# PES UNIVERSITY

## Practical Approach to Deep Learning using hardware Accelerator and software Framework

### ASSIGNMENT – 3

Student name: **SHREYAS KULKARNI**

SRN: **PES1201700450**

Branch: **ELECTRONICS AND COMMUNICATION**

# PART- I

**Problem statement:** Run the text detection demo and mention your observation.

1. Run the command prompt in admin mode.
2. Initialize OPENVINO environment.
   C:\Program Files (x86)\IntelSWTools\openvino_2019.3.379\bin>setupvars.bat
3. In model downloader, download the models: text-detection-0003, text-detection-0004 and text-recognition-0012.
   - C:\ProgramFiles(x86)\IntelSWTools\openvino_2019.3.379\deployment_tools\tools\model_downloader>python downloader.py --name text-detection-0003
   - C:\ProgramFiles(x86)\IntelSWTools\openvino_2019.3.379\deployment_tools\tools\model_downloader>python downloader.py --name text-detection-0004
   - C:\ProgramFiles(x86)\IntelSWTools\openvino_2019.3.379\deployment_tools\tools\model_downloader>python downloader.py --name text-recognition-0012
4. All these models are saved in a particular location.
   - C:\ProgramFiles(x86)\IntelSWTools\openvino_2019.3.379\deployment_tools\tools\model_downloader\intel
5. Now we must run the .exe file present in:
   - C:\Users\sonuk\Documents\Intel\OpenVINO\omz_demos_build\intel64\Release>text_detection_demo.exe -h
6. Final step, we give the image path and xml file path:
   - C:\Users\sonuk\Documents\Intel\OpenVINO\omz_demos_build\intel64\Release>text_detection_demo.exe
     -m_td
     "C:\ProgramFiles(x86)\IntelSWTools\openvino_2019.3.379\deployment_tools\tools\model_downloader\intel\text-detection-0003\FP32\text-detection-0003.xml"
     -m_tr
     "C:\ProgramFiles(x86)\IntelSWTools\openvino_2019.3.379\deployment_tools\tools\model_downloader\intel\text-recognition-0012\FP32\text-recognition-0012.xml"
     -i "C:\Users\sonuk\OneDrive\Desktop\download.jpeg" -dt image
7. Output: The text in the image is detected.

# Output and Observation

```
C:\Users\sonuk\Documents\Intel\OpenVINO\omz_demos_build\intel64\Release>text_detection_demo.exe -m_td "C:\Program Files (x86)\IntelSWTools\openvino_2019.3.379\deployment_tools\tools\model_downloader\intel\text-d
etection-0003\FP32\text-detection-0003.xml" -m_tr "C:\Program Files (x86)\IntelSWTools\openvino_2019.3.379\deployment_tools\tools\model_downloader\intel\text-recognition-0012\FP32\text-recognition-0012.xml" -i "
C:\Users\sonuk\OneDrive\Desktop\images.jpeg" -dt image
InferenceEngine:
        API version ............ 2.1
        Build .................. 32974
        Description ....... API
[ INFO ] Parsing input parameters
[ INFO ] Loading Inference Engine
[ INFO ] Device info:
        CPU
        MKLDNNPlugin version ......... 2.1
        Build ........... 32974

[ INFO ] Loading network files
[ INFO ] Reading input
[ INFO ] Starting inference
To close the application, press 'CTRL+C' here or switch to the output window and press ESC key
text detection model inference (ms) (fps): 361 2.77008
text detection postprocessing (ms) (fps): 34 29.4118

text recognition model inference (ms) (fps): 33.3333 30
text recognition postprocessing (ms) (fps): 0.033 30303

text crop (ms) (fps): 0.123333 8108.11
```

- The output showed that the letters that were visible clearly were detected properly. Some of the fonts were hard to detect as per what I observed. Also if the letters were very close to each other or overlapping a bit, the detection was not that accurate as obtained before.


Input image


Output image

# PART- II

**Problem statement:** Learn about mask RCNN and run the relevant executable present in omz_demos_build\intel64\Release folder.

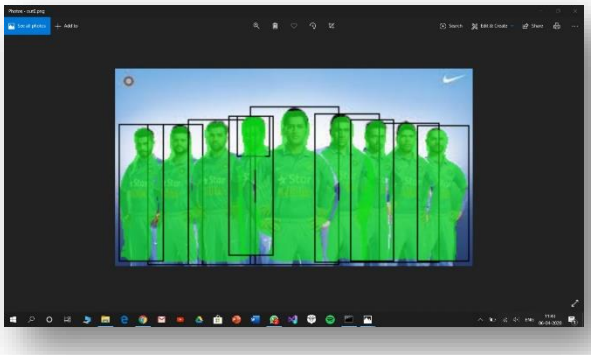
1. Check for the models using the command:
    - C:\ProgramFiles(x86)\IntelSWTools\openvino_2019.3.379\deployment_tools\tools\model_downloader>python downloader.py --print_all

    This gives the list of downloadable models in python.

2. Select the required model: mask_rcnn_inception_resnet_v2_atrous_coco
    - C:\ProgramFiles(x86)\IntelSWTools\openvino_2019.3.379\deployment_tools\tools\model_downloader>python downloader.py --name mask_rcnn_resnet101_atrous_coco
3. Create IR models. XML file and BIN file are generated.
    - C:\ProgramFiles(x86)\IntelSWTools\openvino_2019.3.379\deployment_tools\model_optimizer>python mo.py --input_model "C:\ProgramFiles(x86)\IntelSWTools\openvino_2019.3.379\deployment_tools\tools\model_downloader\public\mask_rcnn_resnet101_atrous_coco\mask_rcnn_resnet101_atrous_coco_2018_01_28\frozen_inference_graph.pb" --tensorflow_use_custom_operations_config "C:\ProgramFiles(x86)\IntelSWTools\openvino_2019.3.379\deployment_tools\model_optimizer\extensions\front\tf\mask_rcnn_support.json" --tensorflow_object_detection_api_pipeline_config "C:\Program Files (x86)\IntelSWTools\openvino_2019.3.379\deployment_tools\tools\model_downloader\public\mask_rcnn_resnet101_atrous_coco\mask_rcnn_resnet101_atrous_coco_2018_01_28\pipeline.config" --reverse_input_channels
4. Give the required frozen.xml file path and image path. The output is stored in Release folder.
    - C:\Users\sonuk\Documents\Intel\OpenVINO\omz_demos_build\intel64\Release>mask_rcnn_demo.exe -m "C:\ProgramFiles(x86)\IntelSWTools\openvino_2019.3.379\deployment_tools\model_optimizer\frozen_inference_graph.xml" -i "C:\Users\sonuk\OneDrive\Desktop\teamindia.png"

# Output and Observation

Input image



Output

```
C:\Users\sonuk\Documents\Intel\OpenVINO\omz_demos_build\intel64\Release>mask_rcnn_demo.exe -m "C:\Program Files (x86)\IntelSWTools\openvino_2019.3.379\deployment_tools\model_optimizer\frozen_inference_graph.xml"
 -i "C:\Users\sonuk\OneDrive\Desktop\teamindia.png"
InferenceEngine:
        API version ............ 2.1
        Build .................. 32974
        Description ....... API
[ INFO ] Parsing input parameters
[ INFO ] Files were added: 1
[ INFO ]     C:\Users\sonuk\OneDrive\Desktop\teamindia.png
[ INFO ] Loading Inference Engine
[ INFO ] Device info:
        CPU
        MKLDNNPlugin version ......... 2.1
        Build ........... 32974
[ INFO ] Loading network files
[ INFO ] Preparing input blobs
[ INFO ] Network batch size is 1
[ INFO ] Prepare image C:\Users\sonuk\OneDrive\Desktop\teamindia.png
[ WARNING ] Image is resized from (300, 168) to (800, 800)
[ INFO ] Preparing output blobs
[ INFO ] Loading model to the device
[ INFO ] Create infer request
[ INFO ] Setting input data to the blobs
[ INFO ] Start inference
[ INFO ] Processing output blobs
[ INFO ] Detected class 1 with probability 0.991511 from batch 0: [113.904, 32.3047], [189.386, 168]
[ INFO ] Detected class 1 with probability 0.973551 from batch 0: [221.123, 46.8688], [272.929, 166.289]
[ INFO ] Detected class 1 with probability 0.958398 from batch 0: [167.091, 38.2315], [222.599, 165.526]
[ INFO ] Detected class 1 with probability 0.950797 from batch 0: [27.1718, 47.7647], [74.6631, 168]
[ INFO ] Detected class 1 with probability 0.949229 from batch 0: [253.655, 48.2337], [297.761, 165.126]
[ INFO ] Detected class 1 with probability 0.94637 from batch 0: [3.66342, 47.6115], [42.547, 165.148]
[ INFO ] Detected class 1 with probability 0.936749 from batch 0: [61.8734, 43.3375], [119.326, 168]
[ INFO ] Detected class 1 with probability 0.569143 from batch 0: [102.359, 40.7499], [130.62, 76.6233]
[ INFO ] Detected class 1 with probability 0.560972 from batch 0: [95.0607, 40.1203], [133.138, 159.411]
[ INFO ] Detected class 1 with probability 0.342357 from batch 0: [197.654, 43.5558], [235.075, 165.085]
[ INFO ] Image out0.png created!
[ INFO ] Execution successful

[ INFO ] This demo is an API example, for any performance measurements please use the dedicated benchmark_app tool from the openVINO toolkit
```

- There is a boundary box created for each person detected.
- The correct head count is obtained even if though the objects are overlapping each other.

Initially we we're using faster_rcnn_resnet101_atrous_coco. But it gave BLOB error. Because faster RCNN doesn't have masking property. So we're asked to change the model from **faster RCNN** to **mask RCNN.** Also a command called --*transformations_config* didn't work for many of us. So we had to use --*tensorflow_use_custom_operations_config*. Then on it worked peacefully.