Capstone Proposal

Omar Ezzat

June 21, 2017

Domain Background Computer Vision

Using the Laptop webcam to estimate the user gaze on the screen. This estimation can be used as an input source for any application. This problem is especially interesting because the eye movement is giving a lot of information about the user. We can use it for mouse movement, command sequences, eye tracking, UX validation, analytics.
This problem is particular interesting for me to improve my portfolio. It held the possibility to build a web application which is entertaining, scientific advanced and easy to use.

Similar project: link
Interesting article to teach an convnet to capture eye movements: link

Problem Statement

We want to map the user snapshot to a normalized plane (screen).
Let X be a set of images and $Y := [0, 1] \times [0, 1]$ the label. We need to find a function $f : X \mapsto Y$. This mapping will result to a point on the screen. The performance can be measured by using the SSE between the label and the estimate y'.
$E(y, y') := \frac{1}{2}((y_1 - y_1^{'})^2 + (y_2 - y_2^{'})^2)$

Datasets and Inputs

The dataset will be created by myself using a 720p webcam.
Example:

image	glasses	label_x	label_y
...	0	0	0.5
...	0	1	0.5
...	0	0.5	0.5
...	0	0.23	0.5

image: The raw image will have $1280 \times 720 \times 3$ bytes ≈ 2.7 Mb
glasses: if the user wears glasses this column will be 1
label_x: glaze on the screen in x direction [0,1]
label_y: glaze on the screen in y direction [0,1]

Solution Statement

Convolutional nets will be used to solve this problem. The system will need to learn first to focus on the eyes to reduce the image size, and focus afterwards on predicting the gaze. We also don't need to use 3 color channels, greyscale is enough.

Benchmark Model

For benchmarking we will use training time and prediction quality.
For all training we will use 2.2 GHz Intel Core i7, 16 GB RAM.
The quality will be measured using the SSE.

Evaluation Metrics

The evaluation metric will simply be the error of the prediction.
Using continuous labels we can use SSE to evaluation the error.

Project Design

Steps 1:
Create an easy to use webpage to capture user snapshot while looking at a specified point on the screen.
Step 2:
Collect data
Step 3:
Pre process the data. Convert it to greyscale. Randomize the set. Divide in training, test and validation set.
Step 4:
Optionally extract the eyes to reduce the image size.
Step 5:
Construct different models. Playing with convnets, layer size, dropout, pooling, regularization.
Step 6:
Evaluate the different models.
Step 7:
Decide which model is the best.
Step 7:

Implement the trained model into the webpage.
Step 8:
Visualize the estimate on the webpage.
Step 9:
Find an entertaining application to make use of the net.
(Maze, painting, simple game, mouse movement ...)
Step 10:
Upload the webpage.