Replication of Original Work

The authors of RetinaFace have developed their work into a full fledged python library ‘Insight face’. The source code to this library can be found at <https://github.com/deepinsight/insightface>. In this library, the original code used for the RetinaFace paper can be found under the RetinaFace subfolder. The repository comes with a well laid out readme file which details steps to recreate the outputs obtained by the authors.

The repository is cloned on a Linux based machine. In this case, we have cloned the repository in google drive and used Colab to execute the code within. As Colab uses a Linux backend, this method offered the best compromise between speed and feasibility as none of us had access to a system with sufficient VRAM to execute the code.

The dataset to be tested is the WIDERFACE dataset which is available at <http://shuoyang1213.me/WIDERFACE/index.html>. WIDERFACE has 3 classes, Train,Val and Test. All three datasets are uploaded to the cloned repository and organized according to the instructions provided in the readme file. The data folder under the repository has the hierarchy:

Data/retinaface/

Train/

Images/

Label.txt

Val/

Images/

Label.txt

Test/

Images/

Label.txt

With the original data in place, python dependencies need to be installed. The packages needed for this project which are not pre-installed on Colab are, mxnet, opencv, easydict, Cython. These are installed using the commands:

!pip install mxnet-cu101

!pip install opencv-python

!pip install easydict

!pip install Cython

On successful installation of the dependencies, the make command is executed to setup the environment. Finally one of the pretrained models provided by the authors has to be downloaded and stored in the models folder. The model used in this testing is the RetinaFace-50 which uses ResNet50 as its backbone and can be downloaded from <https://www.dropbox.com/s/53ftnlarhyrpkg2/retinaface-R50.zip?dl=0>.

Once the above steps are complete, we can execute the test\_widerface.py file to test the RetinaFace-50 model on the WiderFace dataset. The output of this process is in form of .txt files, each file corresponding to an image in the dataset. The output file of an image contains the number of faces the model has identified in the image along with the top left (x,y) coordinates, breadth and length of the bounding box made for each face. These outputs are stored under the wout folder.

The labels produced above can be evaluated by the evaluation tools provided on the WiderFace website. These tools are a bunch of MATLAB executable .m files with instructions on how to obtain the outputs. The tools also contain the true labels of the images which are compared to the predicted labels produced by RetinaFace-50 model. Output of these eval tools is in form of Precision-Recall curves per image as shown below.

It is evident from the highlighted values in the graphs above that the retested model performs comparably to the original work of the authors.

Testing on Original Dataset

To test RetinaFace-50 on the original dataset created by us, we have modified the test.py file present in the original dataset. The original file took an image as input and produced an image with bounding boxes over faces as its output. We modified this file to produce the aforementioned image along with a .txt file with labels similar to that produced by the original program. All 105 images in the original images are run through this code and labels in form of text files are obtained for each of them.

These labels are then compared to the true labels annotated in the creation of the dataset. The precision and recall values for each image is noted and the resulting precision recall curve is shown below. The mean Average Precision for this dataset is…