**Technical Task**

# Continuum Computing for Smart IoT Applications

## Supervisor:

## Prof. Josef Spillner


## Malihe Hariri

Tell: +989155894332

Email: malihehariri1993@gnail.com


February 2023

## 1. Application Area

In the task that was presented, all people who enter or leave public transportation should be recognized by image processing algorithms. In this case, there are a lot of image-processing APIs, SDKs, and software, like Vision Al of Google, OpenCV, and so on, that we can deploy to detect and track people in the ROI(Region of Interest) of the Camera. In Image Processing, an ROI is considered that camera can cover to detect and track people to count the number of entrance and exit from it. There is an important note that detecting algorithms considered people as a blob which means a group of people who enter or leave together and are stuck together consider a blob and this blob later must be processed by blob analysis algorithms to count or estimate the number of people on that blob.

In the following phase, this processed data should be processed by an API linked to a database and the number of people in each vehicle should be stored so that each operator is aware of their own number of passengers as well as the total fleet, including all numbers of passengers in each type of vehicle. For example, bus operators know the number of passengers in each region, and each region knows the number of passengers in buses that serve services in the specific region. These values later show up on an interface. The hierarchy structure of fleet operators is presented in Figure 1.

## 2. Motivation

People counter is one of the predominant subjects that could be used to control the commuting the individuals in the city especially in megacities to expand public transportation. In this way the council can observe how much public vehicles are used

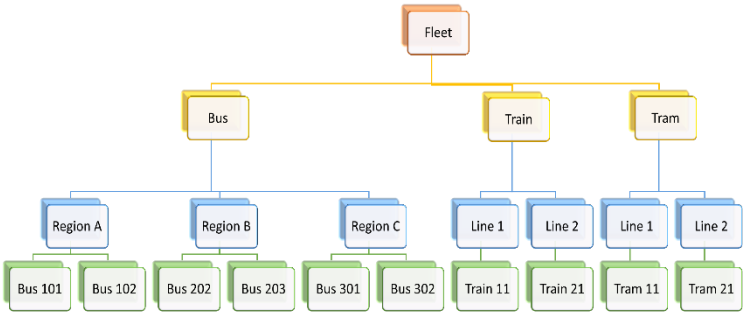


*Figure 1: fleet hierarchy*

by people, and what kinds of vehicles are commonly used and this can assist them to extend the infrastructure of specific transportation in the city. For example, if investigations prove that most people use the tram instead of trains and buses, the government can invest more expenses to extend this type of transportation. Furthermore, by investigating the movement tendencies of people council can control overcrowding. Besides this, movement tendencies advertising companies can install their banner in that way

## 3. System Design

For Implementing these subtasks, I used Python programming including the Paho library.

In Subtask 1, instead of processing a video by image processing algorithms, I decided to produce unreal data as .csv files which are located in the project folders with a similar name. so that, I consider three different vehicles including a bus, train, and tram that all of which have cameras in each vehicle in front of each door. I considered a different number of cameras in relation to the vehicle's

door, for example for the bus I considered three cameras in front of three doors of it, 10 cameras for the train considering 10 doors for it, and 6 cameras for the tram including 6 doors for it. These files include a unique ID, Round, camId, and Direction. I considered Round to make them real data which the algorithm sends every 10 minutes to API. Direction has 4 different directions for individuals who enter or leave vehicles including Up-Left, Up-Right, Down-Left, and Down-Right which I presented in .csv files by UL, UR, DL, and DR characters. There is a note that these values presented just a direction not the number of people and it is a blob of people, but for more simplicity, in the API processing phase each of these is considered one individual who enters or leaves a specific vehicle.

There is two protocol to send data, a standard IoT protocol MQTT is suitable for high-scaling implementation. Here I considered HTTP protocol to send my request related to my data model by POST method to Flask API to process it as a JSON payload. For this reason after producing the data model, I convert my CSV files to JSON files and send each row to the request library and post method by multi-threading programming. The Flask API receives the responses and processes them to recognize if is an entering people or leaving. Based on this, the MySQL database is updated by the queries in API.

## 4. Results

It is better that on a large scale application, we develop by Kubernetes engine or containerized it. In that way, when the requests get more that the threshold that we specified we can increase the number of pods or containers to scale it to support and process more requests in a specific time. Unfortunately, due to sanctions, my Docker desktop engine could not start by many times installing and testing many methods so I could not synchronize it with my VsCode. Software. Thus, I could not measure the scalability of my program.