LaTeX Semantic Segmentation task for OpenEDS dataset using Dual Model with Skip Connections

Adish Rao
PES University
contact.adishrao@gmail.com

Aniruddha Mysore
PES University
aniruddha.mysore@gmail.com

Poulami Sarkar
PES University
poulamisarkar101@gmail.com

Siddhanth Ajri
PES University
y2jsiddajri@gmail.com

Abishek Guragol
PES University
abhishekguragol@gmail.com

Rahul Suresh
PES University
rahs98@gmail.com

Gowri Srinivasa
PES University
Ring Road Campus, Bangalore
gsrinivasa@pes.edu

Abstract

*Many AR/VR applications require accurate and precise eye tracking solutions. This solution should work for all persons having different eye colour, and for different orientations of the eye. One means of achieving this is by semantic segmentation of the key regions of the human eye. Our research is based on the OpenEDS: Open Eye Dataset** paper, that presented a pixel-level annotated data set for key eye-regions: iris, pupil and sclera. We present an alternate solution to this problem that beats the baseline model obtained by the OpenEDS team. Our solution received a score of 0.93 on the leaderboard.*

1. Introduction

This is our submission for the OpenEDS challenge hosted by Facebook on the CloudCV platform. AR/VR applications require accurate and precise eye tracking solutions. In this paper we explore one method of achieving a low complexity model while still maintaining a high accuracy in the semantic segmentation of key regions of the human eye.

2. The OpenEDS dataset

The dataset used in this paper has been created by and released as a workshop challenge at ICCV by Facebook. It has been pre-split into 3 sets, consisting of 8916 train, 2403 validation and 1440 test images respectively. The labels have been included for the train and validation sets.

3. Methodology

3.1. Pre-processing

We first compress the given data from an image of size 640x400 into an image of size 160x96 in order to maintain the image ratio while still retaining an image of dimension $\mod 32$.

3.2. Pipelined encoder-decoder model with skip connections

We have used an encoder-decoder network based on the SegNet architecture to classify different key regions of the eye. The network classifies each pixel of an input image into one of four classes; pupil, iris, sclera and regions exterior to the eye. Our solution posits a two-fold implementation of encoder-decoder networks to classify each pixel of an input image. The first network is a base model that predicts 3 classes; the sclera, iris and outside regions. The second model predicts the pixels forming the pupil of the eye inside the iris using an input image of size 32x32, the general location of which is obtained from the output of the first model. This is done by first locating the initial row and final rows and columns of the image classified as the iris and then extracting the image of size 32x32 using the midpoint of the previously obtained rows and columns values. Both models use skip connections that are interlaced between the convolution layers. Results of experimentation with both additive as well as multiplicative skip connections showed that additive skip connections outperformed the multiplicative ones.

The final segmentation output is obtained by classifying

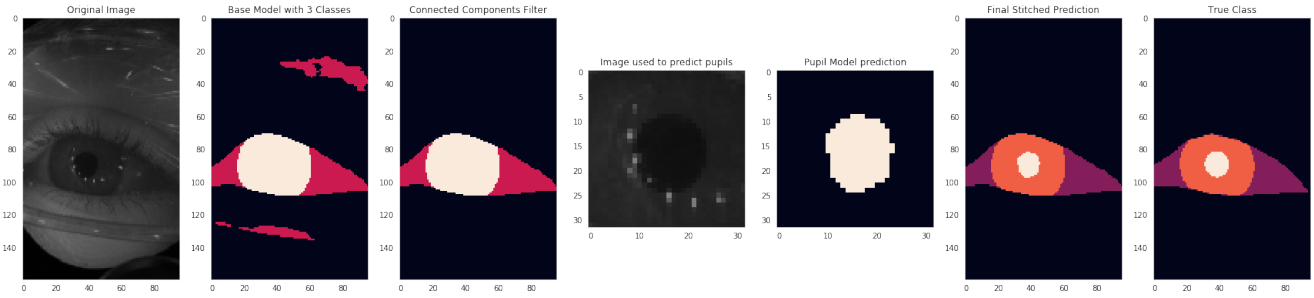

Figure 1: Base-model first predicts the sclera and iris then the pupils-model detects the pupil

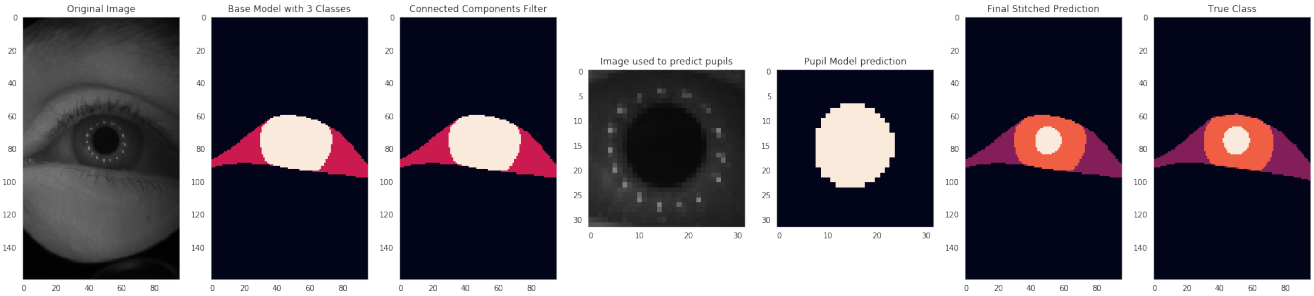

Figure 2: A good prediction

each pixel into 3 categories using the base model and identifying the pupils region using the pupil model.

3.3. The architecture

We have implemented an encoder decoder architecture, with 148,739 model parameters for the base model and 32,000 model parameters for the pupil model.

3.4. Post processing

As seen in Figure1. The predicted output of the stitched neural network produces a significant amount of extraneous detection. We have opted to use image processing techniques in order to detect and remove these erroneous detection. We have used connected components to find extraneous detection in the predicted output of the base model,and eliminate them based on their weighted area. before extracting the smaller image for predicting pupils.

References

Layer (Type)	Shape	Model Parameters
input_1 (InputLayer)	160, 96, 3	0
conv2d_1 (Conv2D)	160, 96, 32	896
max_pooling2d_1 (MaxPooling2D)	80, 48, 32	0
conv2d_2 (Conv2D)	80, 48, 64	18496
max_pooling2d_2 (MaxPooling2D)	40, 24, 64	0
conv2d_3 (Conv2D)	40, 24, 32	18464
conv2d_4 (Conv2D)	40, 24, 32	9248
conv2d_5 (Conv2D)	40, 24, 32	9248
add_2 (Add)	40, 24, 32	0
conv2d_6 (Conv2D)	40, 24, 32	9248
add_3 (Add)	40, 24, 32	0
up_sampling2d_1 (UpSampling2D)	80, 48, 32	0
concatenate_1 (Concatenate)	80, 48, 96	0
conv2d_7 (Conv2D)	80, 48, 64	55360
add_4 (Add)	80, 48, 64	0
add_5 (Add)	80, 48, 64	0
up_sampling2d_2 (UpSampling2D)	160, 96, 64	0
concatenate_2 (Concatenate)	160, 96, 96	0
conv2d_8 (Conv2D)	160, 96, 32	27680
add_6 (Add)	160, 96, 32	0
add_7 (Add)	160, 96, 32	0
conv2d_9 (Conv2D)	160, 96, 3	99
activation_1 (Activation)	160, 96, 3	0

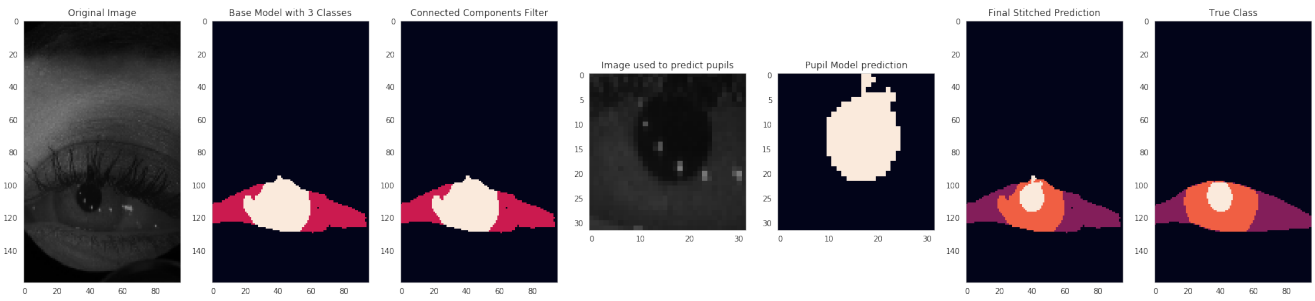

Figure 3: A bad prediction