Face Detection: Identify Gender, Age, Ethnicity and Emotion

Idea Submission

CT/DT Number: CT20182398813

Contestant Name: Ayush Thakur

College Name: Netaji Subhash Engineering College

Problem Statement

The problem states to detect faces in an image and predict the age, gender, ethnicity and emotion of each detected face.

This is a classic computer vision problem the solution of which demands to build a pipeline which can take an image as an input and give the mentioned attributes of the faces in the image.

Understanding

The following are my understanding about this problem statement:

1) This problem statement demands a deep learning architecture which can perform multi task and can provide multiple outputs based on a single image input. This will be a supervised learning based approach.

2) Since the input is an image which is a 2D data, Convolutional Neural Network is a proven architecture to go with. This will be used to extract the features from the image while maintaining the spatial representation of the same.

3) Here multitask correspond to:
    a) **Detection**: The face as an object is to be detected. Since we are unsure about the total number of faces in an image we have to perform classification(face or no face) on multiple crops of the image. Speaking about the architecture on a high level, it classifies the crop of the image as face or no face and perform a regression task to predict the bounding box of the detected face. Thus the dataset will be images along with 4 values for each face in that image as the bounding box.
    b) **Multi-output Classification**: The problem also demands to identify the age, gender, ethnicity and emotion of the detected face. We can either build four separate architectures to perform each prediction or can also build a unified architecture with four separate output layers. This can take in faces as the input and the label will be the age, gender, race, ethnicity and emotion of the face.
4) In my understanding the pipeline should have two units cascaded together with the former responsible with face detection and providing crops of the faces to the later unit. The 2nd unit should give the prediction about age, gender, ethnicity and emotion of the face.

Assumptions

The following general assumptions were made:

1) I considered hispanic and arabic ethnicity under ‘other’ class.
2) I grouped the ages into desired groups as mentioned by the provided dataset.
3) I have assumed that the data provided is to formulate the pipeline and validate the model. Thus have prepared my own data, labeled them and trained my architecture.

The following technical assumptions were made:

1) The size of input images can be a variable thus the architecture should account for that.
2) The provided data points were limited for any substantial training result. Thus either the model will overfit to the data if data augmentation is overloaded or it will underfit which is the most common case.

Solution Approach

The solution consists of two parts:

1. Joint face detection and alignment using multi task cascaded convolutional neural networks[4] and cropping.
2. Prediction of Age, Gender, Ethnicity and Emotion.

Face Detection and Cropping:

I used an open source implementation[7] of the joint face detection and alignment using multitask cascaded convolutional networks. The aim was to use this implementation as the backbone of my architecture and extend the functionalities for my pipeline.

The high level overview of this paper can be shown through figure 1. (Source: [4])

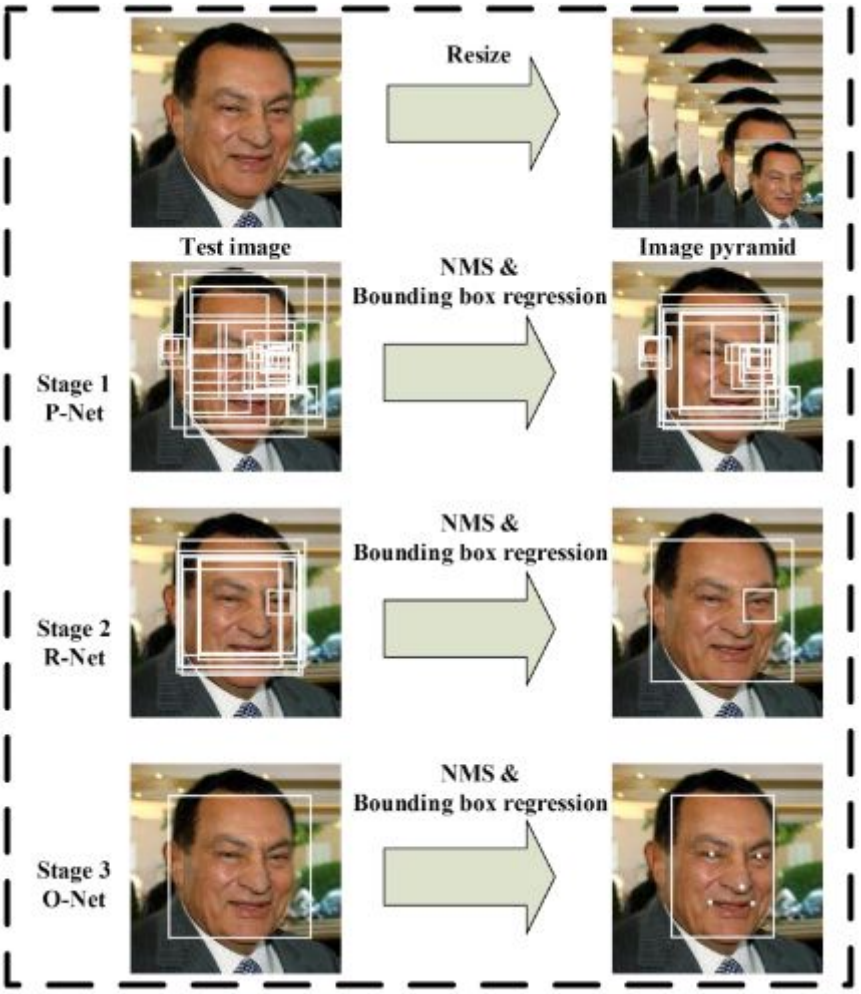


Figure 1

1) The following is the description of the face detection architecture:
    a) **Image Pyramid:** Given an image, it is resized to different scales to build an image pyramid, which is the input to the next three stages.
    b) **Stage 1:** Exploits Fully Convolutional Network which obtain the candidate windows and their bounding box regression vectors. Non-maximum suppression(NMS) is applied to merge highly overlapped candidates.
    c) **Stage 2:** The candidates from first stage is passed through another CNN which further rejects a large number of false candidates, performs calibration with bounding box regression, and NMS.
    d) **Stage 3:** Similar to stage 2 but describe the face in more details. In the original implementation the network will output five facial landmarks' positions. Since facial landmark is not the requirement of my pipeline I discarded this.
2) Since I have the bounding box of each detected faces a method in my extended implementation takes the image and the bounding box as input and

return tightly cropped faces. This will be input to the second unit of my pipeline.

Prediction of Age, Gender, Ethnicity and Emotion:

In order to determine the mentioned attributes of the given face, I iterated through various architectures and experimented with various parameters. I hereby mention the solution that I finally felt comfortable with.

Age Detection:

The problem statement demanded that for the given face the age should belong to either of the following age groups:

1) Age_below20
2) Age_20_30
3) Age_30_40
4) Age_40_50
5) Age_above_50

Since I have 5 age groups I decided to build a multi-class classifier with UTKFace dataset as an extension to the provided dataset.

The UTKFace dataset have age ranging from 1-116 but the distribution of the data is not uniform, in fact it's highly skewed. You can see the distribution in figure 2.

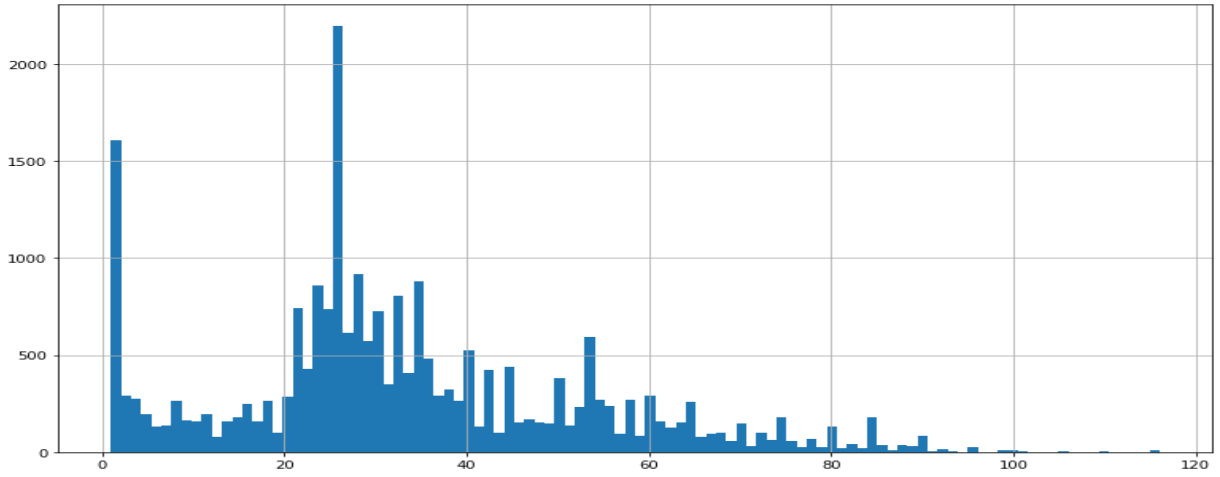


Figure 2

Thus to tackle this class imbalance I grouped together age groups as rationally as possible so that I can get somewhat uniform class distribution. This also converted continuous age values which otherwise should have been tackled with a regression based modelling into 11 classes and thus classification. The distribution of grouping can be seen in figure 3.

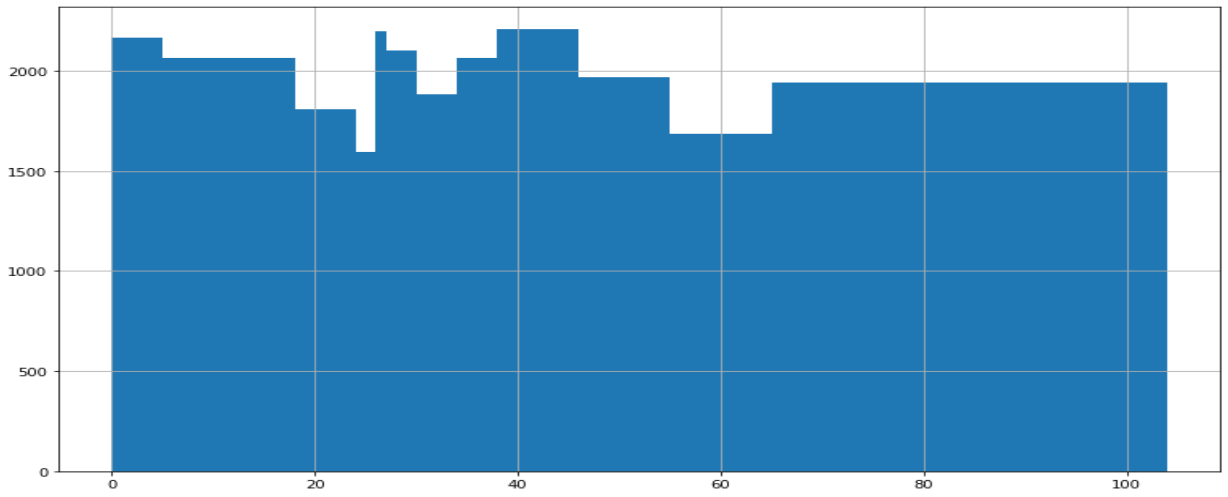


Figure 3

The prediction of the classifier belongs to either one of the 11 age groups. A simple algorithm group the outputs of the classifier into the required five groups.

Gender and Ethnicity Detection:

Using the same UTKFace dataset having gender and ethnicity as labels to the image along with age, I initially went for multi-output classifier. I built two separate variances of this classifier.

- One input and two conv blocks followed by FC layers.
- One conv block followed by two separate heads of FC layers.

But I wasn't satisfied with the result and thus decided to approach this differently. I implemented a multi-label classifier.

The classes in the gender label were:

- Male (0)
- Female (1)

The classes in the ethnicity label were:

- White (0)
- Black (1)
- Asian (2)
- Indian (3)
- Others (4) (Hispanic and Arabic)

In the multi-label classifier approach both the labels were grouped together, like male-white, male-indian, female-asian, etc. With 2 gender class and 5 ethnicity class I have 10 unique classes. Thus I built a multi-class classifier with 10 output nodes.

The data distribution was not uniform as well and can be seen in figure 4.

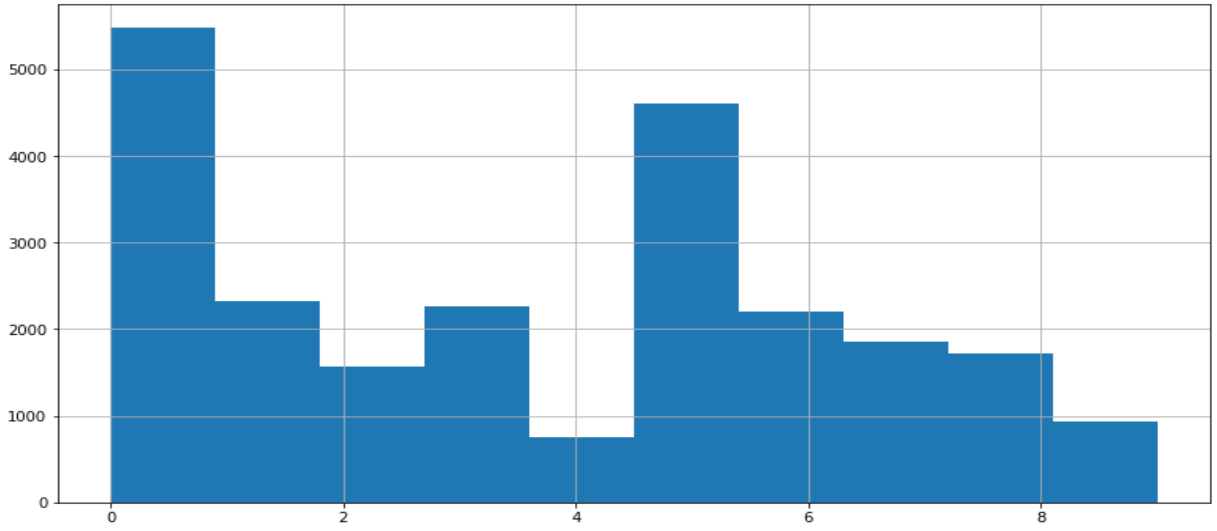


Figure 4

Two classes namely male-white and female-white were dominant classes in this distribution. Thus to tackle this I undersampled these two classes. This gave somewhat uniform distribution.

The prediction was then decoded to respective gender and ethnicity for the sake of the output of the pipeline.

Emotion Detection:

I used fer2018 dataset from Kaggle to train an emotion detector. I used the following emotion classes as per the dataset provided:

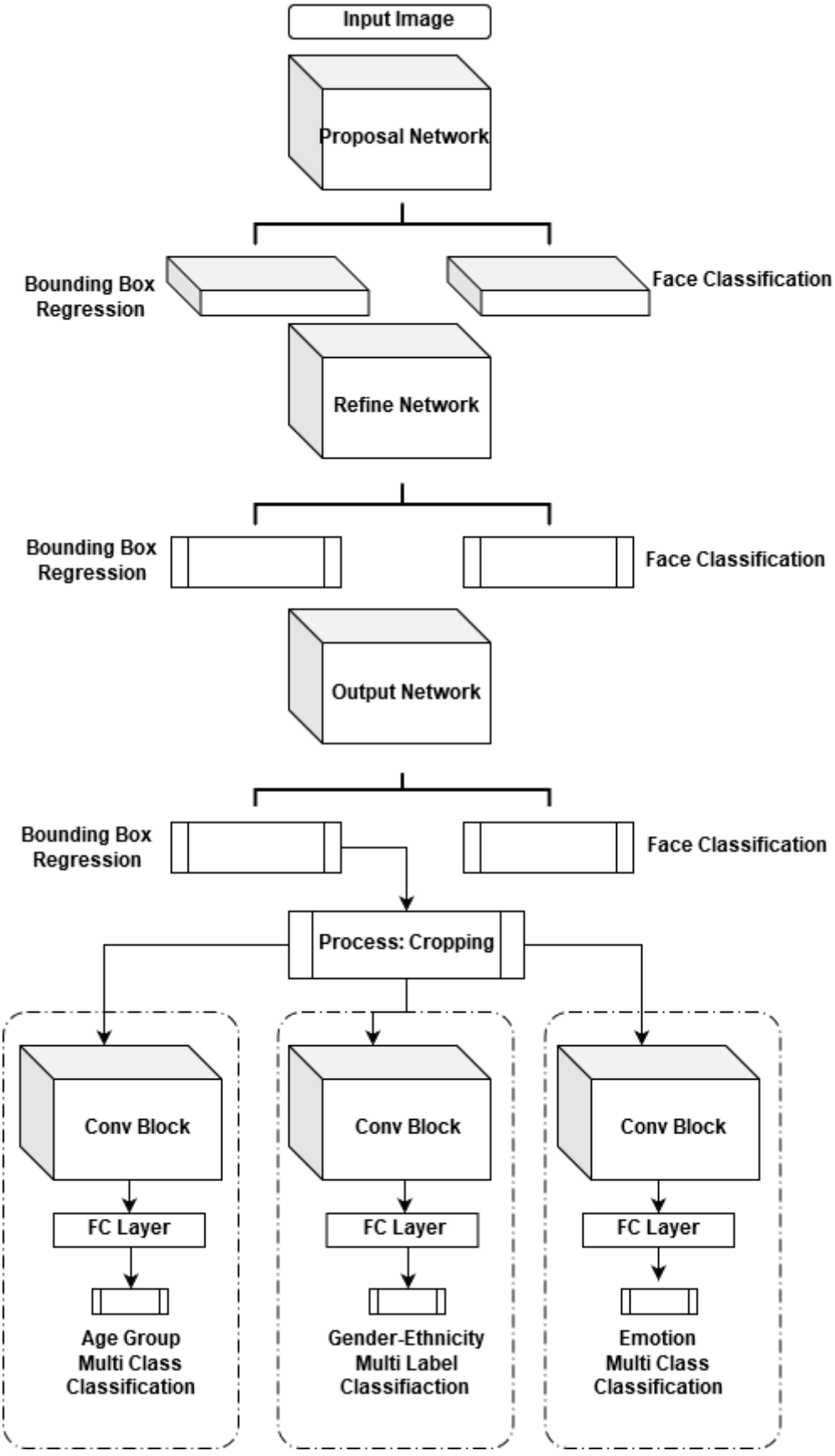
- Angry
- Happy
- Sad
- Neutral

The used dataset had more classes of emotion but they were discarded during data preprocessing.

The data class distribution was more or less balanced.

Putting it all together:

The block diagram shows the high level implementation of the pipeline:


Input Image
Proposal Network
Bounding Box Regression
Face Classification
Refine Network
Bounding Box Regression
Face Classification
Output Network
Bounding Box Regression
Face Classification
Process: Cropping
Conv Block
FC Layer
Age Group Multi Class Classification
Conv Block
FC Layer
Gender-Ethnicity Multi Label Classifiaction
Conv Block
FC Layer
Emotion Multi Class Classification

Implementation Framework

The following frameworks were used to implement this pipeline:

1) For Deep Learning Modelling and inference:
    a) Keras
    b) Tensorflow
2) For image processing:
    a) OpenCV
3) Programming Language:
    a) Python

The models were trained using Kaggle Kernels with GPU and 13Gb RAM.

Solution Submission

You can find my solution here: https://github.com/ayulockin/humAIn_FaceLess

References

Datasets:

The following datasets were used:

1) https://dataturks.com/projects/Mohan/Face%20Dataset%20With%20Emotion_Age_Ethnicity
2) https://susanqq.github.io/UTKFace/
3) https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data

Papers and Reading Materials:

The following papers and blogs were referred to:

4) Multi Task CNN: https://arxiv.org/ftp/arxiv/papers/1604/1604.02878.pdf
5) Multi-output Classification: https://www.pyimagesearch.com/2018/06/04/keras-multiple-outputs-and-multiple-losses/
6) Population age group: https://www.census.gov/prod/cen2010/briefs/c2010br-03.pdf

Open Source Frameworks:

The following open source frameworks were used to build Multi Task Cascaded CNN:

7) https://github.com/ipazc/mtcnn
8) https://github.com/davidsandberg/facenet

Thank You.