MIDS W251 Spring 2020

Section: Monday 6:30pm Suhas Gupta

HW7 – Neural face detection pipeline

1. The homework solution was implemented in a similar architecture as HW3 with the openCV face detector (used in HW3) replaced by the **Tensor Flow Face Detector** [1]. Total 5 docker containers were created with custom built images to create the modular architecture shown in Fig1.

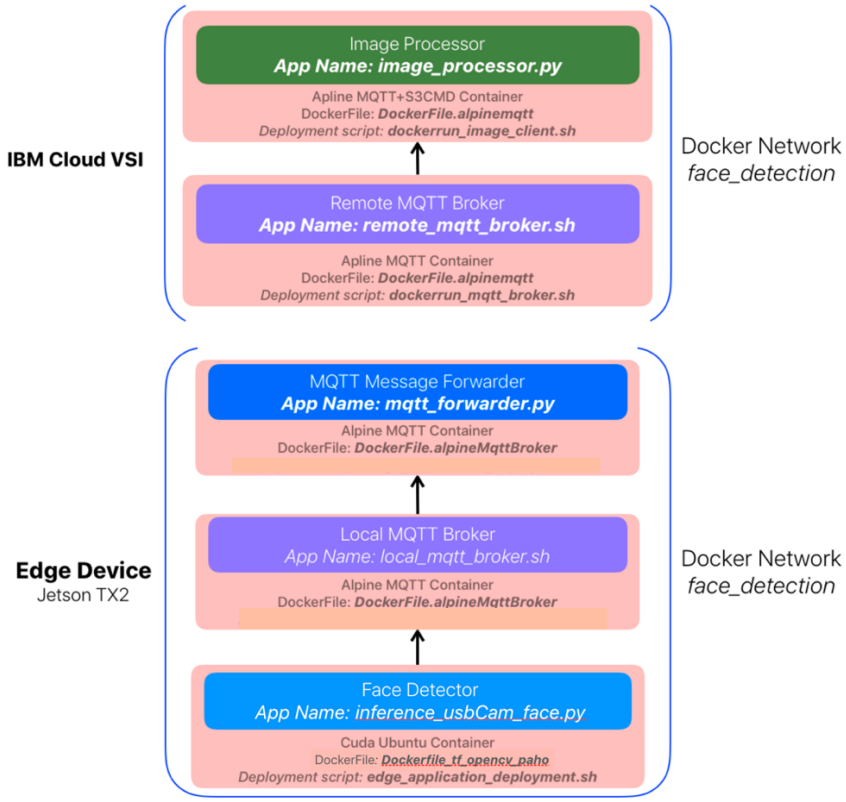
   - The custom images were built from the Docker files that are noted with the Docker keyword in Figure 1 for each container
   - *Docker-compose* was used to setup a swarm of services to deploy the face detection application. The *docker-compose.yml* file contains the deployment parameters.
   - The code implemented in this homework solution is available on the following GitHub repo:

   GitHub Repo Path: https://github.com/suhasgupta791/v2/tree/master/week07/hw
   Repo Clone URL: https://github.com/suhasgupta791/v2.git

   - The local (Jetson TX2) application scripts are located at: https://github.com/suhasgupta791/v2/tree/master/week07/hw/jetson_local_scripts
   - The remote (IBM VSI) application scripts are located at: https://github.com/suhasgupta791/v2/tree/master/week07/hw/ibm_vsi_remote_scripts

   - The images received at the remote cloud host are stored in the following object storage:

   Bucket Name: w251-suhasgupta-hw07
   host_base = s3.us-south.cloud-object-storage.appdomain.cloud
   host_bucket = %(w251-suhasgupta-hw07)s.s3.us-south.cloud-object-storage.appdomain.cloud



2. Face detection solution

a. A MobileNet SSD (single shot multibox detector) based face detector with pretrained model provided, powered by TensorFlow object detection api, trained by WIDERFACE dataset. It achieves an accuracy of up to **98%** for detecting human faces.
b. It does achieve very high accuracy on human faces in empirical tests.
    a. However, a major limitation of the model seems to be coming from using the “eyes” as a discerning feature of a face. While closing the eyes or covering them, the network is unable to detect the human face.
    b. From this empirical result, we can conclude that the training dataset did not have any images of human faces with closed or covered eyes.
    c. Such a model would not be preferable to use in a production system where there is a high probability of the camera not being able to detect human eyes with enough resolution to result in proper face detection.
c. It achieves frame rates of up to 50 frames/second. The bottleneck is the video rendering time of the hardware used (camera, display port link etc.)
d. TensorFlow based detector is better in quality (higher frame rates, higher accuracy) when compared to the OpenCV implementation. It is easy to improve its accuracy with a wider training dataset. The inference is fast and high frame rates can be achieved compared to the OpenCV implementation.