APPLICATIONS FOR SMART AIRPORT PEOPLE DENSITY DETECTION INDIVIDUAL FINAL REPORT

Supervisor:

Dr. Chim T W

Team Members:

Yuen Cheuk Heng 3035553587

Mak Chak Wing 3035564732

(Prepared by) Siu King Hei 3035566405

Table of Contents

Table of Figures

Table of Tables

List of Abbreviations

Abbreviations & Acronyms	Definition
AAHK	Airport Authority Hong Kong
BBox	Bounding Box
CCTV	Closed-circuit television
CI/CD	Continuous Integration/Continuous Delivery
CNN	Convolutional Neural Network
HKIA	Hong Kong International Airport
HKU	University of Hong Kong
KLT	Kanade-Lucas-Tomasi
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MCNN	Multi-Column Convolutional Neural Network
ML	Machine Learning
MLOps	Machine Learning Operations
MSE	Mean Squared Error
NMS	Non-maximum Suppression
RNN	Recurrent Neural Network
RTSP	Real Time Stream Protocol
SDK	Software Development Kit
VGG	Visual Geometry Group
VRAM	Video RAM

1. Project Background

Passenger movement intelligence has been a major interest of the Airport Authority Hong Kong (AAHK) as it advances with its vision of transforming the Hong Kong International Airport (HKIA) into a Smart Airport. With the aim of providing quality passenger services, improving operational efficiency, and guiding future operational and management strategies, big data intelligence stands as a key piece of technology to realize this goal.

1.1. Big Data Intelligence

Big data intelligence is one of the five key enabling technologies outlined in the Smart Airport Vision as a framework for realizing the gains of passenger movement intelligence [1]. Through data collection, artificial intelligence and machine learning algorithms, instrumental performance indicators can be extracted which helps uncover hidden patterns and trends. One of the notable efforts implemented by AAHK is the usage of patrol robots that operates autonomously and monitors Wi-Fi signal strength around the terminal [2]. Following the spirit of this endeavour, each and every part of the HKIA operation should be further analysed and optimized to different extent. Given the great complexity of HKIA operation, big data analytics is the only practical approach, allowing optimizations to be done continuously and effectively as the process runs autonomously with little human intervention.

Being one of the busiest airports in the world, each year the Hong Kong International Airport services over 71 million passengers [3]. Accurately understanding the influx and movement of passengers in the terminal becomes a significant yet difficult task. Predicting passenger movement in the terminal remains almost an impossible task as it involves human psychology, weather conditions, and even seasonal fluctuations. Instead of estimating passenger movement based on the current passenger volume, real time intelligence gathering would be a much more accurate approach to provide valuable insight for proper optimisations to be done.
As a combination of big data intelligence and the aim to improve operational efficiency, an effective people movement detection solution is required as a key enabling solution in this big data intelligence framework.

1.2. Usage of People Movement data

Understanding people movement can give a huge advantage to operational and management staff as it allows methodical optimisations to be performed on their operations. Practically, this includes improved marketing and management strategy, accurate labor force approximation, improved resource allocation, and much more. Considering marketing and management strategies, AAHK staff can understand the true passenger flow pattern of a shop, allowing for better the accuracy in estimating their respective land value. Understanding passenger flow also allows more efficient distribution of facilities and suggests better layout based on actual movement. Another aspect is labor force approximation and resource allocation. From check-in, customs, lounges, to boarding, there are numerous queues in the airport serving as a buffer to demand fluctuations. Understanding the service demand and flow rate of the queue can lead to substantial customer satisfaction

improvement as resources can be quickly allocated to where it is most needed, reducing queue time and potential delays to passengers. From the perspective of the operational staff, knowing the queue length and its development also prompt them to set up or remove queueing equipment for more organized queuing patterns.

1.3. Proposed Approach: Deep Learning Inferencing

Traditionally, such kinds of business intelligence are gathered from manual data collection and staff experience. While such an approach is effective, it has a few major drawbacks. Firstly, in order to cover every area in the massive concourse, labor intensive work in counting and estimating the passenger movement is required. Secondly, accuracy is severely lacking as counting moving passengers and understanding their movement is no easy task for a human. Finally, the scope and frequency of this type of data collection cannot possibly be sustained at a high intensity due to the time and cost involved in surveying such a vast area. To conclude, manual counting's biggest limitation is its low cost effectiveness, sustainability and accuracy.

Consequently, an automated and scalable solution is required to accurately identify and track humans in a scene. Notwithstanding the significance of gathering passenger movement intelligence, there is yet to exist a mature solution for such applications, especially when a camera cannot be positioned close enough to the subject of interest. HKIA's architectural design roof makes it very difficult to cover areas without using very high overhead cameras. Besides surveying the effectiveness of existing human counting solutions, we would also specifically survey the accuracy of such solutions in this specific scenario where the camera overhead is far away from the subject.

1.4. Integration for Visualization

While the project focuses more on implementing and evaluating the effectiveness of people counting and movement detection solutions, visualization remains a key part of the project. Visualization allows the effectiveness of the deep learning model to be easily understood while also serving as a prototype to showcase the potential of people movement detection solutions. By allowing users to interact with the prototype, each user can discover patterns and trends that have business implications to their respective job duties. The prototype will focus on providing real time queueing information as well as real time regional passenger volume data which can all be easily accessed and viewed through a web application. This web application should help users understand the potential in terms of business intelligence and may well inspire suggestion for other metrics to be counted or analysed as an extension to the system.

1.5. Project Contribution

As the world enters a digital era, big data intelligence in terms of people movement becomes a key area where any physical service-oriented businesses are interested in. The more one is able to understand about their customer and operations, the easier one can optimize and improve their services. As there are no mature solutions to date that handles such kinds of technical and business

requirements, such as people detection using footage captured far above the head or queue statistical analysis, our project aims to integrate such solutions together and evaluate the effectiveness of different models under different circumstances. Overall, the result would serve as a benchmark and reference for interested parties who are looking to build their own integrated system for a people movement detection, as well as for managers to understand the effectiveness and limitations of current solutions under different circumstances.

1.6. Report Outline

This report will be arranged into eight chapter.

The first chapter introduces the project motivation mentioned above with the second chapter concluding the project objective.

The third chapter will present the methodology adopted in this project, including system implementation, MLOps strategy and pipeline, as well as data visualisation techniques to be implemented.

The fourth chapter concerns about the results and summary of our evaluation. Mainly it describe the effectiveness of different models using density map generation, as well as the factors impacting the accuracy of bounding box models.

The fifth chapter describes the ML pipeline adopted for the project as well as its integration with the frontend and backend server.

The sixth, seventh and eight chapter mainly discuss about the challenges faced, present the conclusion for the project, as well as propose possible future work in relation to the findings of this project.

2. Project Objective

- Compare state-of-art deep learning models and Machine Learning algorithms in terms of effectiveness in crowd counting
- Extract people density information from inferenced result
- Evaluate effectiveness of PeopleNet on human tracking and queue counting
- Apply MLOps techniques in model training and data management to obtain results for evaluation
- Visualize people density data in heatmap as well as queue statistics on web application
- System integration of MLOps pipeline, backend and frontend as a prototype solution

3. Methodology

The project main objective is to perform comparative analysis on different models and evaluate their effectiveness to be used in the airport scenario, as well as construct a working prototype as a proof of concept. Thus, our methodology will describe and explain the detail application and system architecture employed and the relevant Machine Learning Operations (MLOps) strategy adopted in the project. This chapter will be separated into three sections. The first section describes the overall system architecture to introduce the general business logic and data flow. The second section discuss the reason for choosing the inference toolkit used in this project and the relevant MLOps strategy being applied. The third and final section describes the general strategy used for presentation as a proof of concept.

3.1. System implementation

For the system pipeline of the prototype, video streams will first be passed to deep learning models for inferencing, then the inferencing result will be passed to the backend server for further processing and finally made available to the user through web application. Detail pipeline is explained below.

Firstly, video streams captured from different sources such as CCTV will be made available to the inferencing platform which captures the stream automatically. Secondly, the inferencing platform will pass frames to the deep learning model to obtain different inferencing result depending on the requirement. For density analytics, bounding boxes coordinates on video that encompass each individual human identified in the video stream will be returned. For queuing analytics, the number of identified person that passthrough the queue exit will be returned. Thirdly, all of this information will be passed to the backend server for further processing. At this stage, all the data will be processed such that the information is meaningful for display. As a future extension, additional data analytics may also be performed real time as live inferencing results feeds in. Finally, density and queuing statistics can be view on a web application as the frontend quires or establishes web socket connection with the backend server. Density data are visualized as heatmap and real time video stream with queuing statistics can be viewed.

3.2. Density Detection using Deep learning

Deep learning is an obvious choice for automated human detection task for its ability to adapt to different scenarios, such as different CCTV angles, lighting, and object size. To adapt to such situations, different models as well as training data can be selected in order to achieve better results. As deep learning is a supervised learning method where we are training the models ability to map input to output by systematically adjusting the weights in the model, it is important to evaluate the effectiveness of each model in different scenario. This allows us to select the best architecture for each scenario as well as help us understand how to further improve the training process and the data requirements. To further explain the MLOps adopted for the evaluation, the following will discuss our approach for 1. Data Management, 2. Model Selection, and 3. Model deployment.

3.2.1. Data Management

As our evaluation very much focus on the scenario where footage are captured through high celling CCTV, our data management effort will focus on acquiring look-alike data through different means and process it accordingly for a more representative evaluation.

Data collection

To collect relevant data for inferencing and analysis, it is important to first identify the characteristics of data that is likely to be captured through CCTV inside the HKIA concourse. As explained by AAHK staff, the majority of CCTV cameras are position high up in the ceiling which are a few stories high as shown in Figure 3.1 HKIA ceiling, therefore footage to be analysed should be far away from the subject and captured at a large angle. For other factors such as lighting, since the HKIA operates 24/7 and the majority of the concourse in indoors, we can focus on collecting data with adequate lighting. Finally, it is not likely that there are much change or movement expect passengers passing by, therefore the data collected can also focus on more static scene with less interference.

Figure 3.1 HKIA ceiling and indoor area [4]

The second concern for data collection is to collect data that allows us to perform comparative analysis in respect to different types of result. The first result type is density detection, requiring the model to identify as many humans in the scene as possible. To better evaluate base on this criterion, different settings ranging from a few people to many people are all chosen. This enable testing the performance of the model on issues such as people overlapping in front of camera. The second type of data is queueing statistic data, requiring us to identify the number of people in a queue as well as how many people has left the queue. Thus, we should also collect scenes where people are queuing up to evaluate the effectiveness and limitation on counting the number of people leaving the queue under different camera setting.

The actual data collection strategy will consist of utilizing existing dataset as well as performing manual data collection so as to collect as fulfil the above requirement. Unfortunately, due to the current situation of Covid-19 as well as privacy issue, data collection in the HKIA concourse is not feasible. Therefore, our data collection will come from other sources which datasets that best

fulfil the requirements are chosen. For existing dataset, online datasets used in other crowd counting projects that fits the characteristics required will be used for density map generation evaluation. For manual collected dataset, data collection will be separated into two parts. The first part mainly consists of taking static photos within HKU at locations with significant height difference using a tripod and a camera. The main purpose is to evaluate the effectiveness of different models in identifying humans just to get an initial understanding of their performance with the specific scenario given. The second part will mainly be capturing actual scenes of queuing. To find scenes queuing while much of the indoor activities has been suspended, we decided to capture scenes of people queuing up in outdoor Community Testing Centres. Two drones both equipped with a 4K camera though with different optical system are deployed to capture video at varying heights and distance from the queue, allowing comparative analysis to be done on the effectiveness of different models under different settings.

Data Pre-processing

The data pre-processing task mainly prepares the data for easier evaluation. One notable pre-processing performed is applying video stabilization to the footage captured by the drone before passing it for inferencing. As the drone sometimes experience turbulence while capturing the queueing footage, stabilization is applied so as to simulate a stable video capturing platform using a CCTV, allowing it to theoretically perform better in tracking and counting the number of passes over a virtual line. Further data pre-processing such as downscaling is also required so as to fit a large image into the model as VRAM on the inferencing platform are somewhat limited.

3.2.2. Model Evaluation

Deep learning model operates by identifying latent features within each frame. Different model architecture and size will have varying success in capturing such latent features. The better a model is at identifying such features, the higher chance it has in ‘identifying’ a human in the frame.
To evaluate and suggest the best model for this application, we will survey a number of state-of-art models by comparing its performance on the data we had collected. Traditional models operates by drawing bounding boxes around humans identified in the scene as a form of object detection algorithm, this type of detection is called regression-based counting. While proven to be effective, they are somewhat limited by the model size and only a limited number of people can be identify at once. Under this category, the Nvidia PeopleNet is used for its generally superior performance. A more novel approach will be density-based counting by creating human density map as a direct output of the model. Models to be tested under this category includes VGG16, RestNet50, RestNet101, MCNN, and AlexNet. Existing datasets with ground truth annotated will be used in order to train the models mentioned above for density map generation.
For the evaluation process, the accuracy of the model in terms of human counting as well as queue exit counting will be compared. For publicly available dataset with ground truth available, the data will be used to evaluate the effectiveness of the density map approach by comparing the result across different models. For the queuing footage collected, the count of person passing through computed by the PeopleNet model will be compared against the actual count that is determined manually.

3.2.3. Model Deployment

Model deployment strategy is a significant decision as it will determine if the system is production ready and be able to meet all business requirements. Although the deployment of model is just for proof of concept in our project, best practices are still being followed and designed are made according to the likely business and technical considerations faced in a similar project. The choice of inferencing toolkit, pipeline and infrastructure will be explained below. Mainly, the two major concerns when developing this system are cost effectiveness and flexibility.

Cost effectiveness

Cost effectiveness remains a significant factor as the expected coverage encompasses the whole airport indoor area. To perform inferencing on the thousands and thousands of CCTV cameras install in the airport, a great deal of computing power, infrastructure and manpower would be involved in the setup and maintenance of the system. As the usefulness of such system always depends on how much net utility it can provide, being able to run and scale cost effectively is critical for the project success. Therefore, the architecture and infrastructure included should be lightweight, require little human intervention, and be able to scale efficiently.

Flexibility

The second concern regarding the pipeline is its flexibility to adapt to different environment and changing business requirement. As we are in an age of rapid digital expansion, it's paramount to be forward thinking and design a system that has the flexibility to fulfil quickly changing business requirement. Such changes may involve replacing the model, scaling up the system on demand, or even providing additional features in the output. Overall, considering there are a vast amount of use cases related to people movement analytics and the rapid evolvement in technology, a mature, cost effective toolkit for the job should be a priority and significant design choice not to be overlooked.

DeepStream SDK

In order to fulfils the business requirements and be able to address the above concerns, a mature framework is required to provide a layer of abstraction that allows developer to focus on pipeline design and application development while the rest such as driver installation should handled automatically.

The Nvidia DeepStream SDK is chosen for its power capabilities [5]. The DeepStream SDK is a high level framework that does all the liaising with lower-level SDK and hardware, saving development time by making it directly available on different Nvidia hardware platforms without sacrificing much performance. While the DeepStream SDK plugins are interfaced on a high level, they are usually hardware accelerated natively and very easy to integrate into any existing pipeline. Therefore, users are able to add in and swap out different hardware plugins without having to reconsider the whole pipeline architecture, allowing quick changes to be made on the architectural level and while more focus can be put on developing custom applications relevant to the business requirements.

There are two components critical to the flexibility of the system available in the DeepStream SDK. The first part is the Tao Toolkit and the second part is the DeepStream GStreamer Plugin. The Tao toolkit is again a layer of abstraction provided to train, optimized and deploy model with great

flexibility and ease. Using the Tao toolkit, we are able to load in different models quickly onto the DeepStream SDK while retaining a custom deep learning pipeline. The GStreamer plugin is a part of the DeepStream SDK plugin library which is designed to handle real time stream encoding/decoding. Various capabilities are provided such as handling different codecs, adding filters, and synchronizations.

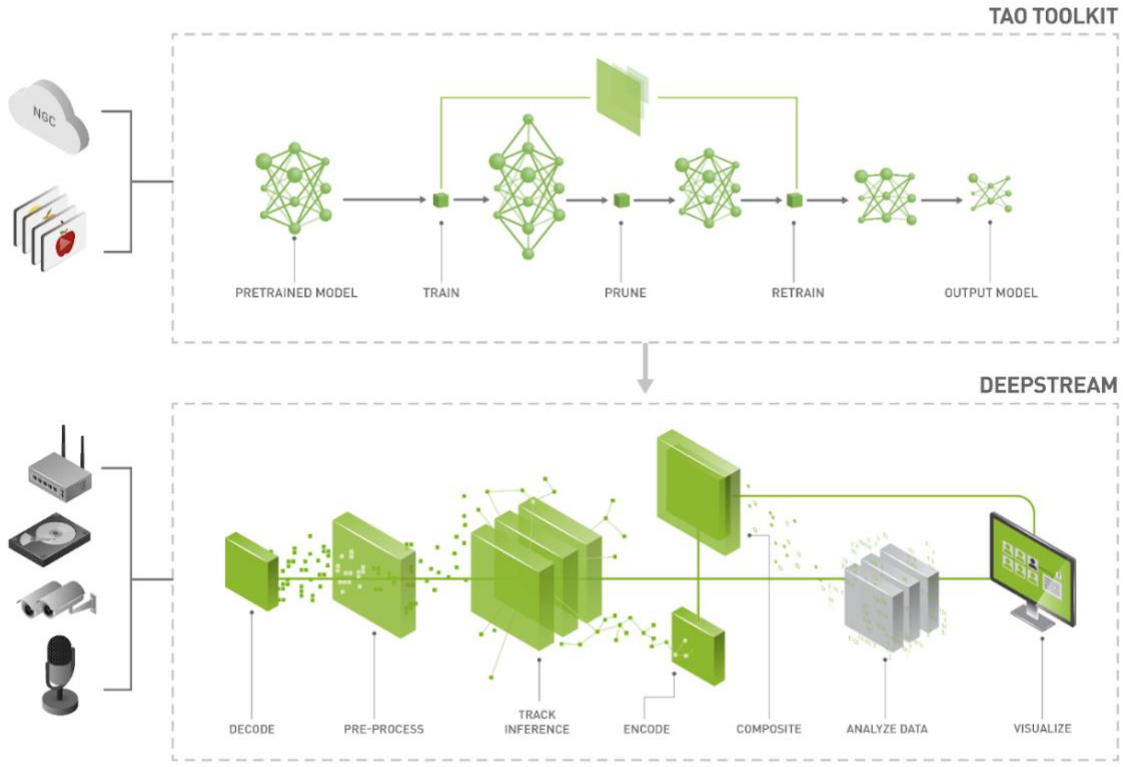


Figure 3.2 Model Optimization [6]

Processing

Utilising the plugins mentioned above, video streams in RTSP format will be listened and decoded into frames and store in the buffer by the DeepStream SDK. As this is a hardware accelerated plugin, it is able to utilise the hardware decoder onboard for better performance. Further pre-processing such as cropping and filtering may also be applied in case required for performance concern. For example, only frames of certain interval and areas reachable by humans are not filtered away to save inferencing workload. Finally, these decoded and processed frames will be inference by utilising the gst-ninverfeserver plugin which runs the model loaded at the time.

3.2.3.1. Additional Features

The DeepStream SDK also provide serval beneficial features that help resolve the cost effectiveness and flexibility concern.

Hardware buffer sharing

The first capability offered by the DeepStream SDK is hardware buffer sharing between independent plugins used in the pipeline [5]. Normally memory has to be copied and pass between independent components, making it slow, memory heavy and unscalable as more plugins are utilized in the pipeline. However, the DeepStream SDK allows hardware buffer sharing where only memory references are passed within the pipeline from input to output, making the number of plugins used having little performance implications. The effort to maintain compatibility between plugins in terms of hardware buffer sharing ensures performance and flexibility are sustained.

Deployment flexibility
The second capability is the ability to be deployed into a docker container and manage by Kubernetes. Hardware passthrough using docker containers are complicated to say the least and very problematic when the underlying hardware changes. The DeepStream SDK allows easy deployment through abstraction, making scalable deployment much simpler, enhancing the system flexibility and adaptability.

Post-deployment update/maintenance flexibility
The third capability is over the air model update [5]. As when operating the system, there may be updated business requirements, or a better trained model is available to be deployed. This feature allows zero downtime model update such that system continuity is ensured. Operators may try out the effectiveness of different models during production without having to reconfigure and reprovision hardware for the task. Moreover, the DeepStream SDK is very adaptable in terms of the model it's able to run. Besides pretrained models provided by Nvidia such as the PeopleNet model mentioned earlier, even models of different architecture or completely custom models can also be adapted into this framework. Ultimately, there is much flexibility in terms of model choice and model usage.

3.3. Data Visualization

As a proof of concept solution, visualization is a key part to help interested parties and stakeholders understand the possible business implications of people movement intelligence. Since two key statistics of human density and queue counting are generated, such data will be displayed using a heatmap and a real time annotated RTSP stream where users are able to see the queueing in real time. The heatmap data will mainly display human density information on an SVG map prepared in advance using intuitive colour coding where users with basic understanding of the floorplan should be able to quickly grasp the human density information within a glance. For the real time RTSP stream, it will be mainly used to display how many humans has exited the queue, and possibly how many persons are currently in the queue.

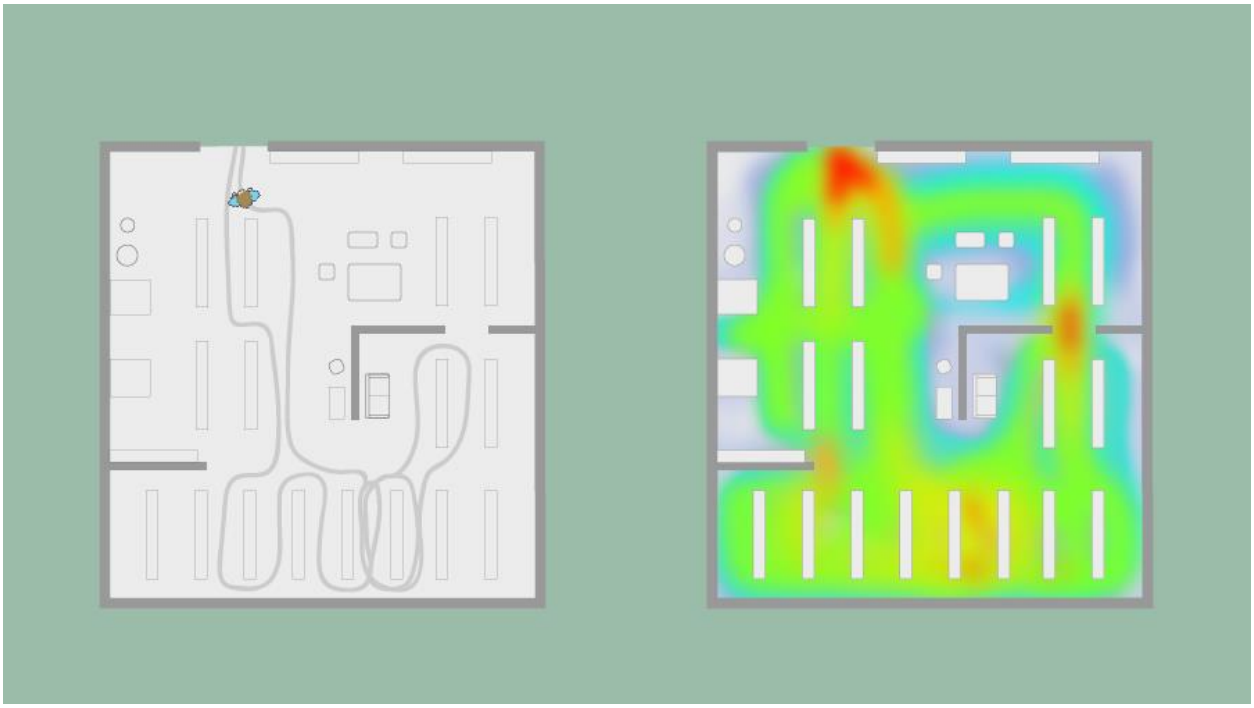


Figure 3.3 Example of heatmap [7]

To display the data, we would first need to prepare an SVG map used for prototyping purpose. Then on the frontend application, data visualization plugin that supports heat map display will be chosen. Moreover, plugin that allows displaying real time RTSP stream will also be needed for the queue counting component.

To fulfil the requirements above, a frontend web application using the React framework will be developed. This is chosen mainly for its popularity and amount of plugins available to choose from to best display the results listed above. The frontend application will be written in TypeScript instead of standard JavaScript since a strongly typed language can be very beneficial in avoiding bugs and saving development time.

3.4. Summary

This chapter of methodology outlines our MLOps tools and strategies adopted in the project as well as our choice of visualization. Firstly, we explain the method of data collection which involves collecting data captured overhead using existing datasets, static photo and drone video footage. Then the details and pipeline of DeepStream SDK with its benefits in terms of cost effectiveness and flexibility is illustrated. Finally, the plan of using a frontend web application containing heatmap and real time annotated RTSP-stream for data visualization purpose is described.

4. Experiments and Results

In this section, dataset used for evaluation, pre-processing work done, results of comparing the effectiveness of density map generation, and capability of bounding box regression will be presented.

4.1. Model Survey

Density map generation is a relatively new approach where a crowd counting algorithm generates density map based on the model's understanding of the scene. For evaluation purposes, we would train five different models using the same datasets and compare their performance with relevant metrics. Details of dataset selection, pre-processing, evaluation metrics, model architecture, and results will be shown below.

4.1.1. Dataset Selection

Two popular datasets have been selected in order to train and evaluate the effectiveness of the five models chosen for this comparison. The first dataset is UCF-QNRF(QNRF) dataset, featuring a variety of scenes with each scene containing a lot of humans. Another data set is the ShanghaiTech Part B (SHTB) dataset, featuring scenes of busy streets with many people packed together. The table below specifies the parameters and statistics of each dataset.

Dataset	Number of Images	Resolution (*H* X *W*)	Count Statistics			
			Total	Min	Ave	Max
UCF-QNRF	1,535	2013 X 2902	1,251,642	49	815	12,865
ShanghaiTech Part B	716	768 X 1024	88,488	9	123	578

Table 4.1 Images specification of the dataset

As seen in Table 4.1 Images specification of the dataset, the main differences between the two datasets are their resolution and count statistics. On average, the QNRF dataset contains much more humans in one frame than SHTB as it is larger viewing angle which covers a larger area. On the other hand, the SHTB dataset contains images that covers a smaller area and has less resolution.

4.1.2. Data Pre-processing

Images originated from the datasets are resized in the data pre-processing phase for two main reasons. Firstly, it is done to control the VRAM usage in the training process. Since the graphic card used for the training process is the NVIDIA GeForce RTX 2080 Ti, only 10GB of VRAM is available. Since the QNRF dataset contains images of quite high resolution, to ensure the training process runs smoothly, images are resized into a lower resolution so as to make sure both the model and images are small enough to fit into the VRAM together while being loaded for computation.

Secondly, resizing is done to fit models that consist of a down sampling layer. Since the down sampling layer requires input image resolution in the factor of 16, images in the QNRF dataset are resized to 1024 X 1024 so as to fit this requirement. Since images in the SHTB dataset is sufficiently small and has a factor of 16 (768 X 1024), no resizing is necessary. After resizing, label normalization is also applied by applying magnification to the ground truth density map, which is then supplied for training purpose. This magnification factor is found to be significant as it greatly reduces training time by reducing time for the model to converge.

4.1.3. Evaluation Metrics

Two metrics, mean absolute error (MAE) and mean squared error (MSE), are adopted for evaluating the accuracy of the model trained for evaluation.
The MAE equations are as follow:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |fi - y_i|$$

The MSE Equation is as follow:

$$MSE = \sum_{i=1}^{n} (fi - y_i)^2$$

In these two equations, f_i denotes the predicted value while y_i denotes the ground truth.

4.1.4. Model Training

We have chosen to train and assess the models using the HKU CS GPU Farm 2 which is equipped with the NVIDIA GeForce RTX 2080 Ti as mentioned above. Each model where only trained with 30 epochs as resources are limited. Pytorch framework was used to run the training and evaluation process.

4.1.5. Model Architecture

The architecture of the five models chosen for evaluation are discussed below so as to understand the potential implications in terms of human detection capabilities as well as performance implications. All the models chosen are CNN models which traditionally are traditionally used in various object detection task with great results.

4.1.5.1. VGG16

VGG16, with VGG short for Oxford University Visual Geometry Group, is a large neural network containing 16 layers. The model is pretrained with millions of ImageNet images and is capable of generic image recognition task. The first 10 layers of VGG16 was used as the encoder and last two layers are used as the decoder. Figure 4.1 Network Architecture of modified VGG [8] shown below is the architecture of VGG16.

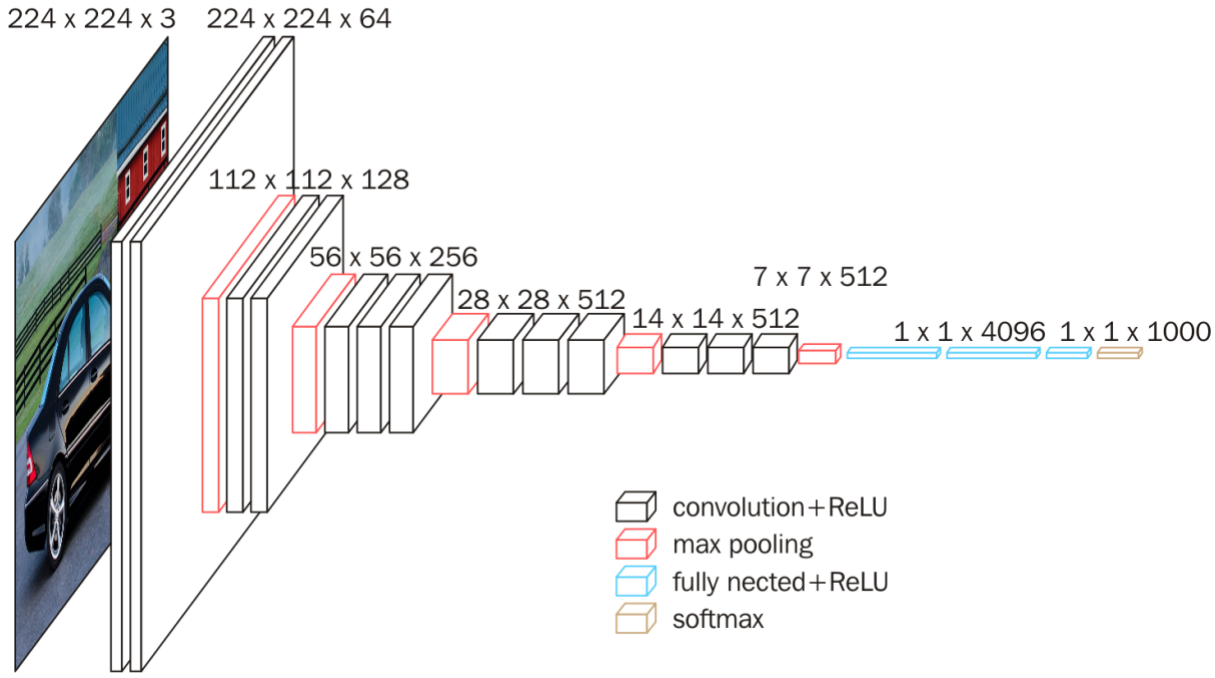


Figure 4.1 Network Architecture of modified VGG [8]

4.1.5.2. ResNet50/101

RestNet is a type of residual neural network (RNN) proposed in order to solve the vanishing and exploding gradient problem of a deeper network. This is done by creating direct connections across multiple layer which allows the model to be trained relatively quicker than similar size models.
In this survey both RestNet50 and RestNet101 will be tested and compared. Only the architecture of the RestNet50 model is as showed below in Figure 4.2 Network Architecture of modified ResNet50 [9] as the RestNet101 architecture are very similar just with more layers in the middle.

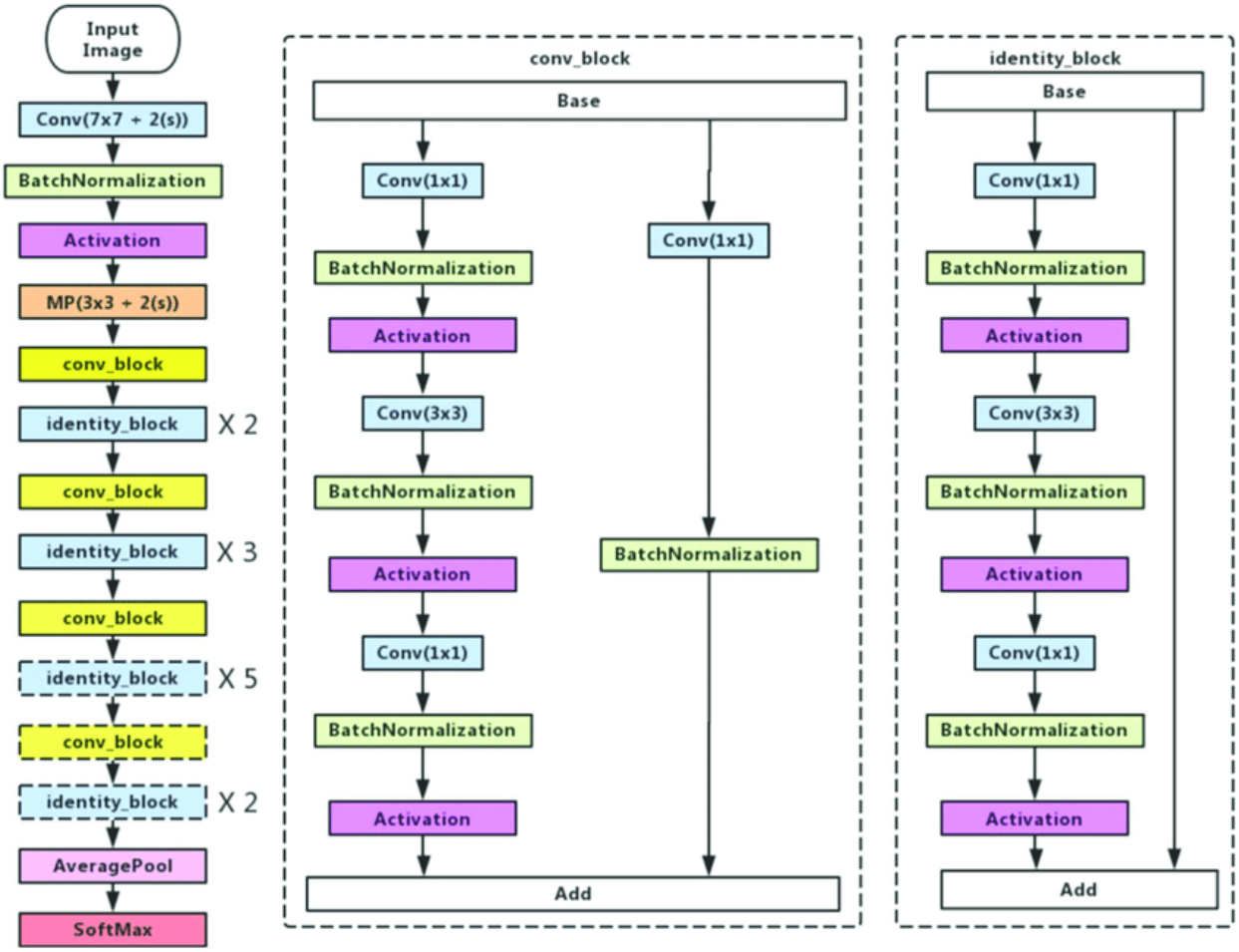


Figure 4.2 Network Architecture of modified ResNet50 [9]

4.1.5.3. MCNN

MCNN, short for Multi-Column Convolutional Neural Network (MCNN), is proposed specifically for crowd counting in 2016. MCNN takes images as the input as divide them into appropriate size before having each segment pass through a parallel CNN network together with the output of each CNN network merged together forming the final density map. The architecture of MCNN is as shown below in Figure 4.3 Network Architecture of modified MCNN [10] with slight modifications applied.

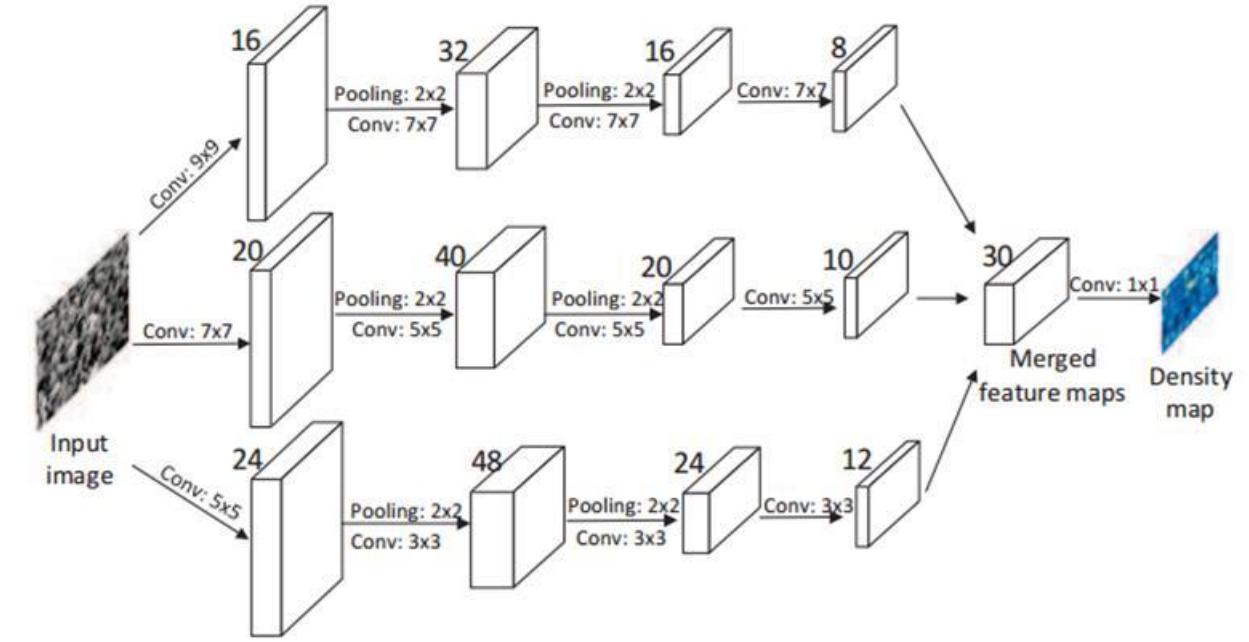


Figure 4.3 Network Architecture of modified MCNN [10]

AlexNet

For the last model. AlexNet, it is an 8 layer CNN model also specifically designed for image recognition task. We have modified the padding parameter in order to make sure the feature map can be divided properly. The first 5 CNN layer is used as the encoder along with a 1000-way softmax later at the end with the decoder architecture similar to VGG, and within the encoder layer also contains the down sampling layer.

The architecture of AlexNet is as shown below in Figure 4.4 Network Architecture of modified AlexNet [11].

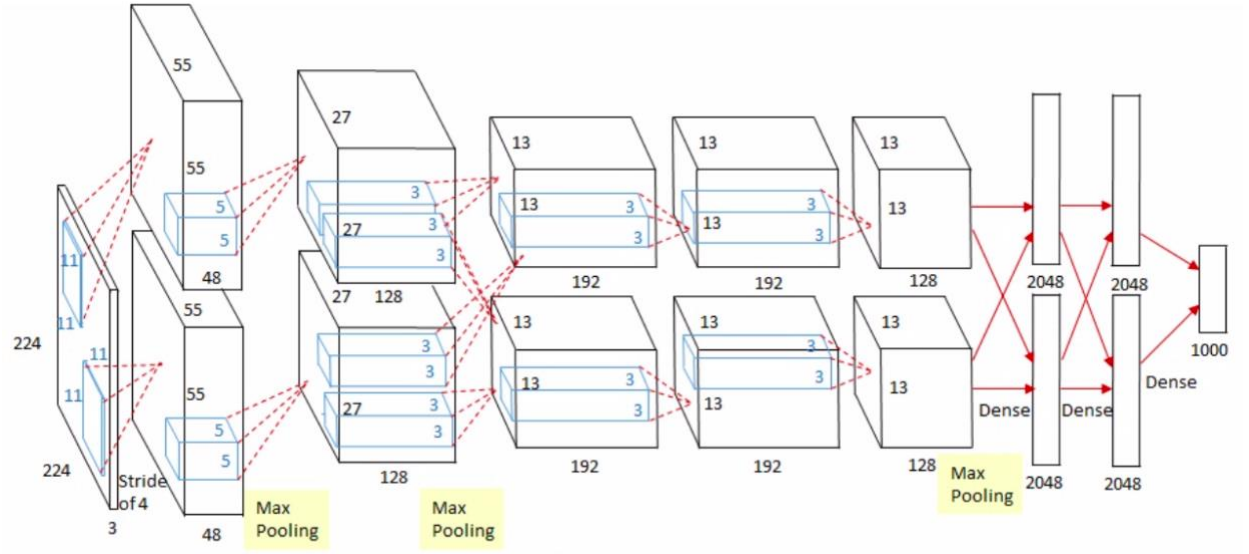


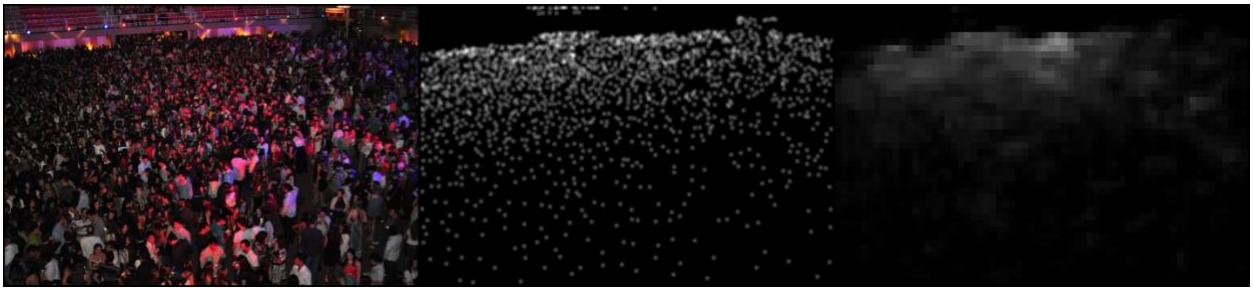
Figure 4.4 Network Architecture of modified AlexNet [11]

4.1.6. Result

We will first present examples of visualization result for each dataset, then mathematically evaluate their performance with the selected metrics.

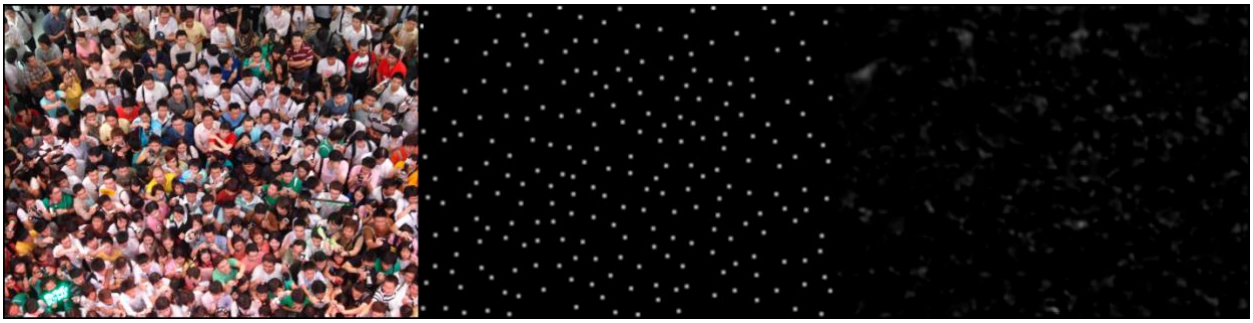
4.1.6.1. QNRF Dataset Visualization

AlexNet



Images Ground-Truth Prediction

MCNN

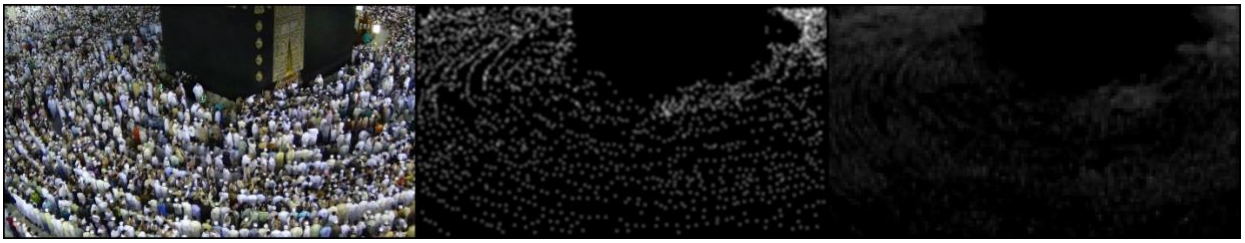


Images Ground-Truth Prediction

ResNet50

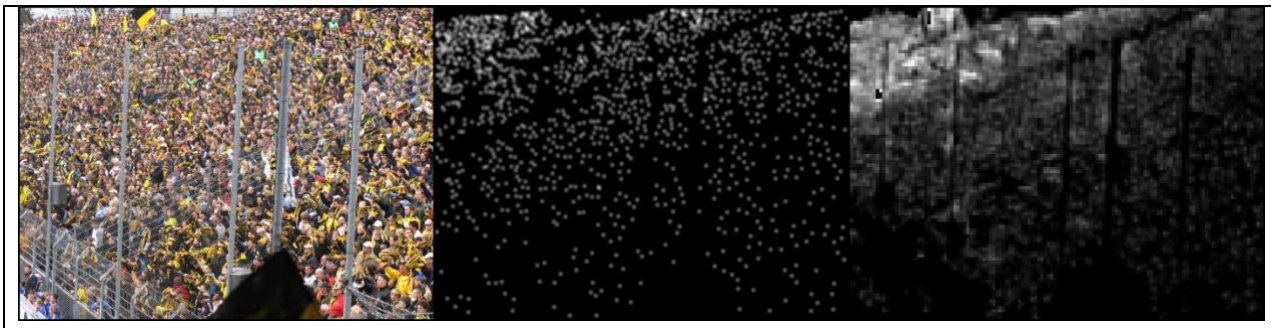
Images Ground-Truth Prediction

ResNet101



Images Ground-Truth Prediction

VGG



Images Ground-Truth Prediction

4.1.6.2. SHTB Dataset Visualization

AlexNet

Images Ground-Truth Prediction

MCNN

Images Ground-Truth Prediction

ResNet50

Images Ground-Truth Prediction

ResNet101

Images Ground-Truth Prediction

VGG

Images Ground-Truth Prediction

4.1.6.3. Model Evaluation Results

Models are compared mathematically using MAE and MSE as the evaluation criteria. These two values basically represent how much it differs from the ground truth, so the lower the number the better the performance.

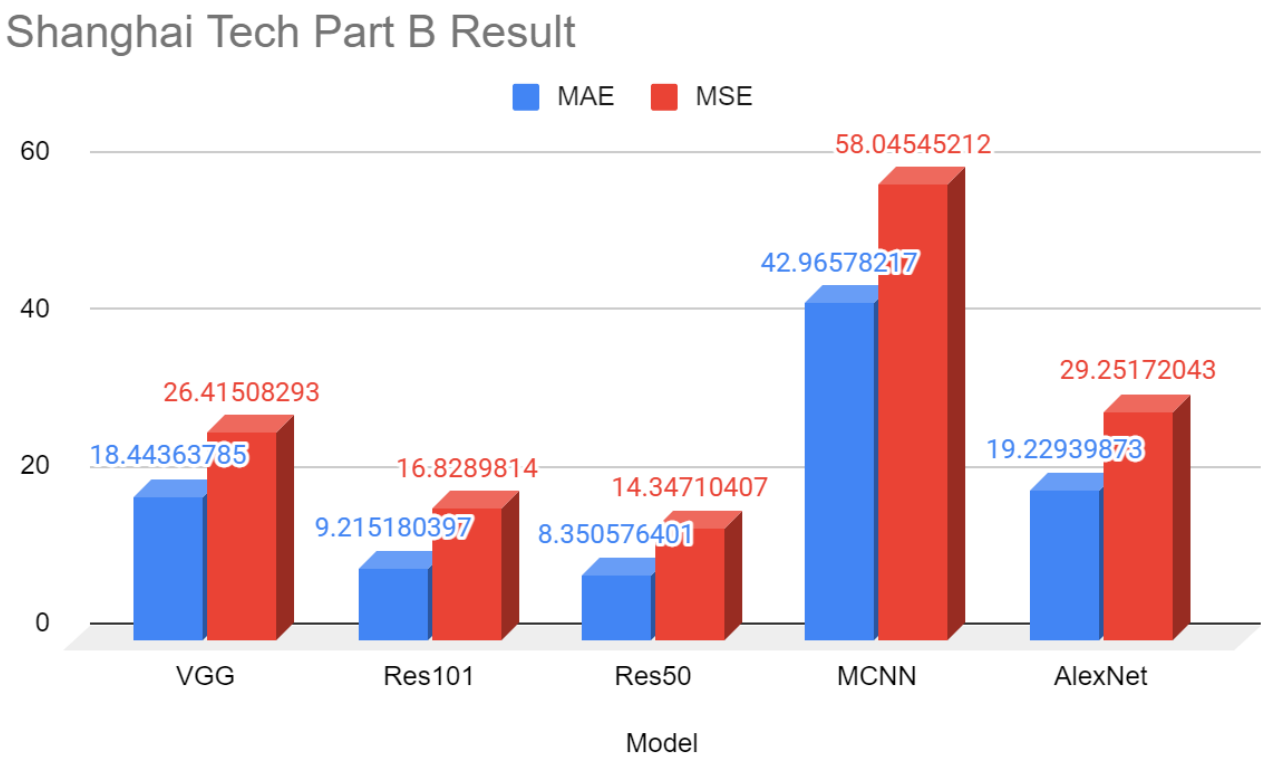


Figure 4.5 Result of Classification model for the Shanghai Tech Part B Dataset

Figure 4.5 Result of Classification model for the Shanghai Tech Part B Dataset compares the result produced by the 5 selected models using the SHTB dataset. The best performing model is ResNet50 with the lowest MAE and MSE value while the worst performing model is MCNN with the highest MAE and MSE value across all models. Interesting point to note is RestNet 50 outperformed RestNet101, possible due to insufficient training as it has basically double the depth compare to RestNet50 but the number of training epochs is the same.

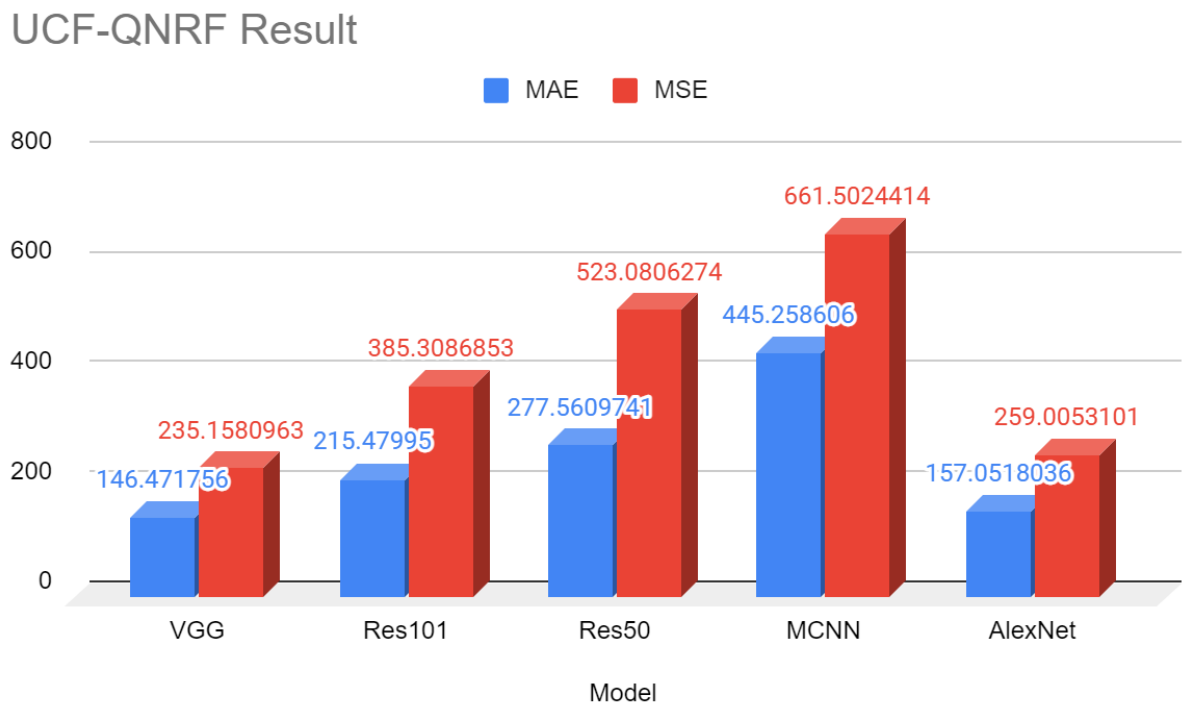


Figure 4.6 Result of Classification model for the UCF-QNRF Dataset

Figure 4.6 Result of Classification model for the UCF-QNRF Dataset compares the result produced by the 5 selected models using the QNRF dataset. The best performing models are AlexNet and VGG with similarly MAE and MSE values while the worst performing model is again MCNN with the highest MAE and MSE value across all models. The RestNet family models while better than MCNN, performance significantly worse than VGG and AlexNet.

4.1.7. Summary

For the SHTB dataset which contains camera much closer to the subject, models from the RestNet family may be a good choice. However, in settings where there are a significant number of people and each person occupies a smaller portion of the frame, VGG and AlexNet seems to be a better choice as both displayed dominating performance using the QNRF dataset. Overall, there are no conclusive answer on which model is superior in all situation.

4.2. Bounding Box Regression Evaluation

Bounding box regression models, with results visualised in Figure 4.7 Bounding Box Regression, are the traditional alternative to density map models shown above for people counting applications. One primary reason density map models are chosen is for its ability to estimate the number of humans in a scene while a tradition bounding box regression model fails to produce meaningful reason. This could be due to each image containing too many people, causing each human to only taking up a small portion of the frame that classification cannot be done.

While the density detection approach is promising, it only output probability density map of humans without actually identifying each individual human. Accuracy is much lower than using bounding box regression in scenes where bounding box regression models work properly. Density map also does not provide additional capabilities such as subject tracking. If the bounding box approach can be made possible in the airport setting despite the limitation imposed by high hanging camera, it will be a much-preferred approach for the reasons mentioned above. Since the main reason is each human being too small in the frame, determining the number of pixels required to do classification will be important. This will serve as a benchmark for different use case for determining the equipment and zoom factor required if the bounding box regression approach is adopted. As an extension to this analysis, the effectiveness of scaling up a photo for queue counting applications will also be evaluated.

The following section will present the arrangement and results of inferencing scaled up images as well as investigate if queue counting is still worthwhile under this arrangement.

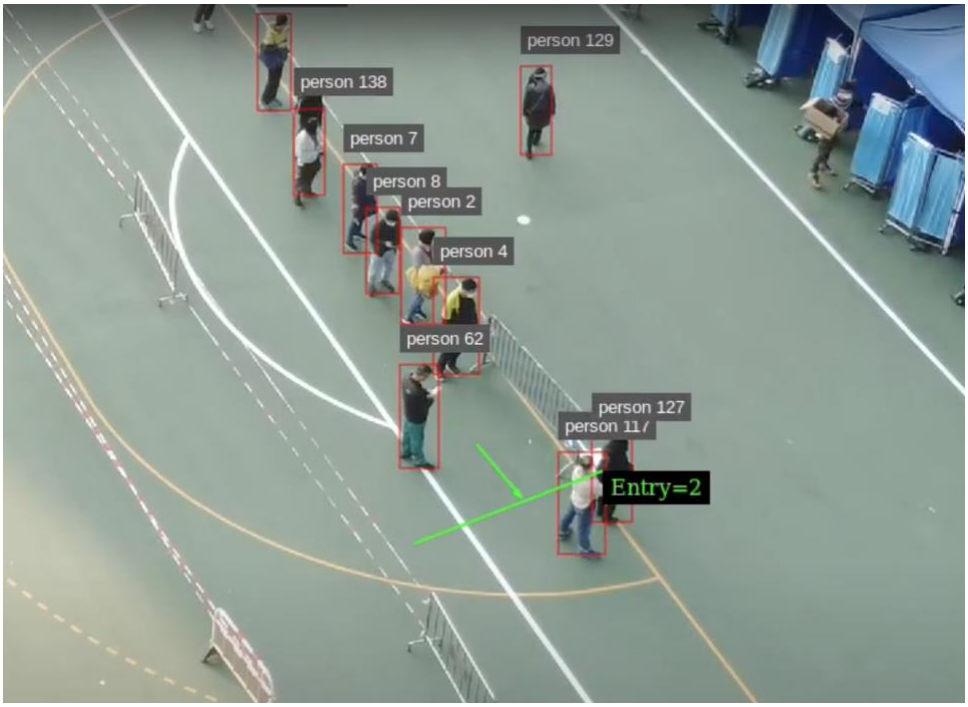


Figure 4.7 Bounding Box Regression

4.2.1. Image Scaling Procedure

As described in the methodology, images used in this evaluation are sourced from aerial videos taken overhead of outdoor Community Testing Centres. Frames are extracted and resized from 3840*2160 to 960*540. Resizing the image is necessary in order to fit the images to the PeopleNet model developed by Nvidia. The PeopleNet model is an accurate and powerful model pretrained by Nvidia with industry leading performance and serves as a baseline for this benchmark.
After resizing the original image, 20 new images are created by having each image scaled down by an additional 5%. From a scale of 5% to 100% of the original image, these 20 images are inferenced by the PeopleNet Model with the NMS clustering algorithm threshold confidence set to 0.6. The empty portions of the image is replaced with a white background such that the scaled images maintains the same image dimension. The result can be seen on Figure 4.8 Gallery of Inferenced Images.

With the inferencing result, statistical analysis is done to figure out the pixel requirement for human classification.

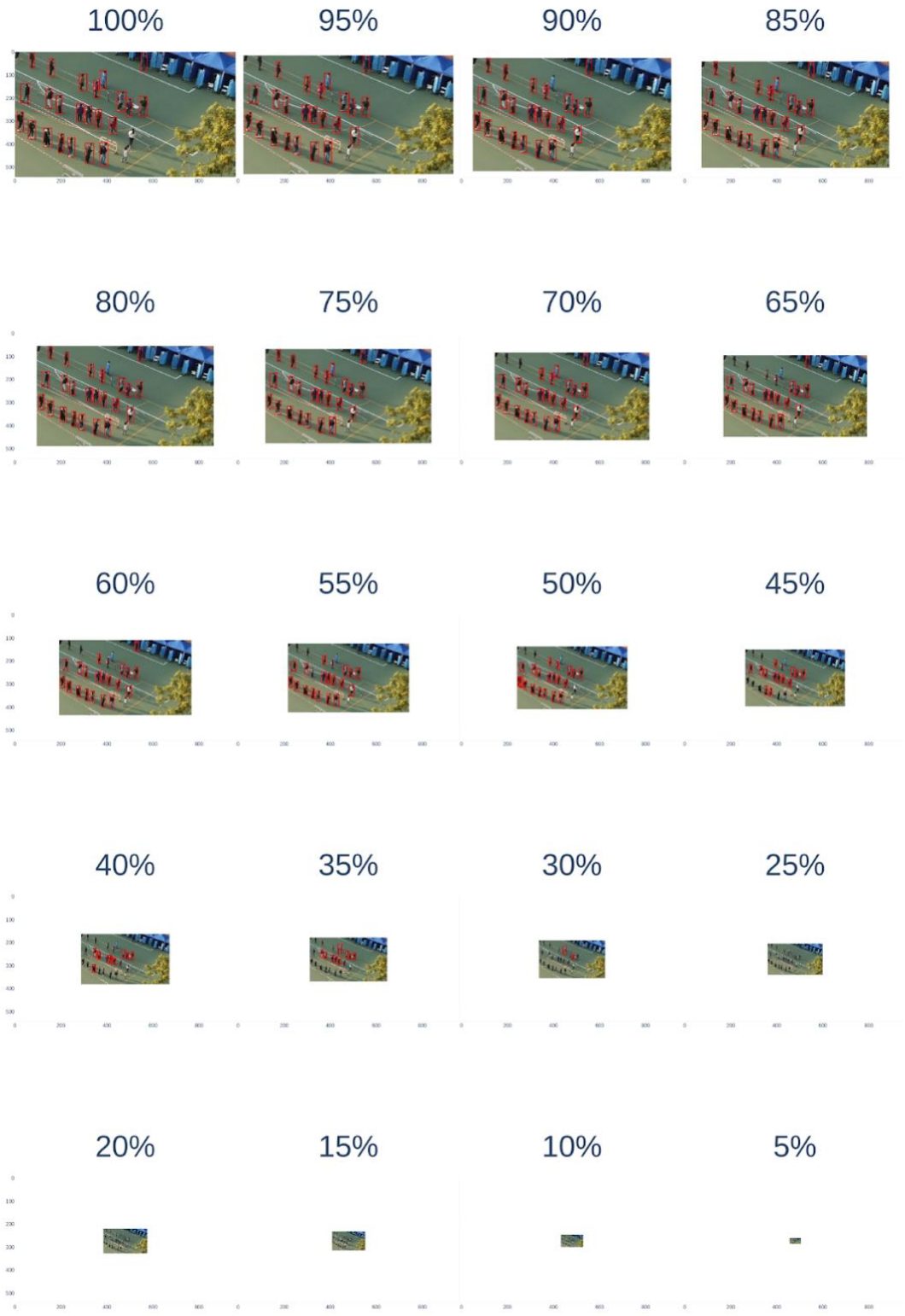


Figure 4.8 Gallery of Inferenced Images

4.2.2. Result

Using the scaled down images, the correlation between the number of pixel and chance of classification can be revealed. For each image, the pixel count of the smallest bounding box is counted, and it compared against with the number of classifications made. Since all classifications are valid with no false positive, we can directly correlate both factors.

Scaling factor	Number of BBox	Pixel count of smallest BBOX	Scaling factor	Number of BBox	Pixel count of smallest BBOX
0.05	0	0	0.55	21	560
0.1	0	0	0.6	21	645
0.15	0	0	0.65	19	810
0.2	0	0	0.7	26	864
0.25	0	0	0.75	25	795
0.3	2	**275**	0.8	24	880
0.35	9	297	0.85	25	1062
0.4	10	372	0.9	24	1152
0.45	13	408	0.95	25	1533
0.5	22	408	1	24	**1420**

Table 4.2 Result of the Analysis of human size inside a scaled down Image

From Table 4.2 Result of the Analysis of human size inside a scaled down Image, we observed that as the scaling factor increase, the number of pixels of the smallest bounding box also increase, such that it confirms the experiment is set up as expected. Starting from the scale of 0.3, PeopleNet is able to detect humans with a minimum pixel count of 275. When the scale is 1, the pixel count of the smallest bounding box is 1420 and the number of detected humans is 24. It is obvious that the larger the scaling factor, generally the more subject it is able to detect. The correlation of these numbers will be further investigated below.

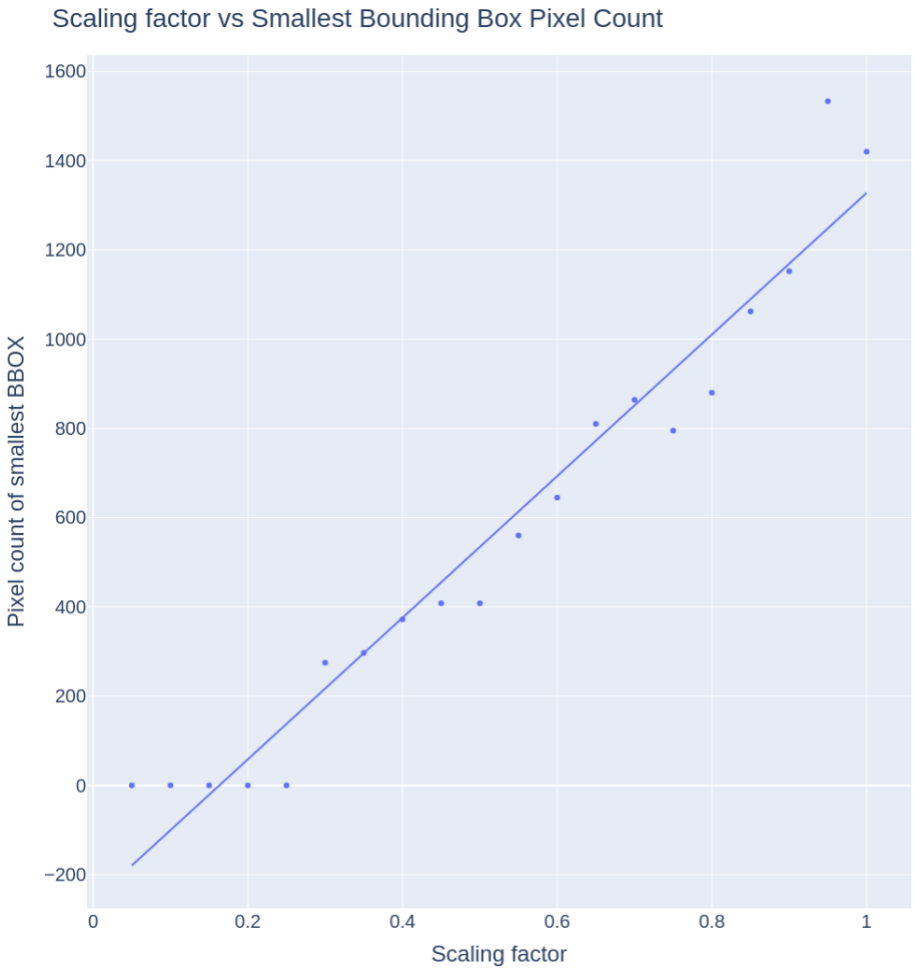


Figure 4.9 Scatter plot between scaling factor and the pixel count of smallest bounding box

Figure 4.9 Scatter plot between scaling factor and the pixel count of smallest bounding box above shows there is a clear strong positive correlation between the scaling factor of the image with the pixel count of the smallest bounding box. This implies the experiment is set up correctly as the goal is to evaluate the minimum amount of pixel required to reliably detect a human.

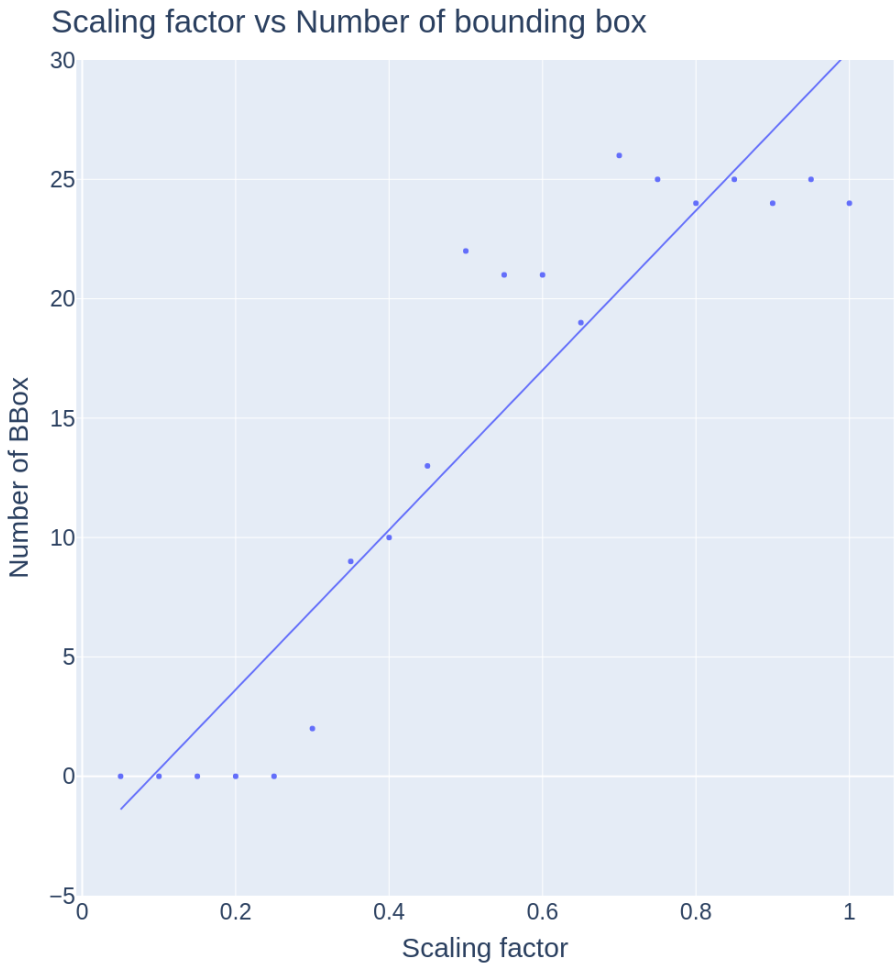


Figure 4.10 Scatter plot between scaling factor and the number of bounding boxes

Figure 4.10 Scatter plot between scaling factor and the number of bounding boxes investigates the relationship between the scaling factor and the number of bounding boxes. There is an obvious strong positive correlation between two factors, suggesting each increase in the scaling factor, which directly relates to pixel count, increases the number of humans the PeopleNet model is able to detect. Though it should be noted that starting from a scale factor of 0.7, the recorded number is more or less the same at around 25, suggesting it reached ceiling accuracy. And starting from a scale factor of 0.5, the recorded number is above 21, suggesting decent performance in detecting humans. Moreover, a minimum scale factor is needed before any detection can be made.

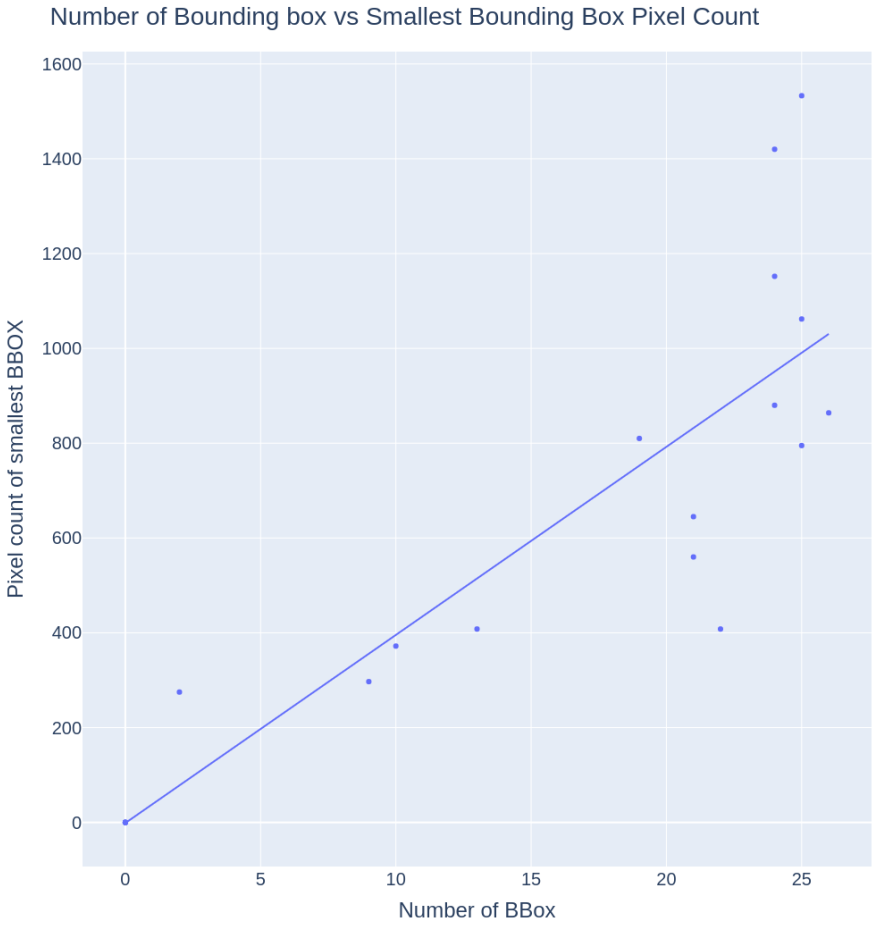


Figure 4.11 Scatter plot between number of bounding boxes and the pixel count of smallest bounding box

Finally, Figure 4.11 Scatter plot between number of bounding boxes and the pixel count of smallest bounding box confirms the above explanation and we can see starting from around 600 pixels per subject, the model is able to detect human with reasonable accuracy. Therefore, depending on the usage, users can choose the type of equipment that suits their budget and obtain results with anticipated accuracy. If higher accuracy is needed, a minimum of 800-1000 pixels should produce results sufficiently accurate to identify most human subjects.

4.2.3. Queue tracking accuracy

In addition to people detection, queue counting is also an important aspect of people movement statistics analysis. To address the concern of humans represented by too little pixels in a frame cannot be detected, video taken with varying height and distance from the queue will be evaluated for 1. If it is possible that scaling up the video allows detection of human with better accuracy, as we have established that pixel count correlates to identification accuracy, and 2. If the tracking capabilities are affected by scaling, as scaling up does not increase the available detail, such that does the tracker still operates sufficiently well when the input video source is scaled up.

To perform such comparison, videos were scaled up such that each human are represented by a sufficient number of pixels, meaning possible celling accuracy should be reached. Since we established that the higher the pixel count, the better the detection. If detection is not possible after

sufficiently scaling up, then it means it simply reaches the limit of the PeopleNet model. After scaling up, video excerpts are inferenced, and its queue count are compared with the actual count which is manually determined. Table 4.3 Comparison of automatic and manual queuing exit count under different settings below shows the specification of each video feature and why its unable to count a person passing the queue.

Height (m)	Horizontal Distance (m)	Distance	File name	Count	Actual	Reason for miscount
25	36	44	drone_1_1_ds.mp4	6	6	
50	50	71	drone_1_2_ds.mp4	8	8	
30	50	58	drone_1_3_ds.mp4	2	2	
25	30	39	drone_2_1_ds.mp4	6	7	Line moved as camera moved
25	30	39	drone_2_2_ds.mp4	5	6	People bunch up together, blocking tracker
20	26	33	drone_2_3_ds.mp4	4	4	
40	25	47	drone_3_1_cropped.mp4	N/A		Unable to detect or track due to limited resolution/overhead angel Although the camera is still 4K, the image quality is significantly worse
50	25	56	drone_3_2_cropped.mp4	N/A		
50	85	99	drone_3_3_cropped.mp4	N/A		

Table 4.3 Comparison of automatic and manual queuing exit count under different settings

Generally, as long as the human can be identified with consistency, it is able to count the number of persons exiting the queue. The only reason it cannot be counted are due to other issues such as the detection line moved relative to the scree as the drone camera shifted its perspective temporarily, as show in Figure 4.12 Normal Perspective versus Figure 4.13 Perspective drifted where the perspective drifted. Additionally, the exit was not counted as human was not identified in the first place due to being blocked by another person when passing the exit line (see Figure 4.14 Overlapped subject). Even in the case where the identification was not consistent when a subject passed the line, the queue counting operation is still successful in recording the exit.

Figure 4.12 Normal Perspective

Figure 4.13 Perspective drifted

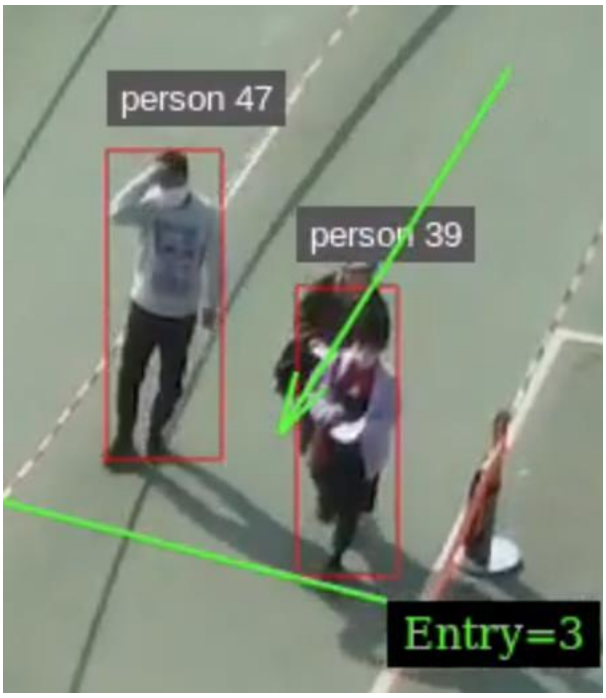


Figure 4.14 Overlapped subject

The inferencing result of the last 3 video shows there is a limit to the scaling up technique. There are a few reasons why no meaningful result was produced. Firstly, 3_1 and 3_2 were captured at almost overhead angle than the rest of the footage (see Figure 4.15), as it only has 25m horizontal distance from the subject but 40-50m vertical distance above, the incident angle is about 27-32°. Secondly, it is largely due to the different image quality as they are captured with different lens and CMOS. Images are taken as snapshot using VLC to preserved quality and then cropped to a size of 300*300. Image of better quality are placed on the left while the one with lower quality are placed on the right. The original images are also available at Figure 4.18 and Figure 4.18. image. We can observe a significant difference in quality when comparing Figure 4.16 and Figure 4.16. The video of 1_x and 2_x is captured by a better optical system while video of 3_x is captured with an inferior optical system. It may be hard to see on paper, while both are 4K camera, they provide significant difference in terms detail available as the inferior one (on the right) contains a lot of fuzziness while the better one (on the right) contains much more detail when zoomed in.

Figure 4.15 Overhead capture


Figure 4.16 Cropped Image with better quality


Figure 4.17 Cropped image with lower quality

Figure 4.18 Original image with better quality

Figure 4.19 Original image with lower quality

4.2.4. Summary and Implications

To conclude the findings in this section, it is estimated that pixel count representing a human and the details exhibited by those pixels are the two most important factor that determines if the human subject can be detected. It is also determined that the tracker perform consistently well as long as the human subject is consistently detected. Therefore, using scaled up footage for queue counting application does not pose any significant concern. This information provides some implications as to the hardware choice, positioning of hardware, and inferencing parameters that can be tuned.

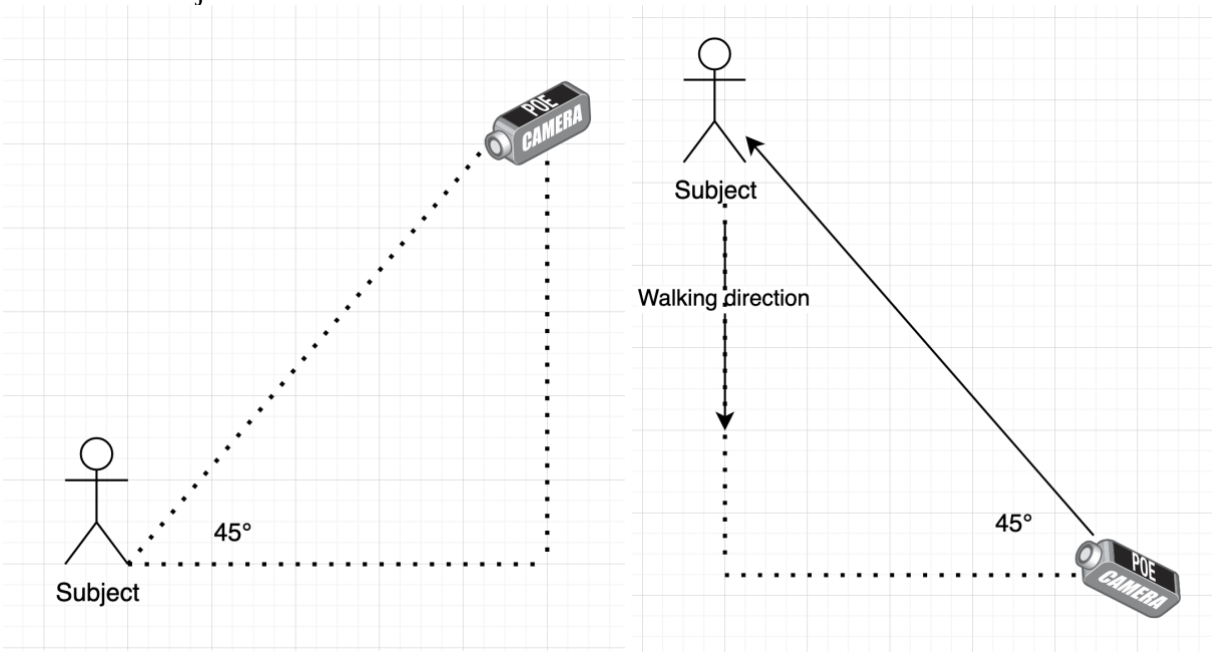
Capturing device requirement and consideration

For hardware choice, the required resolution of the camera depends on the task given. For example, if the primary use is just to count number of humans, even if only a portion of humans are detected due to limited resolution, the number can still be reliably correlated to such that the actual number in the scene can be inferred. In such case, having just 500-600 pixels per human would be sufficient. However, if human tracking is the primary usage, then a higher pixel count is at around 900 pixels will be necessary, where the model is already able to consistently detect most of the human subjects in the scene. We observed that if the detection is hit and miss between frames, the tracker is very likely to assign a different ID to the subject, making tracking of individual subject less effective. Moreover, the quality of the optical system is also a main concern. If the image is blurry, even when there are enough pixels to represent a human, there are still not enough information in the details that allows the model to classify if it is a human or not. Overall, if the camera has good image quality, a lower resolution or zoom factor can also be chosen with the video sourced scaled up in pre-processing phase before inferencing. As long as there are enough pixels and enough information, it is possible for the model to track the subject.

Positioning of capturing device

For the position of the camera, care has to be taken in order to avoid blocking of subjects and capturing sufficient details. For example, if the camera is placed directly facing an alleyway, it is very likely that one person will block the line of sight from the camera to another person behind him. Another example is when the camera is placed overhead with a very small incident angle, i.e. overhead of the subject, this not only makes the human smaller in terms of pixel count, but the head of the subject also blocked a lot of information that is necessary for classification. Optimally, the camera should be placed at least 45° above the subject to avoiding being blocked by another subject, but also not too high such that only the head of the subject is captured while details such as the body and limbs cannot be seen. Moreover, the camera should be placed 45° incident to the

usual flow of people, giving the best chance of capturing even if others are walking in front of, or besides the subject. The idea is illustrated as below.



Hyperparameter tuning

Finally, inferencing parameters can be tuned to achieve optimal result. One possible parameter to be tuned is the clustering confidence threshold. In the current setup, NMS clustering algorithm with a threshold confidence value of 0.6 was used. In the inferenced result of 3_x, it can be observed that the detection is kind of hit and miss. The confidence parameter can be tuned such that even when there is less detail or less pixel, it is still possible to extract meaningful result, though it might exhibit a higher error. This tuning can also be adjusted together with the scaling factor for optimal result without modification to the hardware.

5. System Design

The following chapter introduce the system design and how it is integrated together. Figure 5.1 below is the system architecture diagram, with detail of each component discussed further below.

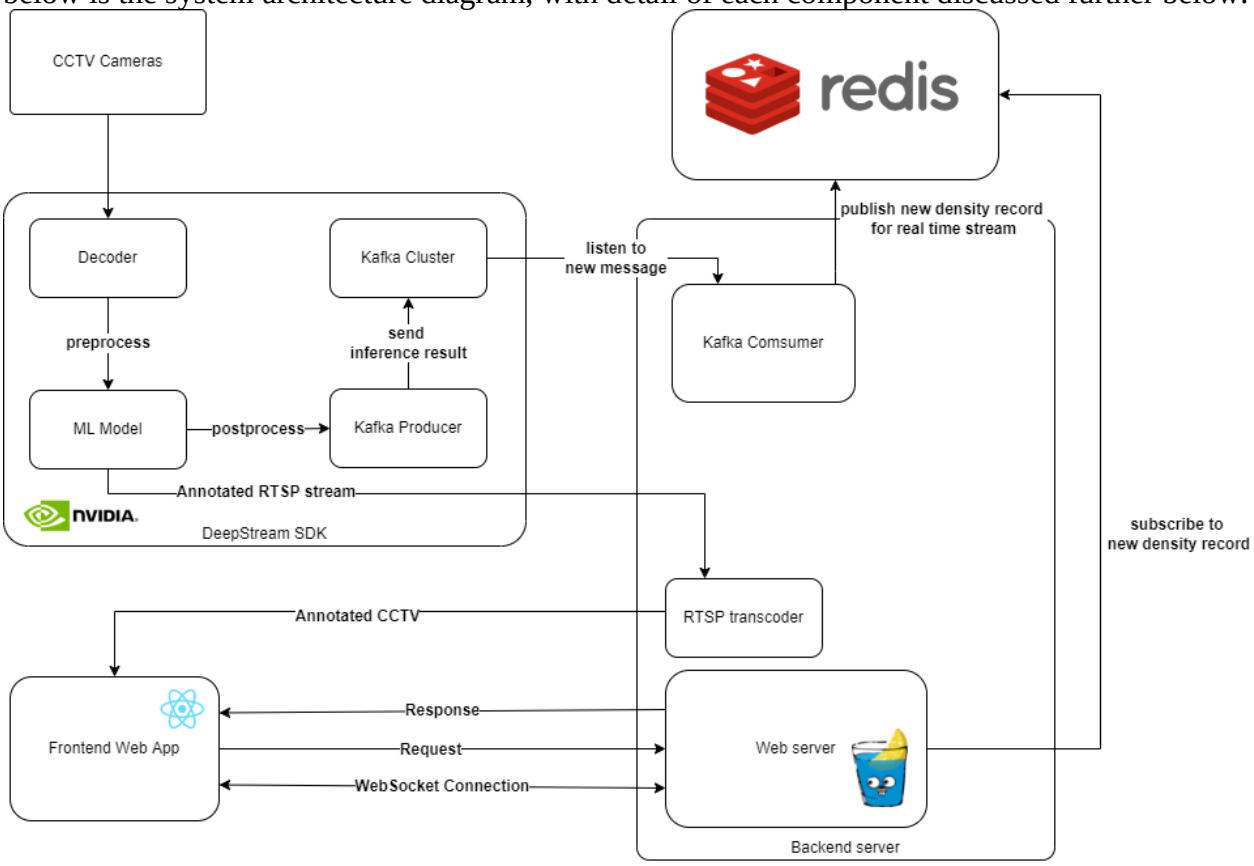


Figure 5.1 System Design Diagram

5.1.1. ML pipeline

This section will discuss the system architecture as well the machine learning pipeline adopted. In a production environment, video from CCTV or other source will be streamed to the DeepStream SDK for inferencing. However, for testing purpose, we store the file locally on the inference server, which is running the DeepStream SDK, and use it as a source. The DeepStream SDK takes this local file and emulates a RTSP stream by looping the file over and over again. The video will be decoded by the Gstreamer plugin as discussed where the decoded frames are dynamically batched for inferencing. After inferencing where multiple bounding box are proposed, the NMS clustering plugin will filter and cluster the results. Then, the bounding box chosen are passed on to another plugin that utilizes the KLT tracker, where subjects are uniquely identified and tracked across frames. All of such results will then be made available to a Kafka cluster as the result are published by the Kafka producer. Kafka is designed for low latency data streaming with distributed capabilities, where data are organized by topics which represents different regions. This pipeline marks the end to end journey from video input to the DeepStream SDK that decode, inference and publishes the result. Next step, we will discuss how is this result captured and processed by the backend server.

5.1.2. Backend

The backend server is an integrated system consisting a Kafka Consumer, a web server, RTSP transcoder, and a Redis container. Their functionalities are explained below.

Kafka Consumer

The Kafka consumer is responsible for listening to new updates from the Kafka cluster in the DeepStream SDK where the inferencing results are stored. Whenever new information is available from the cluster, the data will be pulled from the cluster and published to the Redis container. This arrangement allows future expansion for a distributed system where there can be multiple inferencing and backend server and each only listen to the topic they are assigned to.

Web Server

Gin is the chosen web framework with production grade performance and good middleware support. The web server connects the frontend and the Redis container by allowing the frontend to get updates through a WebSocket. The WebSocket provides a layer of abstraction such that the frontend can have a simple way of getting real time updates. The backend is actively listening for WebSocket handshake request that is initiated by a frontend client.

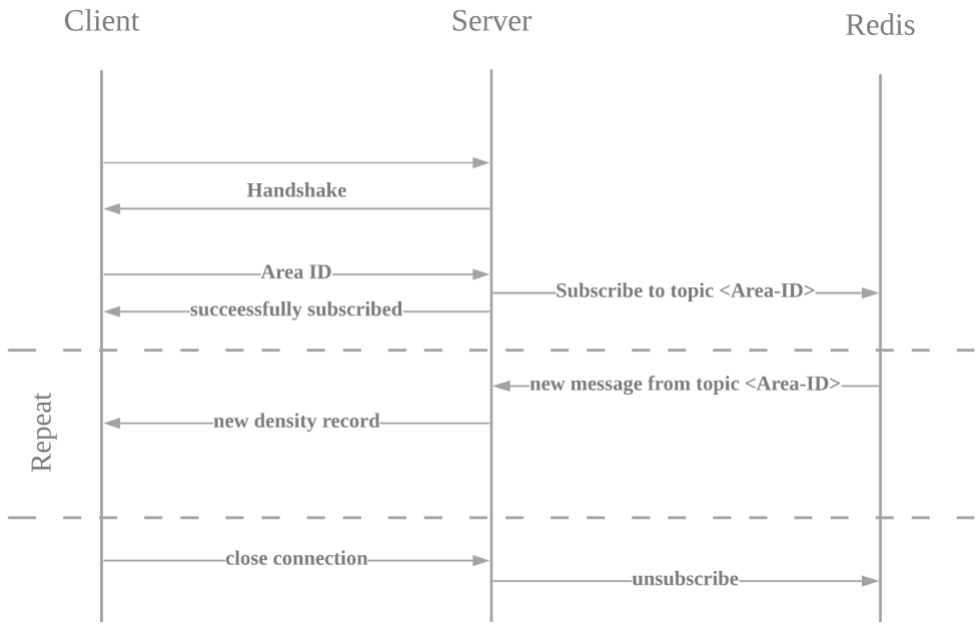


Figure 5.2 WebSocket Sequence Diagram

In our design, the frontend client establishes a WebSocket connection with the backend with the procedure as shown in Figure 5.2. After the handshake phase, the client will send the area IDs it intends to get updates from. The server will fetch from its cache or subscribe to the Redis container if it hadn't done so on the area IDs the user requested. Under this arrangement, users will get real time update whenever new inferencing result are made available.

Redis Container

The Redis container is a Redis instance run inside a container with the image pre-packaged by Redis. It is a key value storage with good performance for real time streaming application.

RTSP transcoder
Real time video feed of the annotated inference result is intended to be streamed on the front-end web application. As the video available straight out of the DeepStream SDK is of RTSP format, since there is no mainstream support for streaming RTSP video natively on the web browser, the video is first transcoded to MPEG format before being streamed to the web browser via WebSocket.

5.1.3. Frontend

Data visualisation provides user with a way of intuitively understanding the value of human movement statistics by seeing it for themselves. Easy access is also a serious concern as no one would like to download and install an application when they can easily access all the result using a web browser. Therefore, we have chosen to develop a web application using the React framework. In the development perspective, not only this provides support for most platforms using a single codebase, it will also be beneficial for future development where updates can be directly pushed to the end user through CI/CD, making it much easier to publish updates and new features.
As for prototyping purpose, two types of information will be available on the web application. The first is a real time heat map and another is a real time annotated RTSP stream coming from the backend server which transcoded the stream to a web friendly format. Live data are presented for operational awareness to the situation development on scene, such that operators and staff can be informed in advance and react to situations proactively.

Real-time heatmap
Heat map is an obvious choice for displaying people density data in an intuitive way as along as the viewer as some basic understanding of the floorplan. To display the user's own map, they can first convert their floorplan into a GeoJSON format which is a standard way of storing data points in terms of groups of coordinates, with each group representing a structure. The generated GeoJSON file can then be uploaded onto the backend server, where other users will also be able to access it. The location id on the GeoJSON will have to correlate to the id stored in the backend server. Figure 5.3 is an illustration of a heatmap using the map of Hong Kong separated by districts.

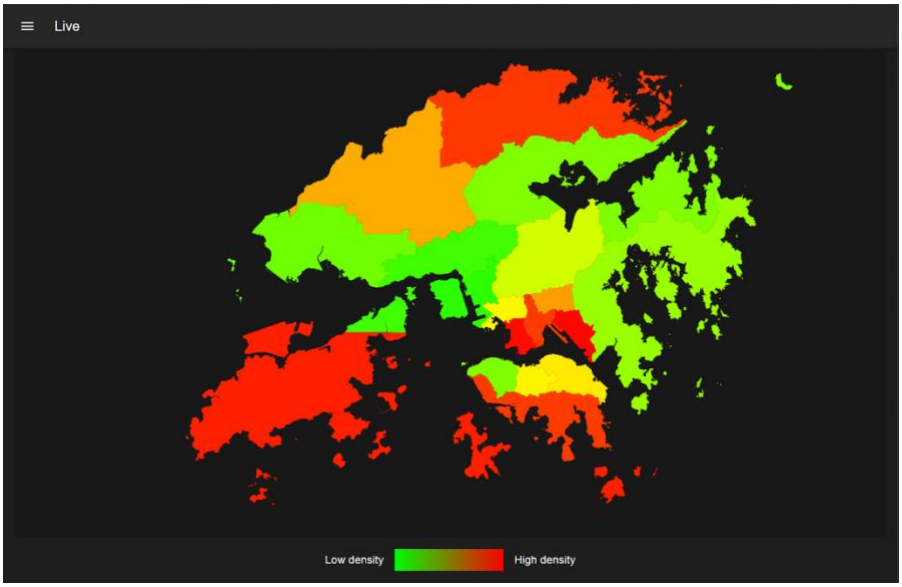


Figure 5.3 Density heatmap

Users can quickly gain overall situation awareness without having to look at any numbers or words. With a simple green to red colour gradient representing low people density to high people density, users can determine for themselves if an area is too crowded or there exist some sort of bottleneck causing one area to be too crowded and its adjacent area to be much less crowded. If user is interested in the actual statistic of the area, they can simply hover over to get the occupancy count of the region as shown in Figure 5.4 below.

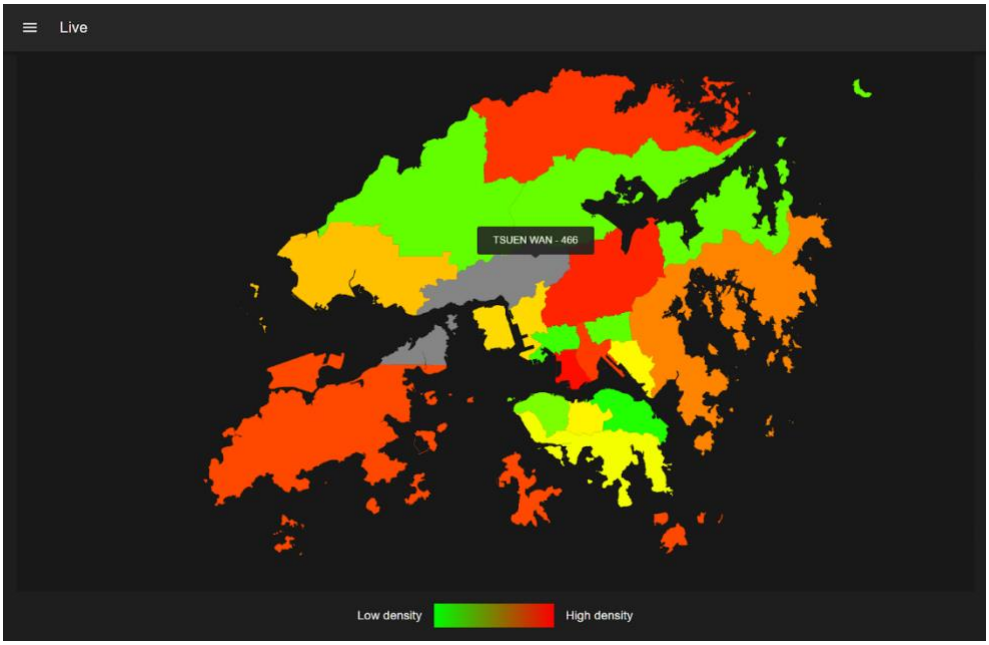


Figure 5.4 Density heatmap hover state

Currently, all the data presented are live data so users can view live changes and react accordingly. To achieve this, the frontend establishes WebSocket connection with the backend and get live updates of new count result.

Annotated RTSP stream

Another feature of the frontend is to provide the annotated RTSP stream output by the DeepStream SDK after the result is inferenced. The transcoded stream from the backend is made available also through web socket and displayed onto the users screen. Figure 5.5 is a screenshot of a working example of how the stream is annotated. Through the live RTSP stream, user can get the entry number over a certain period of time, as well as understand if the detection is working as expected. In case the queue direction is changed or the model struggle to identify humans in the scene due to reason such as fault in the camera equipment, it is obvious to the user through viewing this annotated stream that something is not working as expected and fixes can be applied as soon as possible.

Figure 5.5 Annotated RTSP Stream

6. Challenges

The following chapter will discuss the challenges faced in this project, as well as their respective solution and workarounds adopted.

6.1. Data collection

Data collection in this project was not as straight forward as expected due to the requirement imposed to get CCTV-like footage that captures scenes of people queueing up. Due to privacy concerns, CCTV footages are uncommon, let alone footages that consistently capture a large crowd. Even if some CCTV footages are available, due to the unknown position and height of the camera, it is impossible to do comparative analysis based on the distance, height and angle to the subject. Furthermore, footage of people queueing up is also hard to come by in open datasets. Therefore, the only real possible way is to do manual data collection.

Due to Covid-19, queuing activities in the open inside Community Testing Centres are commonplace. To be able to adjust the height, distance and angle of the camera, the only real choice is to use a drone for the data collection. Care has to be taken in order to ensure the safety of operating the drone, and location where this can be safely done are limited. Additionally, difficulties also arise as the drone experiences turbulence during the capturing phase, causing the point of reference of the frame to drift significantly. This made the queue counting analysis difficult as the counting base on a static line on the frame, when the camera perspective moves, the line will be moved relative to the scene. Video stabilization technique has thus been employed to stabilize the footage and only footage with less drift are selected as a result.

6.2. Human tracking for queue counting

As explained previously, density map has an advantage over bounding box regression due to its ability of identifying a human subjects who only take up a few tenths of pixels in the scene while it takes a minimum of few hundred pixels for a bounding box regression method to work. However, when trying to do tracking and queue counting using a density map result, it was determined that it is not likely to generate any meaningful result due to two main reason. Firstly, the output of a density map is too sparse, meaning that looking at the density map it is impossible to pinpoint each individual human. This is due to the inherent design of density map where each annotation is not a representation of an individual human, but the radial probability of the pixel being part of a human (see Figure 6.1). Secondly, no unique identification can be given in a density map result. As soon as two persons overlap, just by looking at the probability density map, it is impossible to identify which dot originates from which dot since there are no individual identification for each subject. Therefore, combining these two inherent limitations, while queue counting might still work if there is a tracker that is able to reliably track dots and the model is well trained enough that inference output are not sparse, individual tracking is impossible as differentiating information are discarded in the inference process.

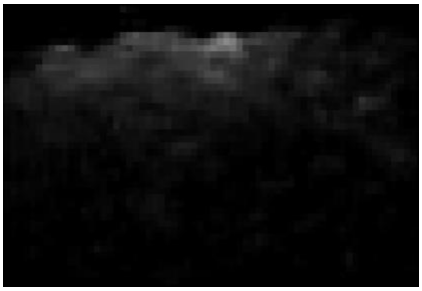


Figure 6.1 Density map Sparse result

Therefore, for tracking and queue counting purposes, the only method that produces meaning result would be the well tested bounding box regression method. However, the reason why normal bounding box regression failed to provide meaningful reason in the first place is due to camera footage captured being too far away, such that the pixel count representing one human is not sufficient for the model to detect and identify its existence. As a result, we review the effectiveness of queue counting on scaled footage which help understand what the other requirements for good counting accuracy.

7. Conclusion

Project Background and Objective

People movement intelligence is the centrepiece to the Smart Airport Vision introduce by AAHK as one of the prominent solution in the key enabling technologies of big data intelligence [3]. Our project aims to develop an end to end prototype that helps showcase the significance of the project through intuitive visualization. At the same time, evaluation of state-of-art models and detection methods options allow us to provide recommendation from model adoption to hardware choice, especially for cases where footages can only be captured far away. Finally, MLOps best practices relevant to this project including the use of DeepStream SDK as well as configuration options are also outlined in this report.

Methodology Adopted

The project methodology in this report mainly concerns about MLOps strategy and system integration.

Regarding MLOps strategy, key components include data collection, model evaluation and the use of DeepStream SDK. Online dataset together with data collected manually were used in the project for evaluation. The online dataset was mainly used for comparative analysis on the accuracy of density map generation across 5 different models. Data collected manually was done through video capturing using a drone and is primarily used for evaluation of different configuration. Configuration options includes scaling, distance, and angle, and its effect on queue counting accuracy are evaluated. Since the data is collected manually, manual annotation and counting is required, while the online dataset comes with ground truth data for much easier evaluation. Finally, the DeepStream SDK serve as the cornerstone of the MLOps pipeline. From handling and transcoding data input, inferencing, clustering, tracking, to ultimately publishing the result, the DeepStream SDK provide performant plugins that can be utilize easily. The DeepStream SDK is used in both the model evaluation process as well as the development environment that integrates with the prototype system.

As for the system design, the whole system mainly consists of three components, namely 1. Inferencing platform, 2. Backend, and 3. Frontend application. The inferencing platform in this project is managed by the DeepStream SDK and emulates incoming RTSP stream to the inferencing server. Inferencing result will then be passed on to a Kafka cluster where the backend server can subscribe for new changes. When data is requested from the frontend server, information is fetched from a Redis container which stores result obtained from the DeepStream SDK Kafka cluster. The backend server is written in Golang using the gin web framework which is responsible for serving all request. The frontend web application is written in React Typescript and queries the backend for live data, where this communication link is handled by WebSocket. The prototype allow user to view the live density data as well as the annotated RTSP stream through the web application.

Key Findings

Key findings regarding 1. Model evaluation on density map generation, 2. Factors that affects bounding box regression accuracy, and 3. Implication on hardware and image requirement are summarized below.

Five models were trained and compared for their accuracy in generating density map for a crowd counting assignment. The accuracy is measured simply using MSE and MAE by comparing with the ground truth provided. Performance of different model varies between datasets as each dataset feature different type of scenes. VGG and AlexNet seems to be superior when it comes to generating density map of scenes which the camera is closer to the subject and each subject is relatively larger. However, when it comes to crowded scenes captured far away, RestNet family models provides better accuracy. Therefore, depending on the usage, different models can be selected and there are no conclusive answer. It should also be noted that the models were not trained thoroughly so improvements are definitely possible. Furthermore, the magnification factor that applies to the ground truth density map also plays an important role in the training speed of different models.

Accuracy of bounding box regression is found to be strongly correlated with the minimum pixel count that represents a human, as well as the amount of details that is exhibited by each subject. This also links to the requirement in hardware, image quality, and camera positioning. Basically, the bare minimum required to track a human is found to be around 300 pixels. Reasonable accuracy can be expected if humans are of at least 500-600 pixels while satisfactory accuracy can be expected at around 900 pixels. Users should purchase hardware that is able to capture humans around this resolution according to their accuracy expectations. Moreover, it was shown that scaling up the footage as a pre-processing step before inferencing do improve detection accuracy, though sufficient details that helps identify a human is still required for the model to do the classification. As long as the model can detect the subject continuously, accuracy of queue counting remains satisfactory. Therefore, not only this implies the optical system has to possess reasonably good quality, the positioning of the capturing device should also be less than 45° incident to the subject for avoiding overlapping of subjects. Overall, we are satisfied with the accuracy of the PeopleNet model in terms of people detection and queue counting when tasked with inferencing scale up footage that is captured far away from the subject.

Significance of Prototype and Findings

We hope that this project serves as a practical guide for interested parties to develop, integrate, and add visualization capabilities to their own people movement intelligence solution. We have discussed practically how the prototype system is integrate together using DeepStream, gin web server, and with a React frontend web application. Moreover, to specifically address AAHK's concern of their CCTV too far away for such detection analysis, we compared the accuracy of density map generation and bounding box regression based on their use case and determine that bounding box regression on scaled up footage will the preferred approach for obtaining meaningful and accurate result. Furthermore, factors that affects accuracy are documented and these comparative findings leads to the practical suggestions proposed for AAHK or any other party to determine the hardware required, positioning of capturing device, and possible ways to increase detection and queueing accuracy.

Concluding Note

Overall, we would like to reiterate the importance of capturing people movement intelligence especially for service-oriented business. As the old Chinese saying that intelligence wins wars, intelligence is just as valuable as gold in the business world. Understanding how your customer move, knowing their preference, figuring out the bottlenecks in your system, gain insight that help optimize your workflow, so on and so on, are absolutely key to business success. To have any meaningful efficiency in capturing this information at a scale as large of HKIA, automation is the only feasible way. Therefore, AAHK, as well as other business, should consider investing more resources in this regard by utilizing the findings in this report and use it as a framework to develop their own system that captures people movement intelligence according to their own business requirements. Finally, we hope AAHK can gain insight to their own operation and stay competitive through constant improvement.

8. Future Works

During the development phase of the project, we have identified multiple features or modifications that would boost the utility of the system remarkably. These mainly involve prediction, improving accuracy, and queue identification.

8.1. Prediction

Prediction extends beyond the scope of understanding the current situation to provide estimation in demand of the near future, as well as allow users to plan for situations they have yet to experience. Estimation of demand in the near future helps identify potential bottlenecks as well as excessive demand, allowing AAHK to proactively allocate the necessary manpower and resources before the situation goes out of control. Aside predicting the near future, one of the major operating concern is simulating a lot of "what if" scenarios. These scenarios may be due to unexpected events such as typhoons, or expected scenarios such as public events or even future growth. There is no trivial way to predict how the facilities will handle the extra demand and if the demand exceeds the capacity. Therefore, if there exist a way that can model human movement and servicing capabilities in the airport, operators would only need to plug in the numbers to obtain a rough estimate of how the flow would look like. Understanding the weak points allow focused resources and attention on areas that needs the most improvements. One possible way for doing such prediction would be using utilizing LSTM, where it will perform prediction on density and queuing time series data and have integrate it into flow simulation software for more accurate predictions.

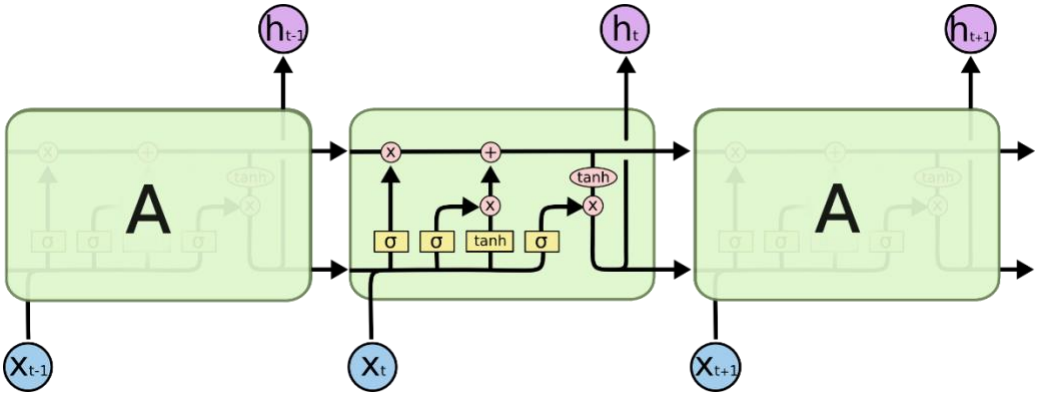


Figure 8.1 LSTM architecture [12]

8.2. Additional research with comparative analysis

8.2.1. AI upscaling

Additional research can be done to improve the accuracy of human detection even further. As we have established that human identification also depends on the details and features exhibited by each human subject, to increase the image quality without changing the hardware, it is very much possible that pre-processing with an AI upscaling model could greatly improve the identification accuracy. AI upscaling does not simply add more pixel to a frame, but also add detail according to

what the model is trained for (see Figure 8.2 AI upscaling illustration [13]). For example, if the upscaling model is trained specifically in upscaling scenes where there are lots of humans, such a specialty model may provide even more details than a generic upscaling model. This overall strategy of upscaling before inferencing will be particularly useful if upgrading existing hardware or installing new hardware in favourable location is not feasible.

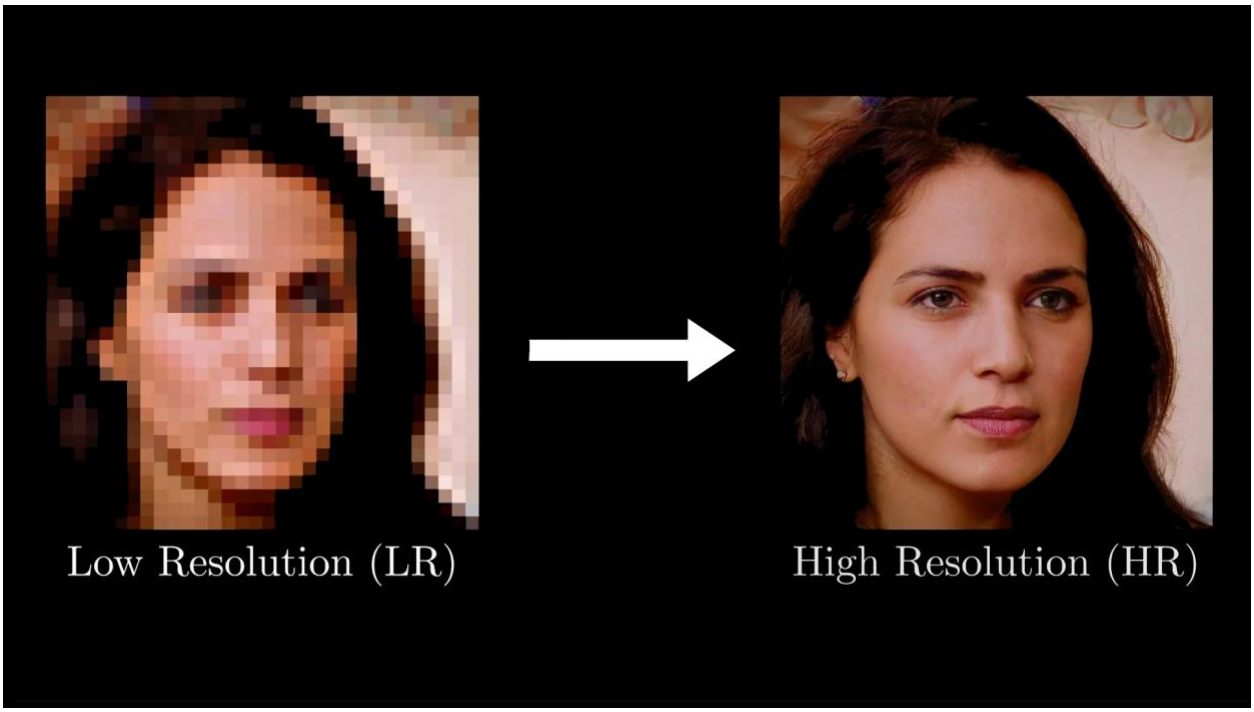


Figure 8.2 AI upscaling illustration [13]

8.2.2. Effects of reduced framerate

Cost effectiveness is a major concern in this project as it greatly affects the net utility and scalability of the project, considering the heavy processing power and storage required. Further analysis should be done to survey what is the minimum framerate required for a tracker to consistently track a subject that is of a certain size and speed. Reducing the framerate requirement will massively reduce the cost required for the inferencing infrastructure and even storage for storing all the inferencing results. With 30 frames per second, there are 30 frames to be inferenced and 30 results to be stored for each camera. With thousands of CCTV cameras in HKIA, this will not be scalable at all. If the frame rate can be reduced to a few frames per second without sacrificing tracking accuracy, it will massively reduce infrastructure cost.

8.3. Queue Identification

Queue identification is another possible way of increasing the utility of the system. With the current prototype, only the number of persons in the frame as well as the count of how many exited the queue is available. To know how many persons is in a queue and how many are just passing by, automatic identification of a queue is necessary. One possible direction will involve clustering

techniques that separates people queueing and passing by. Due to the complexity of operations in HKIA, the direction and position of the queue are constantly changing to handle fluctuating demands, with only the head of queue remaining relatively static. Therefore, it is not feasible to simply crop out the area of the queue. To automatically identify such a queue, the area head of the queue which remains relatively static will first be manually predefined, and all subjects inside this area will be considered to be part of the queue. After having the people inside this area as the primary population of a cluster, clustering based on this group, i.e. all the people linked directly or indirectly, will also be considered to be part of the queue. Therefore, using this clustering technique, the queue can be identified automatically even when the queue is extend or moved outwards. Future research of use of clustering algorithms and tuning of relevant parameters can be done for better identification accuracy.

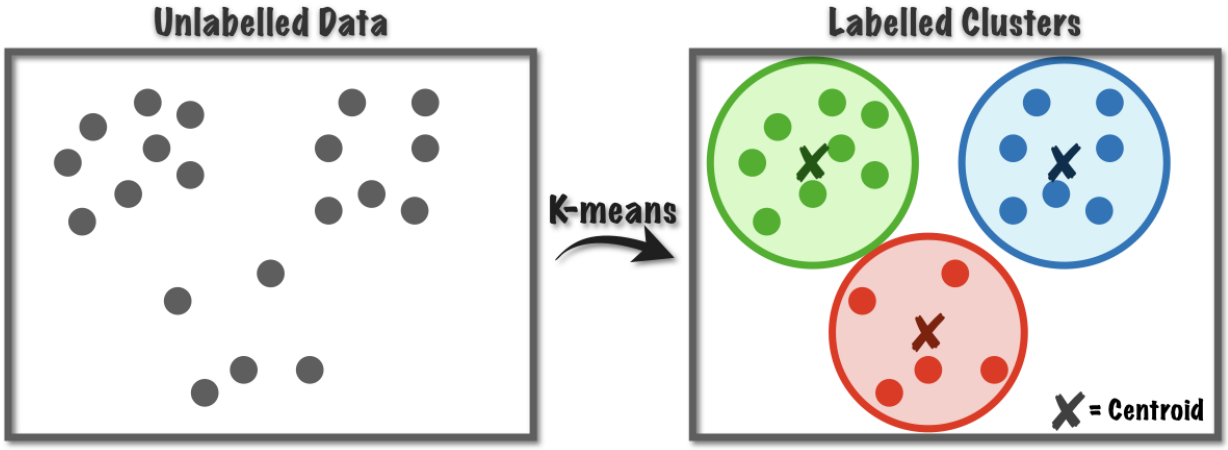


Figure 8.3 Clustering illustration [14]

9. Bibliography

[1] "AAHK Sustainability Report 2018/19," Hong Kong International Airport, [Online]. Available: https://www.hongkongairport.com/iwov-resources/html/sustainability_report/eng/SR1819/airport-city/smart-airport-city/. [Accessed 29 September 2021].

[2] "A Smart Airport Experience - Hi-Speed Wi-Fi + Indoor Patrol Robot," Hong Kong International Airport, 27 November 2019. [Online]. Available: https://www.youtube.com/watch?v=KlwO0M49hkA&ab_channel=hkairportofficial. [Accessed 29 September 2021].

[3] "Hong Kong International Airport Overview," [Online]. Available: https://www.hongkongairport.com/en/the-airport/hkia-at-a-glance/fact-figures.page. [Accessed 15 4 2022].

[4] I. A. Review, "e-Security gates to be implemented in Hong Kong International Airport," 28 09 2018. [Online]. Available: https://www.internationalairportreview.com/news/75655/e-security-gates-hong-kong/. [Accessed 16 April 2022].

[5] Nvidia, "DeepStream SDK," 2022. [Online]. Available: https://developer.nvidia.com/deepstream-sdk. [Accessed 16 April 2022].

[6] "TensorRT Integration Speeds Up TensorFlow Inference," Nvidia Developer, 27 March 2018. [Online]. Available: https://developer.nvidia.com/blog/tensorrt-integration-speeds-tensorflow-inference/. [Accessed 29 September 2021].

[7] ClockInstant, “Smart Factories: Indoor Positioning Heatmap,” 2019. [Online]. Available: https://clockinstant.com/smart-factories-indoor-positioning-heatmap/. [Accessed 29 September 2021].

[8] D. HAN, “VGG16 学习笔记,” 2018. [Online]. Available: http://deanhan.com/2018/07/26/vgg16/. [Accessed 22 January 2022].

[9] Q. Ji, J. Huang, W. He and Y. Sun, “ptimized Deep Convolutional Neural Networks for Identification of Macular Diseases from Optical Coherence Tomography Images,” *Algorithms,* vol. 12, p. 51, 2019.

[10] Y. Zhang, D. Zhou, S. Chen, S. Gao and Y. Ma, “Single-Image Crowd Counting via Multi-Column Convolutional Neural Network,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 589-597, 2016.

[11] A. S. Krizhevsky, I. Hinton and G. E., “ImageNet Classification with Deep Convolutional Neural Networks,” *Proceedings of the 25th International Conference on Neural Information Processing Systems,* vol. 1, p. 1097–1105, 2012.

[12] C. Olah, “Understanding LSTM Networks,” 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed 16 April 2022].

[13] S. Wyndham, “ Decline red-shark-logo PRODUCTION POST & VFX AUDIO TECHNOLOGY CINEMATOGRAPHY JOBS MORE SUBSCRIBE This new AI system could be science fiction, but it’s very real indeed,” 2020. [Online]. Available: https://www.redsharknews.com/this-new-ai-system-could-be-science-fiction-but-its-very-real-indeed. [Accessed 16 April 2022].

[14] J. Jang, “ Decline red-shark-logo PRODUCTION POST & VFX AUDIO TECHNOLOGY CINEMATOGRAPHY JOBS MORE SUBSCRIBE This new AI system could be science fiction, but it’s very real indeed,” 2022. [Online]. Available: JaeHoon Jang. [Accessed 16 April 2022].