Mobile AR Applications

Yongzhe Yan\textsuperscript{1,2}  Benjamin Bout\textsuperscript{2}  Anthony Berthelier\textsuperscript{1,2}  Thierry Chateau\textsuperscript{1}  Stefan Duffner\textsuperscript{3}  Christophe Garcia\textsuperscript{3}

\textsuperscript{1} Université Clermont Auvergne, France  \textsuperscript{2} Wisimage, France  \textsuperscript{3} Université de Lyon, CNRS, INSA Lyon, LIRIS, UMR5205, Lyon, France

yongzhe.yan@etu.uca.fr

1 Introduction

The goal of face parsing is to classify every pixel of an image into a category of facial components. Detecting different facial components is of great interest for a lot of augmented reality (AR) applications such as facial image beautification and facial image editing. For example, given the lip area, we can apply virtual lipstick-wearing effect on it by colorizing the region with proper colors. In this paper, we focus on designing an efficient face parsing methods by neural networks since lots of these applications are aimed at the mobile platforms such as iOS and Android.

Deep convolutional networks have been proved to be the leading methods for lots of computer vision tasks especially semantic segmentation\cite{1, 8}. Nonetheless, these methods are either time-consuming or over-sized due to the excessive amount of parameters and computation. Most of the semantic segmentation methods are designed for general complex scenes or street scenes. However, face parsing is a quite different task for the following reasons.

— Face parsing is usually done based on a ROI given by preliminary face detection compared to the general semantic segmentation which is generally performed on the entire image.
— Sharp boundaries are demanded for facial AR applications in order to render better visual effects.
— Facial components have more deformable variance but less position and size variance compared to semantic segmentation.

In this paper, we present a real-time AR face parsing demonstration on iPhone with an efficient deep convolutional neural network. Unlike the precedent face parsing methods,
we consider the adaption of neural networks on video in order to provide a fluid and temporally consistent rendering. The users are able to visualize the segmentation results by a mask which indicates different facial regions. A rendering result is shown in figure 1. More AR applications can be realized based on our results.

2 Related Work

A commonly shared consensus in deep semantic segmentation area is that there exists two kinds of mainstream methods[3], the spatial pyramid pooling (SPP) module based structures and encoder-decoder structures[1, 8]. Encoder-decoder structures adopt progressive upsampling with skip connection to reconstruct the object boundary as sharp as possible. Skip connections play an important role in the network structure so that the CNN can transfer the low-level detailed information to the output layers.

To ensure the best user experience in AR applications, short inference time and low latency are required. Some researchers proposed to use optical flow and reuse the feature map of the past frames if the static scene persists. Another way to accelerate the inference time is to use network acceleration techniques like quantification or pruning. Recent works such as Mobile-net[9] and Shuffle-net proposed to use depth-wise separable convolution to reduce computational complexity of CNN with almost the same performance.

CNN based methods[6] brought a huge advance to face parsing compared to exemplar based methods[10]. Liu et al.[6] proposed to use Conditional Random Field (CRF) to provide sharp facial component contours. Sharing the same motivation, another work[5] proposed to use spatially variant recurrent unit which enables the regional information propagation.

3 Mobile Face Parsing Demo Description

In this part, we present the design of our segmentation network, how we adapt it to the video as well as some implementation details.

3.1 Mobile Hourglass Network

We follow the design of the hourglass model in human pose estimation[7]. Our network structure is presented in Figure 2. The Hourglass model is a symmetric encoder-decoder fully convolutional network with a depth of 4, which means the encoder downsamples the input for 4 times and the decoder upsamples the feature map for 4 times to reconstruct the output. Each yellow block represents a network block which enables the CNN to learn the information flow at each stage and skip connection.

We drop several convolutional and max-pooling layers at the beginning of the network and augment the size of the output map to 256, which is the same size as the input image. This will make the boundaries of facial objects sharper but increase the computational complexity as well. In order to accelerate the inference, we replace all of the ResNet blocks in the hourglass network by MobileNet[9] inverted bottleneck block. Expansion factor $\tau$ is an important hyper parameter which indicates how many times the number of feature channels are expanded in its blocks. We chose the number of features $f$ as 32 and the expansion factor $\tau$ as 6 by finding the best compromise between the speed and segmentation quality. A comparison of the models using different $f$ and $\tau$ is shown in Tab. 1.

3.2 Video Adaption

Inspired by this blog[2], we implemented two strategies to adapt our model to video face parsing. For inference, we take the mask of the frame $t-1$ as input to the frame $t$ to stabilize the segmentation. Due to the lack of video dataset, we apply a randomly transformed mask as a fake previous mask during the training. According to our experiments, this will eliminate the random segmentation noise which is present without taking the previous frame mask as input.

We add four dense layers[4] with a growth rate of 8 at the end of the network for more robust rendering.

4 Experiments

We train our model on the Helen dataset[10] which contains 2330 images manually labeled in 11 classes including background, skin, hair nose, left/right eyebrows, left/right eyes, upper/bottom lips as well as inner-mouth. The models are trained on all of the labels except the hair because the hair annotations are not precise. The images are cropped with a margin of 30%-70% of bounding box size according to the facial landmark annotations.

We use the RMSprop as optimizer and softmax cross-
TABLE 1 – Quantitative evaluation on Helen dataset

<table>
<thead>
<tr>
<th>Model</th>
<th>Overall F-score</th>
<th>Num. of parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>SegNet[1]</td>
<td>92.90</td>
<td>29.45M</td>
</tr>
<tr>
<td>Mobile-Hourglass-f16-τ3</td>
<td>93.00</td>
<td>0.09M</td>
</tr>
<tr>
<td>Mobile-Hourglass-f16-τ6</td>
<td>93.08</td>
<td>0.12M</td>
</tr>
<tr>
<td>Mobile-Hourglass-f32-τ3</td>
<td>93.18</td>
<td>0.17M</td>
</tr>
<tr>
<td>Mobile-Hourglass-f32-τ6</td>
<td>93.55</td>
<td>0.27M</td>
</tr>
</tbody>
</table>

TABLE 2 – Run-time (in ms) profiling on iPhone X

<table>
<thead>
<tr>
<th>Model</th>
<th>FD</th>
<th>MI</th>
<th>Colorization</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-net[8]</td>
<td>7</td>
<td>203</td>
<td>54</td>
<td>269</td>
</tr>
<tr>
<td>Mobile-Hourglass-f16-τ3</td>
<td>8</td>
<td>77</td>
<td>18</td>
<td>106</td>
</tr>
<tr>
<td>Mobile-Hourglass-f16-τ6</td>
<td>7</td>
<td>75</td>
<td>18</td>
<td>103</td>
</tr>
<tr>
<td>Mobile-Hourglass-f32-τ3</td>
<td>7</td>
<td>76</td>
<td>18</td>
<td>104</td>
</tr>
<tr>
<td>Mobile-Hourglass-f32-τ6</td>
<td>7</td>
<td>78</td>
<td>18</td>
<td>106</td>
</tr>
<tr>
<td>Dense Mobile-Hourglass-f32-τ6</td>
<td>7</td>
<td>75</td>
<td>18</td>
<td>110</td>
</tr>
</tbody>
</table>

FD : Face Detection  MI : Model Inference

entropy function as loss function. We apply an initial learning rate of 0.0005 with decay of 0.1 for each 40 training epochs until total 190 epochs are finished. We use ONNX as an intermediate format to transform our Pytorch model to CoreML model, which is optimized for iOS devices. We adopt the Vision framework for face detection that is anterior to face parsing.

We compare our methods with several well-known segmentation networks[8, 1]. The results are measured in F1-score in Table1. The number of parameters are also listed aside to provide more information about the model size, which is critic for mobile platform.

We measure the runtime of different MobileNet block settings by changing the number of channels f and expansion factor τ. We also provide a profiling time analysis on the face detection, array transformation and colorization in Tab. 2.

5 Conclusion

We present a real-time encoder-decoder video face parsing mobile AR demonstration. Our method will draw interest because it is simple and interesting for everybody to test. More importantly,

— Face parsing is crucial for a lot of facial editing applications for example virtual make-up, hair dying, skin analysis, face morphing, reenactment etc.
— Our method is not only limited to facial AR applications but also interesting for fine-grained segmentation based AR applications.

Références
