Project report

BIOMETRIC SYSTEMS

# Gender Detection using Keras pre-trained CNN network


Aakash Nandrajog

1734002

Prof. Maria De Marsico

## Abstract

In this project we are going to estimate the **Face detection using HAAR cascaded frontal face classifier**, **eye detection using HAAR cascade eye Classifier** and **Gender Classification problem** of the user with deep learning using **VGG16 model.**

Download the pre-trained model from below link:

https://s3.ap-south-1.amazonaws.com/arunponnusamy/pre-trained-weights/gender_detection.model

Our CNN neural network use **VGG-16 architecture** and are **pre-trained on ImageNet for image classification**. The proposed method is Deep Expectation (DEX) of apparent age. First, we detect the face in the test image and then extract the CNN predictions from an ensemble of 20 networks on the cropped face. The CNNs of DEX were finetuned on the crawled images and then on the provided images with apparent age annotations.


# 1.Introduction to the Model and the (DEX) Method:

### 1.Introduction

The goal of this work is to study the apparent age/gender estimation from single face image and by the mean of deep learning.

Our CNN used VGG-16 architecture and are pre- trained on ImageNet for Classification.

- **VGG -16 Architecture**

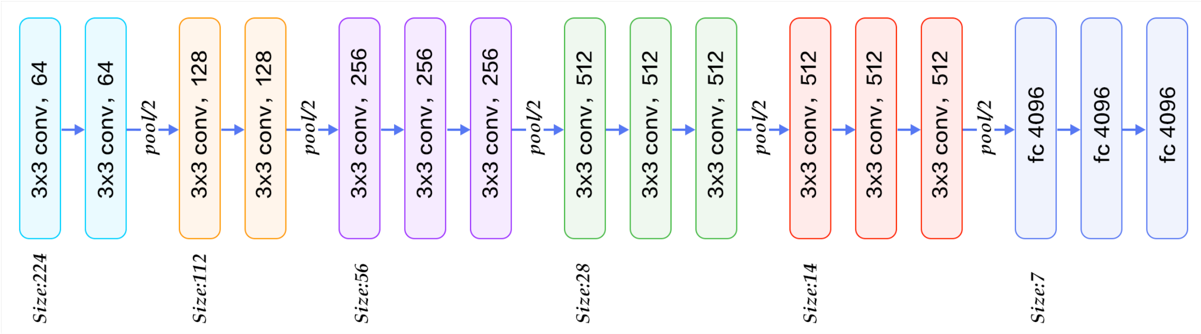
During training phase, the input of CNN is fixed-size 244 × 244 RGB image.

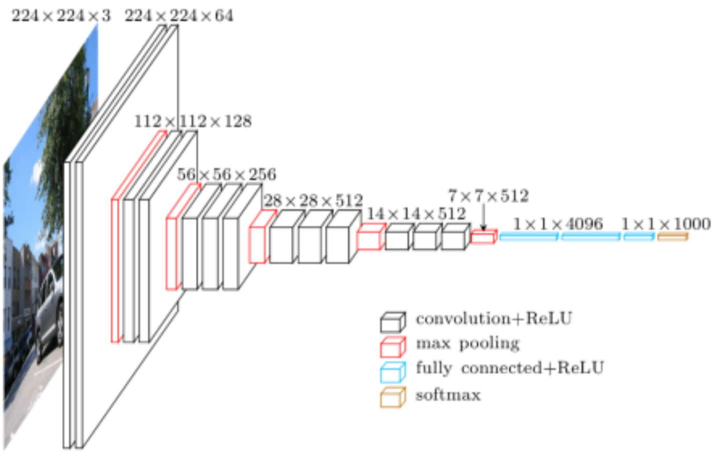
In **pre-processing** we subtract the mean RGB value, compute on the training set, from the each pixel.

The image is passed through a stack of convolutional layers

- Convolutions layers (used only 3*3 size)
- Max pooling layers (used only 2*2 size)
- Fully connected layers at end
- Total 16 layers

All the hidden layers are equipped with RELU function.
VGG network contain LRN normalization. This will increase memory consumption and computation time.

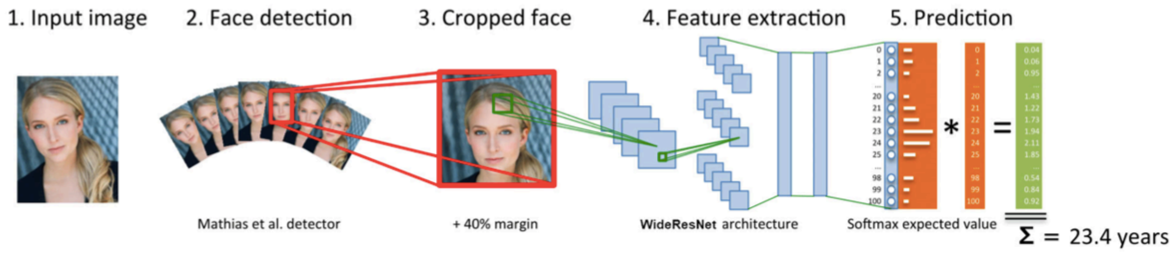



Age estimation is a regression problem and we train CNNs for classification where the age value is rounded into 101 years label. By posing the age regression as a deep classification problem followed by a SoftMax expected value refinement we improve significantly over direct regression training of CNNs.

As we Proposed (DEX) method. First we detect the face in the test image and then extract the CNN predictions from an ensemble of 20 networks on cropped face. DEX is pre- trained on ImageNet, finetuned on  IMDB-WIKI dataset, and then on LAP images.

## 2.Proposed Method (DEX):




### 2.1 Face Detection:

In Face Detection they used **Mathias *et al*** face detector to obtain the location of the face. Face detector not only run on the original image but also rotated version of images.

There are some cases where Detector couldn't find the face. In those case we just take the entire image. If the face covered the entire image, then we perform padding on the last pixel at the border. The resulting image is then squeezed to 256 × 256 pixels and used as an input to a deep convolutional network.

### 2.2 Face Apparent Age Estimation:

After detecting the face, the apparent age prediction is obtained by applying DNN.

#### 2.2.1 Deep learning with CNNs:

CNN starts from the **VGG-16 architecture** pretrained on the ImageNet dataset for image classification and then finetuned on IMDB dataset.

When training for classification the output layer is adapted 101 neurons corresponding to 0 to 100 natural numbers, which is used for age class labels.

#### 2.2.2 Expected Value:

Age estimation can seen as a regression problem. The larger the number of classes the smaller the error. In this case it is a one-dimensional Regression problem with age sample from ([0,100]).

Classification can be improved by increasing the number of classes which result better approximation. For improving the accuracy of the prediction, we computed a SoftMax expected value E as follows.

$$E(O) = \sum_{i=0}^{100} yioi$$

Where O ={0,1…100} is 101 dimensional output layer.

### 2.2.3 Ensemble of CNNs:

Here we split the dataset where 90% image goes for training and 10% goes for validation. The splits are chosen randomly for each age separately, i.e. the age distribution in the training is always the same. We do the augmentation after splitting the data into the training and validation set to ensure that there is no overlap between the two set.
Each network is trained and we pick the weights with the best performance on the validation set.
The final prediction is the average of the ensemble of 20 networks trained on slightly different splits of the data.

# 2. Experiment:

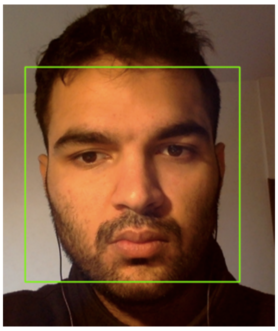
Here I talked about the implementation detail of the project.

### 2.1 Face detection and eyes detection :

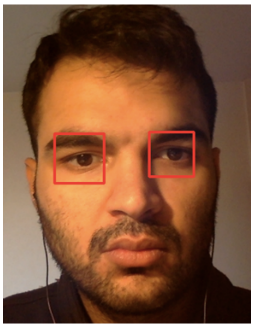
Before preceding into the face detection we need to know what Haar like features and cascade classifiers are. A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is used to categorize the subsections of an image. For e.g. in human face images, the region of eyes is darker than cheeks.
So when detecting an object in an image, the window of target size is moved over the input image and for each subsection of the image the Haar-like feature is calculated. This difference is then compared to the learned threshold that separates the object from non-objects.
The Haar-like features are weak learners or classifiers so large number of Haar-like features are necessary to classify images with higher accuracy. This is because one simple image can contain 180, 000+ Haar-like minute features. That's why in Haar-like features are organized into something called classifier cascade to form a strong learner or classifier. Due to the use of the integral image, the Haar-like features can be calculated in very higher speed than the traditional approach.

We can download the following two files from the OpenCV's GitHub repo:

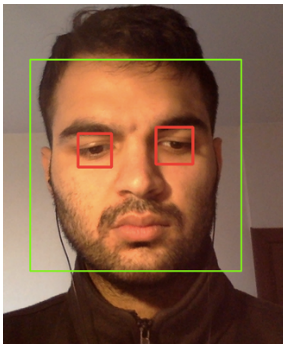
1) Haarcascade_frontalface_default.xml
2) Haarcascade_eye.xml



a.Face detection using HAAR



b. Eyes Detection using HAAR



c.Face and eyes detection using HAAR

## 2.2 Gender Detection :

Here we talk about Gender detection.

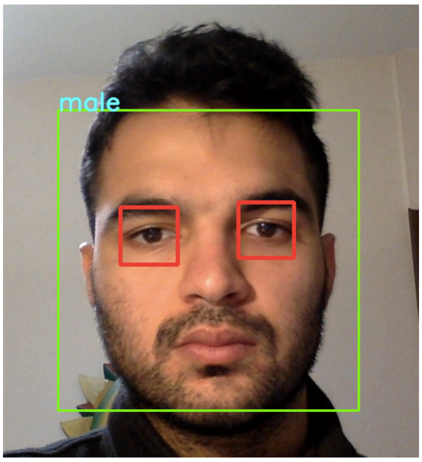
Download the pre-trained model from "https://s3.ap-south-1.amazonaws.com/arunponnusamy/pre-trained-weights/gender_detection.model " and load the model using keras load_model funtion.

Make sure to convert the image into grayscale and then use HAAR classifier to detect the face.

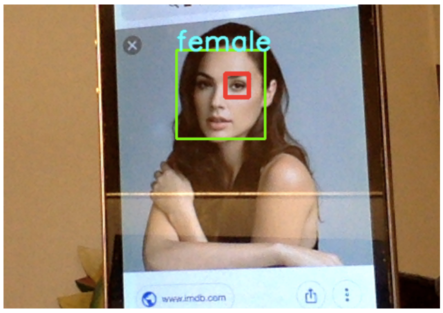
In pre-processing step first, we re-size the image, convert the crop image into float type, After, we use keras image_to_array function which convert the image into array and then we expand the dimension of image using numpy exp_dim function.

Next step will predict the gender of the user by using predict function.

## 2.3 Results:

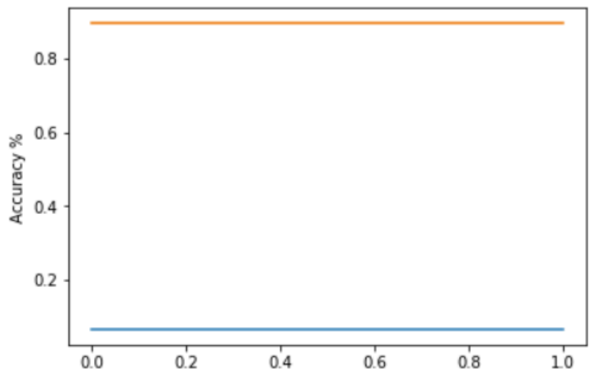



male



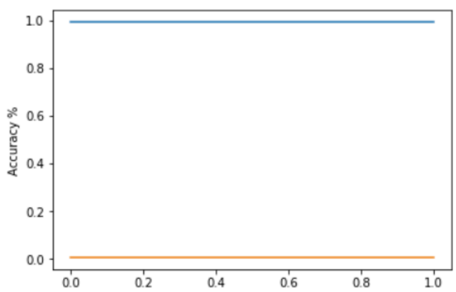

female

```
detect faces 1
0.06658005
female
```




**female accuracy graph**

```
detect faces 1
0.99311733
male
detect eyes 2
detect eyes 2
```




**Male accuracy graph**

**Conclusion:**

This Gender recognition classifier provides accurate information of the gender. We tackled the estimation of apparent gender in still face images. Our proposed Deep EXpectation (DEX) method uses convolutional neural networks (CNNs) with VGG-16 architecture pretrained on ImageNet. In addition, we crawled Internet face images with available age to create the largest such public dataset known to date and to pretrain our CNNs. Further, our CNNs are finetuned on apparent age labeled face images. Therefore, This Gender recognition classifier provides accurate information of the gender in real time.

**References:**


**[1]** https://www.vision.ee.ethz.ch/en/publications/papers/proceedings/eth_biwi_01229.pdf

**[2]** https://www.vision.ee.ethz.ch/en/publications/papers/articles/eth_biwi_01299.pdf