**Age and Gender Detection Project**

**By – L.Liketh**

The main aim of this project isto identify the gender of any person and classify his/her age.

First we train our machine with different data and then with the help of this data our machine tries to display the gender and age of a person.

We have used the [Adience dataset](https://talhassner.github.io/home/projects/Adience/Adience-data.html) for training the model.

**1. Gender Prediction**

We have framed the Gender Prediction as a classification problem. The output layer in the gender prediction network is of type softmax with 2 nodes indicating the two classes “Male” and “Female”.

**2. Age Prediction**

Ideally, Age Prediction should be approached as a Regression problem since we are expecting a real number as the output. However, estimating age accurately using regression is challenging. Even humans cannot accurately predict the age based on looking at a person. However, we have an idea of whether they are in their 20s or in their 30s. Because of this reason, it is wise to frame this problem as a classification problem where we try to estimate the age group the person is in. For example, age in the range of 0-2 is a single class, 4-6 is another class and so on.

The Adience dataset has 8 classes divided into the following age groups [(0 – 2), (4 – 6), (8 – 12), (15 – 20), (25 – 32), (38 – 43), (48 – 53), (60 – 100)]. Thus, the age prediction network has 8 nodes in the final softmax layer indicating the mentioned age ranges.

It should be kept in mind that Age prediction from a single image is not a very easy problem to solve as the **perceived age** depends on a lot of factors and people of the same age may look pretty different in various parts of the world.

The code can be divided into four parts:

1. **Detect Faces**

We use the DNN Face Detector for face detection. The model is only 2.7MB and is pretty fast even on the CPU. The face detection is done using the function getFaceBox.

1. **Detect Gender**

We then load the gender network into memory and pass the detected face through the network. The forward pass gives the probabilities or confidence of the two classes. We take the max of the two outputs and use it as the final gender prediction.

1. **Detect Age**

We load the age network and use the forward pass to get the output. Since the network architecture is similar to the Gender Network, we can take the max out of all the outputs to get the predicted age group.

1. **Display output**

We will display the output of the network on the input images and show them using the imshow function.

## Result

## We saw that the network is able to predict both Gender and Age to high level of accuracy.

**Observations**

Although the gender prediction network performs well, the age prediction network falls short of our expectation.

* The age groups 0-2, 4-6, 8-13 and 25-32 are predicted with relatively high accuracy. ( see the diagonal elements )
* The output is heavily biased towards the age group 25-32 ( see the row belonging to the age group 25-32 ). This means that it is very easy for the network to get confused between the ages 15 to 43. So, even if the actual age is between 15-20 or 38-43, there is a high chance that the predicted age will be 25-32. This is also evident from the Results section.

**Conclusion**

## Overall, the accuracy of the models is decent but can be improved further by using more data, data augmentation and better network architectures.

## One can also make use of a regression model instead of classification for Age Prediction if enough data is available.