ISTANBUL TECHNICAL UNIVERSITY, COMPUTER ENGINEERING

BLG521E Artificial Intelligence Project RAD Report

Anıl Öztürk

504181504

October 31, 2019

1 THE PROJECT

The aim of the project is to develop **a concept software that provides the most compatible advertisement according to the population distribution in front of a billboard**. The software tries to show the advertisements of the companies whose target is closest to the current audience by extracting the age and gender statistics of the audience comes from the camera feed on top of the billboard. The algorithm will be running on **Python3** and models will be created, trained and executed on **Keras** or **PyTorch**. **State-of-art object detectors** and **own estimators** will be used.

2 KEY ASPECTS

- The companies will send the advertising videos to the fictitious company of me by specifying which audience they are targeting.
- A billboard prototype with a camera on it will detect the faces of those passing by and try to predict age and gender.
- The system will generate a statistic of the values it is currently measured and predicted
- The advertisements offered by the companies which have the most similar statistical values with the measured ones will be published at that moment
- This layout will repeat periodically depending on the period of ad renewal

3 PROJECT REQUIREMENTS

- A robust face detector
- An age predictor
- A gender predictor
- A cost function to be used in selection of the next advertisement

4 PROJECT PLAN

- An image processing based face detector will be used or an a-NN network will be created and trained on open-source face datasets
- An age regressor network will be created and trained on an open-source dataset
- A gender predictor network will be created and trained on an open-source dataset
- A cost function will be created for the optimal advertisement selection

5 FACE DETECTION OPTIONS

5.1 HAAR CASCADE TEMPLATES

Haar Cascade based Face Detector was the state-of-the-art in Face Detection for many years since 2001. There has been many improvements in the recent years. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. It is well known for being able to detect faces and body parts in an image, but can be trained to identify almost any object. It's more source efficient compared to a deep neural network model. It has following steps;

- Haar Feature Selection
- Creating Integral Images
- Adaboost Training
- Cascading Classifiers

Advantages: Source efficient, simple, detecting capability on faces with different sizes
Disadvantages: Fails at non-frontal faces, has many false positives

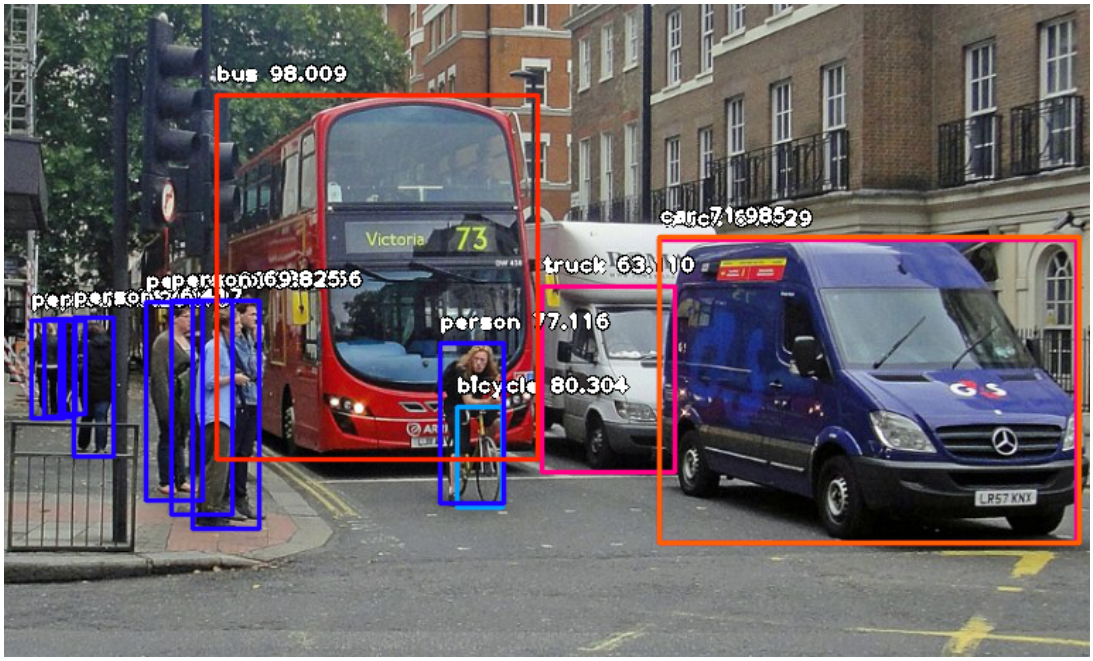
5.2 HoG (Histogram of Oriented Gradients)

This is a widely used face detection model, based on HoG features and SVM. The HOG descriptor technique counts occurrences of gradient orientation in localized portions of an image, or region of interest. It has the following steps;

- Divide the image into small connected regions and for each cell compute a histogram of gradient directions for the pixels within the cell.
- Discretize each cell into angular bins according to the gradient orientation. Each cell's pixel contributes weighted gradient to its corresponding angular bin.
- Groups of adjacent cells are considered as spatial regions (blocks). The grouping of cells into a block is the basis for grouping and normalization of histograms.
- Normalized group of histograms represents the block histogram. The set of these block histograms represents the descriptor.

Advantages: Very fast, robust performance at non-frontal faces, storage efficient ML model
Disadvantages: Fails at side faces, fails at small faces

5.3 Deep Learning Based Detectors



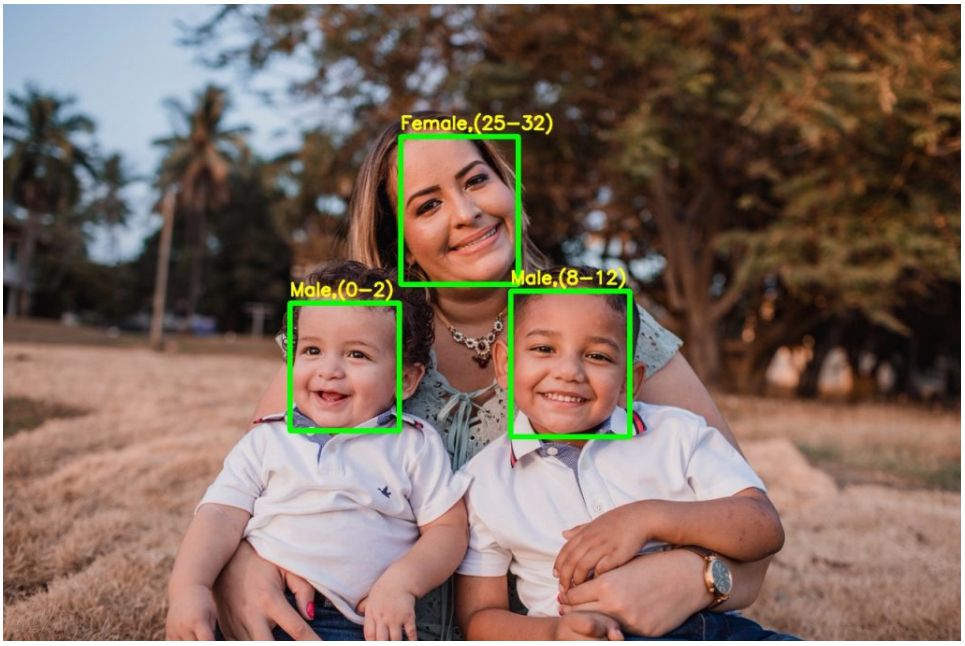
In recent years, so many feature extractor and object detector network have been published. These networks are running based on CNN architecture. They are running convolutional filters, pooling operations, dense layers on image matrixes and run a prediction layer at the end of the network. There are several object detector options;

- YOLOv3
- YOLOv2
- SSD
- R-CNN

5.4 My Choice

I will likely be using **YOLOv3** or **YOLOv2** because of their speed. Because face detection not one of that hard problems in these days. Accuracy won't differ too much between these alternatives, so the greatest pick would be the fastest one.

6 GENDER AND AGE PREDICTION



There is not a standard algorithm or model structure for predicting age and gender information of people. I need to construct my own model, train it from scratch. I will create and train a DNN models to predict these values. There should be two main structures;

- Gender Predictor
- Age Predictor

The face detector will give a bounding box that refers to a face. Since the gender and the age will be predicted from that cropped image, I need to use CNN-type network. There are several robust networks in the field. Some of the options are;

- VGG-16
- ResNet
- AlexNet

I will try each of them as the backbone of my own network. Then I will chose the one performs best.

6.1 GENDER PREDICTOR

Since it's a binary classification problem, I will likely be using a sigmoid at the end of my network. I will use the fraction points of the output to calculate the probability of predicted gender for that person.

6.2 AGE PREDICTOR

There are several ways to calculate final output for the age;

- Regressing a value
- Giving an interval
- Calculate an expected value from all of the age probabilities
- Calculate an expected value from all of the intervals

6.2.1 REGRESSING A VALUE

Regressing age is a hard thing to do with only looking at the face of a person. There are many dominant factors in any image. The factors like lightning, make-up, angle might affect the prediction in a huge scale. Predicting age intervals would bring more robust predictions.

6.2.2 GIVING AN INTERVAL

It may be a better solution to give an interval instead of exact values. Simple outputs may be like these;

- (0-8)
- (8-12)
- (12-16)
- (16-22)

6.2.3 CALCULATE AN EXPECTED VALUE FROM ALL OF THE AGE PROBABILITIES

We can set each age as a different class. Let's say we have predict ages between **0** and **100**. We have **101** classes. We can use a **softmax layer** with **101** outputs. If we multiply each class value with their own probability and sum their results up, we will get an expected age value for that image.

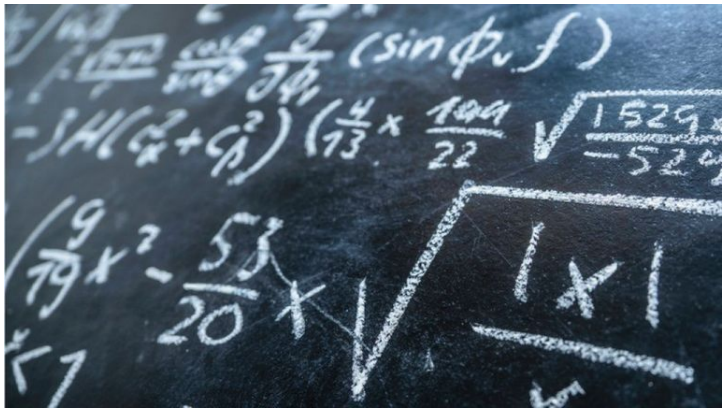
6.2.4 CALCULATE AN EXPECTED VALUE FROM ALL OF THE INTERVALS

We can set each age interval as a different class. Let's say we have predict ages between **0** and **100** in **10** intervals. We have **10** classes. We can use a **softmax layer** with **10** outputs. If we multiply each interval's mean value with their own probability and sum their results up, we will get an expected age value for that image.

6.2.5 MY CHOICE

Since the commercial marketing targets ages with intervals more than exact values. I will probably go with the **interval** option. But I might try the last two option, too.

7 SETTING UP THE COST FUNCTION



The cost function will be used for selecting the best-fit advertisement for the crowd at the given moment. The advertisement that gives the **minimum cost output** with the given crowd statistics will be selected as the next advertisement to be shown. It will likely include these variables;

- Women Ratio (R_W)
- Men Ratio (R_M)
- Women's Average Age (A_W)
- Men's Average Age (A_M)
- Targeted Men Age ($A_M^{'}$)
- Targeted Women Age ($A_W^{'}$)
- Importance of Women Customer (I_W)
- Importance of Men Customer (I_M)
- Total display count of the ad (C)

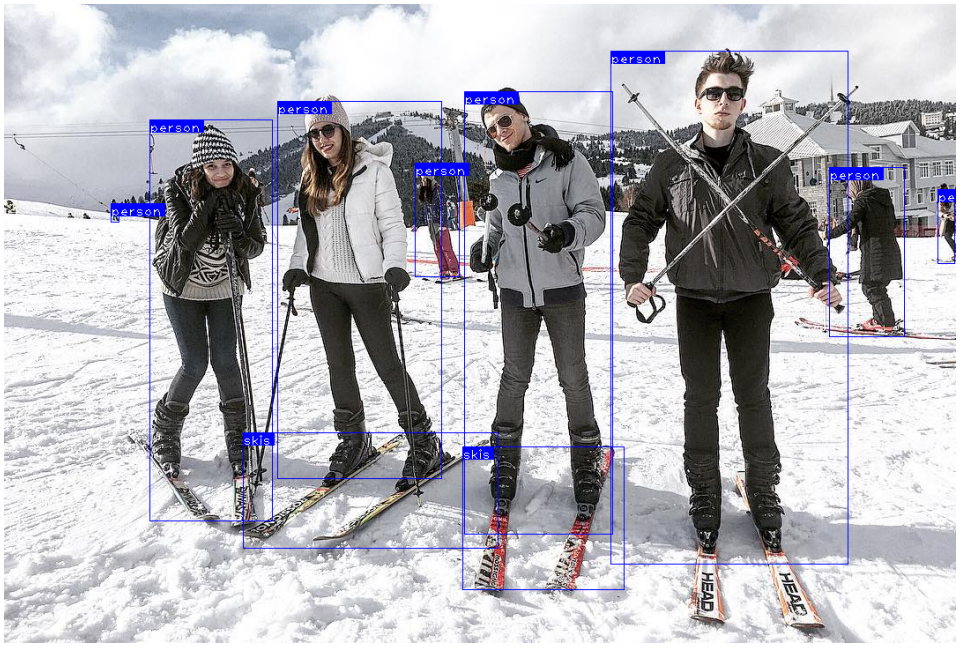
The cost function output E for the given ad x would be like;

$$E(x) = ln(C+e)\left(\frac{I_M}{R_M}MSE(A_M, A_M^{'}) + \frac{I_W}{R_W}MSE(A_W, A_W^{'})\right)$$

The cost function will be calculated for all available advertisements with minimum time cost with the help of **vectorization**.

8 Related Experiences

8.1 Creating a YOLOv3 Class from Scratch

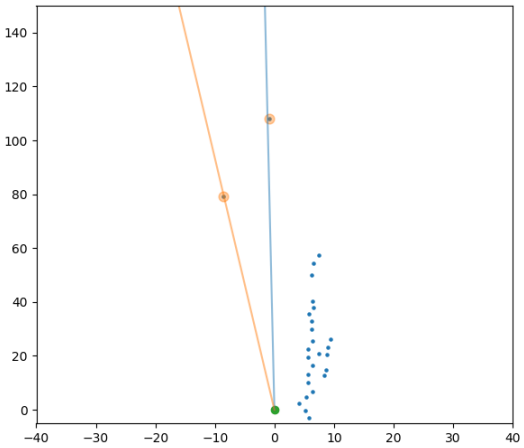


I have written YOLOv3 model from scratch in PyTorch with the help of open-source sites, trained on the COCO dataset and ran successfully. With that, I have learned;

- Using PyTorch
- Creating a whole object detector from scratch
- CNN data pre-processing
- Training a model
- Using and writing testing metrics

Later, I've implemented asynchronous image-processing based tracker on it, which grants extra performance up to 40%.

8.2 Training a Distance Estimator Network



I have created a distance estimator network that takes bounding boxes and predicts the vertical and horizontal distances of it. With that information, a line will be drawn in a 2D space which already has radar data from that moment. This way, we can match the prediction with the actual measure with the help of intersection points. With that, I have learned;

- Using Keras
- Creating a dataset for model training
- Running a live-analysis on a data
- Learning model cost-functions more theoritically

9 THE DATA SOURCES

Following datasets will be likely used for this project;

- https://drive.google.com/file/d/1Of_EVz-yHV7QVWQGihYfvtny9Ne8qXVz/view
- http://www.robots.ox.ac.uk/ vgg/data/vgg_face2/
- https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/
- https://talhassner.github.io/home/projects/Adience/Adience-data.html

10 THE PROJECT ENVIRONMENT

- i9 9900K Processor
- NVIDIA GTX1080TI GPU
- Python3.7
- Keras
- PyTorch
- Jupyter for Visualization