Project Description - Face Detection

Points: 30 points

Project Description

In this hands-on project, the goal is to build a face detection model which includes building a face detector to locate the position of a face in an image.

Dataset: WIDER Face Dataset

WIDER FACE dataset is a face detection benchmark dataset, of which images are selected from the publicly available WIDER dataset.
This data have 32,203 images and 393,703 faces are labeled with a high degree of variability in scale, pose and occlusion as depicted in the sample images.
In this project, we are using 409 images and around 1000 faces for ease of computation.

We will be using transfer learning on an already trained model to build our detector. We will perform transfer learning on the Mobile Net model which is already trained to perform object detection. We will need to train the last 6-7 layers and freeze the remaining layers to train the model for face detection. To be able to train the Mobile Net model for face detection, we will be using the WIDER FACE dataset which already has the bounding box data for various images with a single face and multiple faces. The output of the model is the bounding box data which gives the location of the face in an image. We learn to build a face detection model using Keras supported by Tensorflow.

Reference

Acknowledgment for the datasets. http://mmlab.ie.cuhk.edu.hk/projects/WIDERFace/
Mobile Net paper: https://arxiv.org/pdf/1704.04861.pdf

Overview

In this problem, we use "Transfer Learning" of an Object Detector model to detect any object according to the problem in hand.

Here, we are particularly interested in detecting faces in a given image. Below are the steps involved in the project.

- Load the dataset given in form .npy format.
  - “images.npy” contains details about the image and it’s bounding box, there is no separate CSV file for that
  - There is no separate train and test data given
- Create Features(images) and labels(mask) using that data.
- Load the pre-trained model and weights.
- Create a model using the above model.

- Define the Dice Coefficient and Loss function.
- Compile and fit the model.
- Evaluate the model.

Instructions for all the above steps are given in the notebook.

Project Support

You can clarify your queries related to the project by dropping a mail on Olympus

Happy Learning!