APPLICATIONS FOR SMART AIRPORT

PEOPLE DENSITY DETECTION

FINAL REPORT

Supervisor:

Dr. Chim T W

Student:

Yuen Cheuk Heng 3035553587

[Experiential Learning Project] Applications for Smart Airport

The University of Hong Kong

Table of Contents

Table of Figures

Table of Tables

List of Abbreviations

Abbreviations & Acronyms	**Definition**
AAHK	Airport Authority Hong Kong

AI	Artificial Intelligence
BBOX	Bounding Box
CNN	Convolutional Neural Network
ER	Entity Relation
FLOPS	Floating-point operations per second
GB	Gigabtye
GPU	Graphics Processing Unit
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
MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
NMS	Non-maximum Suppression
RNN	Recurrent Neural Network
RTSP	Real Time Stream Protocol
UI	User Interface
VGG	Visual Geometry Group

1. Project Background

The Airport Authority Hong Kong (AA) is particularly interested in passenger movement information as it works toward its goal of transforming Hong Kong International Airport (HKIA) into a Smart Airport. Big data intelligence is a critical piece of technology that can help achieve the objective of providing high-quality passenger services, increasing operational efficiency, and directing future management plans.

1.1. Big Data Intelligence

One of the five major enabling technologies identified in the Smart Airport Vision as a foundation for realizing the value of passenger movement intelligence is big data intelligence. Instrumental performance indicators may be derived via data collecting, artificial intelligence, and machine learning algorithms, which can assist identify hidden patterns and trends. Patrol robots that work independently and check WiFi signal quality around the terminal are one of AA's outstanding initiatives. Following the spirit of this initiative, each and every aspect of the HKIA operation may be examined and enhanced to varying degrees. Given the high complexity of HKIA operations, big data analytics is the only viable way to undertake ongoing and successful improvements, as the process can be done automatically with minimum human participation.

Hong Kong International Airport serves about 71 million people each year, making it one of the world's busiest airports. It becomes a crucial but tough duty to accurately grasp the intake and movement of people at the terminal. Passenger flow at the terminal is very hard to predict due to human psychology, weather conditions, and even seasonal swings. Instead of forecasting passenger flow based on current passenger numbers, real-time intelligence collection would give far more precise information for optimization.

An efficient people movement detection solution is required as a vital element in this big data intelligence architecture as a mix of big data intelligence with the goal of improving operational efficiency.

1.2. Usage of People Movement Data

With the information of people movement may provide operational and management staff a significant edge as they seek to optimize their operations. In practice, this entails things like better marketing and management strategy, labor force approximation, resource allocation, and more. AA staff can better comprehend the genuine passenger flow of a business by considering market and management tactics, which increases the accuracy of land value estimation. This also enables for more effective facility distribution and offers a better structure based on real traffic. Approximation of the labor force and resource allocation are two further factors to consider. There are several waits in the airport that serve as a buffer to demand fluctuations, including check-in, customs, lounges, and boarding. Understanding the service demand and queue flow rate may lead to significant improvements in customer satisfaction by allowing resources to be swiftly allocated to where they are most required, decreasing wait time and potential passenger delays. Knowing the length of the wait and how it is progressing allows the operational team to set up or remove queueing equipment for better ordered queuing patterns.

1.3. Proposed Approach: Deep learning Inferencing

Traditionally, manual data collection and employee expertise have been used to acquire this type of business insight. While such a strategy has certain advantages, it also has some significant disadvantages. To begin, it necessitates time-consuming counting and estimation of passenger flow in order to cover all parts of the vast concourse. Second, precision is substantially inadequate since counting moving people and comprehending their movements is a difficult task for a human to do. Finally, owing to the time and cost needed in surveying such a large region, the breadth and frequency of this form of data gathering cannot be continued at a high level. To summarize, manual counting's major flaw is its inability to scale up at a reasonable cost.

As a result, to reliably recognize and track people in a scene, an automated and scalable solution is necessary. Despite the importance of acquiring passenger movement intelligence, there is currently no mature solution for such applications, particularly when a camera cannot be placed near enough to the site of interest. The high ceiling of the HKIA makes it very hard to cover regions by having no choice but to use very high above cameras. In addition to analyzing the efficacy of existing human counting approach, we would explicitly examine their accuracy in this circumstance when the camera overhead is at a long distance from the human.

1.4. Integration for Visualization

While the project is primarily focused on implementing and analyzing the performance of people counting and movement detection technologies, visualization is still an important component. It enables AAHK personnel to assimilate multi-dimensional material in a compact, information-rich, and visually appealing manner. Given the enormous area covering the HKIA premise, visualization allows AAHK employees to easily monitor data in all areas or a specific region, at a specified timestamp or in a continuous manner. It is hoped that the visualization program's intuitive usage and presentation would make it simple to get started and will not necessitate expert experience. The prototype will focus on delivering real-time queueing information as well as real-time passenger volume statistics by area, all of which will be accessible and viewable via a web application. This web application should assist users in comprehending the possibilities of business intelligence and may also recommend other metrics to be counted or evaluated as a system enhancement.

1.5. Project Contribution

As big data become more important, big data intelligence in terms of people movement is becoming a major topic of interest for any physical service-oriented business. It would be beneficial for business management to know about their customer and operation. Because there are now no mature solutions that handle such business needs as people detection using footage shot far above the head or queue statistical analysis, our study intends to combine such solutions and assess the efficacy of alternative models under various scenarios.

1.6. Report Outline

This report is arranged into five chapters. The first chapter introduces the benefits and approaches of crowd counting. The objectives and significance of the work within and outside the project are also discussed. The second chapter simply outlines the project objective.

The third chapter details the methodology of the project. The Machine Learning Operations (MLOps), data visualization techniques, and finally the whole system architecture would be discussed in detail as to how it supports the long-term operation of this system.

The fourth chapter outlines the experiment and result of model survey, DeepStream queuing output accuracy and Analysis of human size inside a frame, as well as the system design.

The fifth chapter discuss the challenges we have faced and how we tackle them.

The sixth concludes the report. Outlining the motivation, methodology, and fundings.

The seventh and final chapter discuss the future works that can be extends from 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 inference result
- Evaluate effectiveness of human tracking and queue counting
- Apply MLOps techniques in model optimization and data management to obtain results for evaluation
- Visualize density data in heatmap as well as queue statistics on web application
- System integration of MLOps component, backend and frontend as a prototype solution

3. Methodology

To successfully enhance operational efficiency using Big Data Intelligence, thorough preparation is essential to work out the application implementation architecture and build up a project-specific Machine Learning Operations (MLOps) strategy. There are three sections to the topic. The first section covers the methods and scope of data collection as well as the business logic for data processing. The second section goes over the many components of Deep Learning and MLOps that were used in the project, including data management, model learning, model verification, and deployment. The technique to data visualization and presentation is discussed in the third section.

3.1. Software implementation

The CCTV video streams will be fed into the machine learning/deep learning model. After that, the model will identify whether or not humans are present in the video stream. The bounding box(es) will next be drawn to include each individual human and to determine that individual's center point. The density data will be presented as a heatmap on a web application after density analysis of the frame (e.g. linear transformation), where users with access permissions can see the data with any suitable device. APIs with critical performance indicators will also be available.

3.2. Density Detection using Deep learning

Deep learning is utilized to perform person detection and bounding box calculation in order to implement density detection. Deep learning is part of the machine learning family that includes representation learning and neural networks. Neural networks are computer networks based on biological neural networks, which are comparable to those found in the human brain.

Deep learning is a supervised learning approach that learns existing features by establishing a mapping between input and output. This form of learning necessitates feeding a dataset to the neural network for training that captures essential characteristics inside the frame. To maintain the model health, a robust mechanism is required to govern the system. The MLOps cycle is used to accommodate to our low latency streaming techniques. This cycle is constructed with four parts. 1. Data Management 2. Model Learning 3. Model Deployment [1].

3.2.1. Data Management

The performance of a deep learning model is governed by the network's architecture and the quality of the data. Data has long been a stumbling block to enhancing model accuracy. Data management is a crucial step in improving data quality and debugging data. Data management also makes certain that the dataset has all of the essential embeddings and characteristics. The term "embedding" refers to a low-dimensional vector representation that captures linkages between higher-dimensional data and the input data.

In data Management, there are four essential steps. 1. Data Collection, 2. Data Preprocessing 3. Data Augmentation, 4. Data Analysis.

Data collection

Data Collection will be divided into two parts, leveraging existing datasets and manual data collection. To leverage existing datasets, we have selected an online dataset that was used in a crowd counting project. As for manual data collection, currently access to HKIA is restricted during Covid, therefore our project testing site will be in HKU InnoWing. Since InnoWing environment is different from the airport, the data collection process will be recording video on a date that the site has one or more events, simulating the scenario where a flight is going to take off.

Data Collection will be split into two parts: using existing datasets and manual collected data. We chose an online dataset that was utilized in a crowd counting effort. Due to the fact that access to HKIA is now prohibited during Covid, our project testing location will be a community testing center in Hong Kong where large number of people and long queue can be found there.

Data Preprocessing

We need to label every person in a cartesian coordinate system in every frame for our project, therefore data preprocessing entails creating a one-to-one mapping. As a result, Deep Learning detects and captures the hidden characteristic of the photographs.

Data Analysis

Data analysis is a necessary component of data collection and data augmentation. This stage ensures that the data is not damaged and performs data cleaning before feeding it to the model, for example, to guarantee that there is no class imbalance and that augmented images based on deviating photos are appropriately created.

3.2.2. Model Learning

Model Learning is a crucial step in deep learning that allows computer neural networks to extract feature in a image. latent features are 'hidden' features that is hard to be observed by human. It is ideal that the model can extract important feature from the training data and understand the relationship between input and output. More important, the model should also perform well on unseen datasets.

In Model Learning, there are four major steps: 1. Model Selection, 2. Model Training, 3. Transfer Learning, 4. Hyperparameter Selection.

Model Selection

This stage seeks to discover the best model to reproduce the one-to-one mapping inside the training set among some of the state-of-the-art deep learning model, including VGG, RestNet32, RestNet50, and People Net. The selection will be made based on the training and validation accuracy.

Hyperparameter Optimization

Hyperparameter optimization refers to choosing the optimal hyperparameter for a mode in the training stage. Different constraints, weights or learning rates can be applied to a deep learning model, and result in different model performance and efficiency. The optimal hyperparameter can only be found by experiment as deep learning is still a black box process. For example, the model's accuracy in detecting humans depends on the learning rate, which is a hyperparameter that allows the model to cover local minima. Hyperparameter Selection is very important, and it could help the development of a deep learning model a lot.

3.2.3. Model Deployment

The ML cycle's purpose is to create a production-ready model and put up the right environment for it to match the business needs. It is critical to first comprehend the commercial and technical factors that drive pipeline and infrastructure selection.

The first and most important consideration is cost effectiveness. Because the airport is covered by hundreds of cameras, inferencing on all of this material would need a significant amount of infrastructure and labor for planning, capturing, inferencing, and provisioning. The potential investment and operating costs of running this system indefinitely would be enormous, therefore finding a cost-effective solution is vital to the project's success. As a result, we're aiming to create a system that's light and does not require a lot of human resources to manage. The second point of concern is flexibility, because the airport industry is rapidly growing, our solution must be forward-thinking and able to adapt to new requirements without fundamentally altering the present system. Model modification, scalability, and ease of deployment in various contexts are among them.

We will go through the pipeline adopted for the model deployment.

DeepStream SDK

To accomplish the intended goals, we would need a framework that provides a layer of abstraction to handle all of the tedious details while we concentrate on model development and application. Nvidia's DeepStream SDK was chosen for this purpose. The Deepstream SDK will communicate with the lower-level SDK and hardware in order to give a level of abstraction while maintaining acceptable performance. This is because, while the Deepstream SDK's pipeline is similar to those of other machine learning pipelines, Deepstream has native hardware accelerated plugins for simple integration. In practice, this means we simply need to use all of the SDK's plugins and create our own unique apps on top of them, which cuts down on development time and increases flexibility.

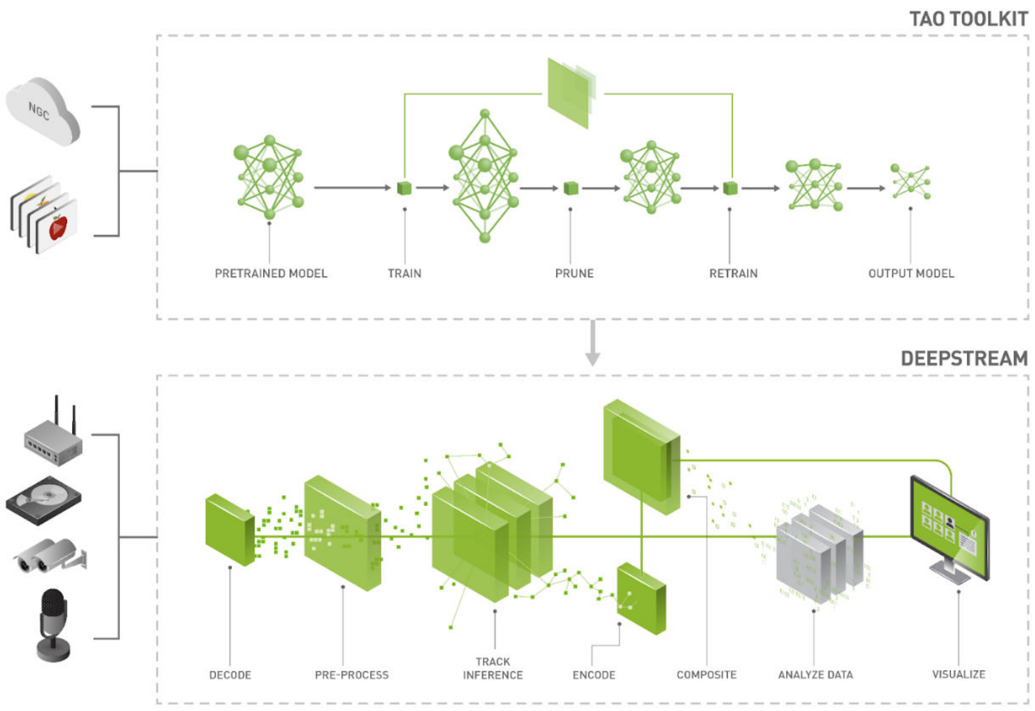


Figure 3.2.1 Deep Stream Brief Summary [2]

Tao Toolkit and the Deepstream App are two essential components of the Deepstream SDK.

Tao Toolkit intends to speed up the development of AI models by allowing for model flexibility, transfer learning approaches, and model optimization for quicker inference. The Tao toolkit also includes model retraining, which allows the model to be readily updated and adapted.

GstStreams-based Deepstream app is an all-in-one Vision AI framework. It is simple to configure to accept a large number of video sources and do inference. Additionally, with optimization settings, the model deployment is done automatically. Deepstream additionally has an analysis module with features such as region-of-interest filtering and counter threshold alert.

Processing

The hardware will listen to RTSP video streams and decode the frames into the buffer using the onboard hardware decoder during the pre-processing phase.

The image will then be pre-processed if it is required. For instance, it will crop each frame differently to only contain important parts, hence conserving inferencing capacity.

Finally, frames from various streams will be automatically batch processed for improved performance. We'll use the gst-ninverfeserver as a plugin for the inferencing portion. As previously said, the given plugin will handle all liaising, so there is no need for complicated setup. The SDK has the ability to leverage the triton inference server for optimization. This plugin has the ability to optimize any model, allowing for speedy testing. Finally, the inference result would be clustered and tracked.

3.2.3.1. Additional Features

For hardware buffer sharing, it means only the memory references are transmitted during the whole process but not the copy of the memory, which result in better efficiency and less computational cost. For K8s containerize technology, it is a well-known scalable solution for deployment. The model can be updated through internet which could result in less model downtime, easier update and easy maintenance. Nvidia provided a lot of model such as PeopleNet and other deep learning model that can be easily deployment into the SDK. With minimum effort of replacing a model and twerking a few variables, this architecture may be modified to meet any requirements or capabilities.

3.3. Data Visualization

The data will be shown using a heatmap to better present the population density data to AA staff. A map in GeoJSON will be created initially for the heatmap. The map will then be divided into distinct areas. People density data may be shown on the map in an easily understandable fashion using intuitive color codes. Users may view the data in further depth by hovering over the region of interest. For a better user experience, hovering effects will be implemented. The map will be accompanied with a legend that explains the meaning of the various hues.

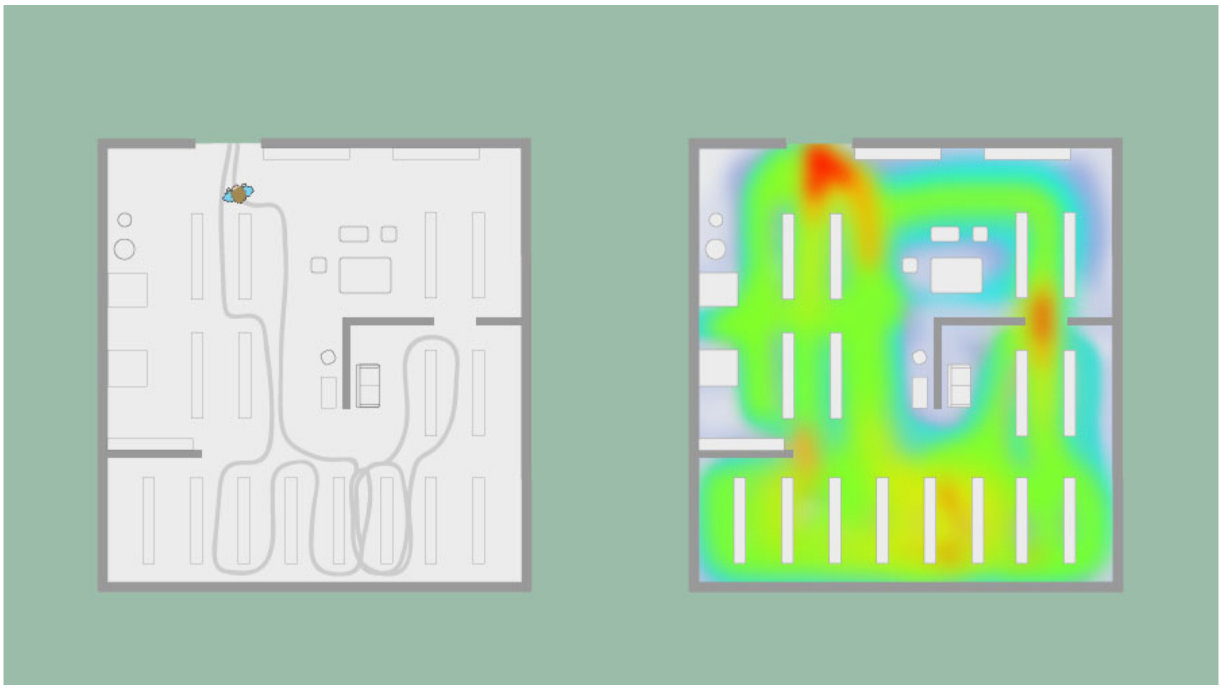


Figure 3.3.1 Example of heatmap [3]

D3.js, a JavaScript toolkit for manipulating documents based on data, will be used. It could deliver an interactive and real-time data visualization experience using the airport's layout and density data as input. D3.js is a fast and versatile framework that can handle massive datasets and dynamic interactions and animations with minimal overhead. It may be used in conjunction with HTML, SVG, and CSS to design our own map and view the data on it.

Frontend

React with Typescript is chosen for the frontend web application. React is a community-supported frontend library that is commonly used in web development. It's quick and light at the same time. Typescript is a strongly-typed superset of JavaScript that comes with a slew of added capabilities. It was selected to save time and avoid faults during development by catching syntactical mistakes early on.

3.4. Summary

The usage of MLOps tools in the project is described in the sections above. First, we went through the whole system's pipeline, which includes the machine learning component. The capabilities of

DeepStream and their advantages in our system are then explained in depth. Finally, a strategy is presented for visualizing the data and making it available to approved AA employees.

4. Experiments and Results

4.1. Model Survey: Density Map

This experiment aims to find out the model which is best suited for real-time crowd counting using the density map approach. The survey is conducted in the HKU CS GPU Farm 2 which has NVIDIA GeForce RTX 2080 Ti GPU and a CPU with 44 TFLOPS FP2 computational power.

Because of the computation power and the limited time, all models are only trained with 30 epochs. We chose the Pytorch to be the framework for training. It utilizes eager mode computing, therefore it can create neural graph dynamically which is also very helpful for debugging.

4.1.1. Dataset Selection

Two popular datasets are chosen for this experiment, including the ShanghaiTech Part B dataset and the UCF-QNRF dataset. The ShanghaiTech Part B dataset is a large-scale crowd counting dataset which is collected in busy street. While the UCF-QNRF dataset is famous for its high count crowd image and the wider variety of scenes. The specifications for the image are as follows:

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 4.1.1 Images specification of the dataset

In terms of resolution and count statistics, the discrepancies between UCF-QNRF dataset and ShanghaiTech Part B dataset are substantial, according to Table 4.1.1. UCF-QNRF has 14 times the number of persons as ShanghaiTech Part B, indicating that people density in UCF-QNRF is higher, whereas ShanghaiTech Part B offers a closer view angle.

4.1.2. Data Preprocessing

Data preprocessing is done for two major reasons. Because the NVIDIA GeForce RTX 2080 Ti only has 10GB of VRAM, the primary reason is to reduce VRAM usage. As the models and images are loaded into the GPU's memory, it is essential to first scaled down the image size before loading into the GPU as the resolution of the image in QNRF is relatively high. The second reason is to reserve VRAM for loading in the CNN model. There are models that contain a down-sampling layer with the requirement that the image resolution has to be a factor of 16, hence the images in QNRF are downsized to 1024 x 1024 while SHTB's image is not affected (768 x 1024).

For validation, ground truth version of density maps for both datasets are created using the kernel size of 15 and the Gaussian kernel function. It is for converting the centroid point of the human into a radial basis probability distribution with a total of area of 1

4.1.3. Training Options

To improve model accuracy, we applied a strategy, label normalization. The Magnification factor, according to the C3F Experiment, can be used to optimizing the model. After data preprocessing, a magnification factor is multiplied to the ground truth density map, resulting in a new density map. Below is the formula:

Equation 1 Magnification Factor Equation

$$New\ Density\ Map\ =\ DensityMap *\ MF \qquad where\ MF\ is\ Z$$

The magnification factor has a significant influence on the error rate in the C3F Experiment using the Resnet50 model because the model and dataset have their own distributions. If the difference is significant, it will be difficult for the model to converge to the dataset [4].

4.1.4. Evaluation Metrics

Mean absolute error (MAE) and mean squared error (MSE) are chosen as the evaluation metrics for this experiment.

The MAE equations are as follow:

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

The MSE Equation is as follow:

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

Where f_i is the predicted value, y_i is the ground truth.

4.1.5. Model Architecture

In this survey, the five models that are evaluated are VGG, ResNet50, ResNet101, MCNN and the AlexNet. They are all popular state-of-art CNN model. Their feature extraction performance is good and their result in competition are also excellent

4.1.5.1. VGG16

VGG stands for the Oxford University's Visual Geometry Group [5]. VGG16 has 16 layers that have weights. The model was pre-trained with over a million ImageNet photos of different categories, covering almost everything in the human field of view. VGG16 is usually used for general image recognition with an accuracy rate of up to 92%.

The first 10 VGG convolutional layers are retrieved and used as the encoder in this experiment, while 2 convolution layers make up the decoder. Below is a diagram of a neural network:

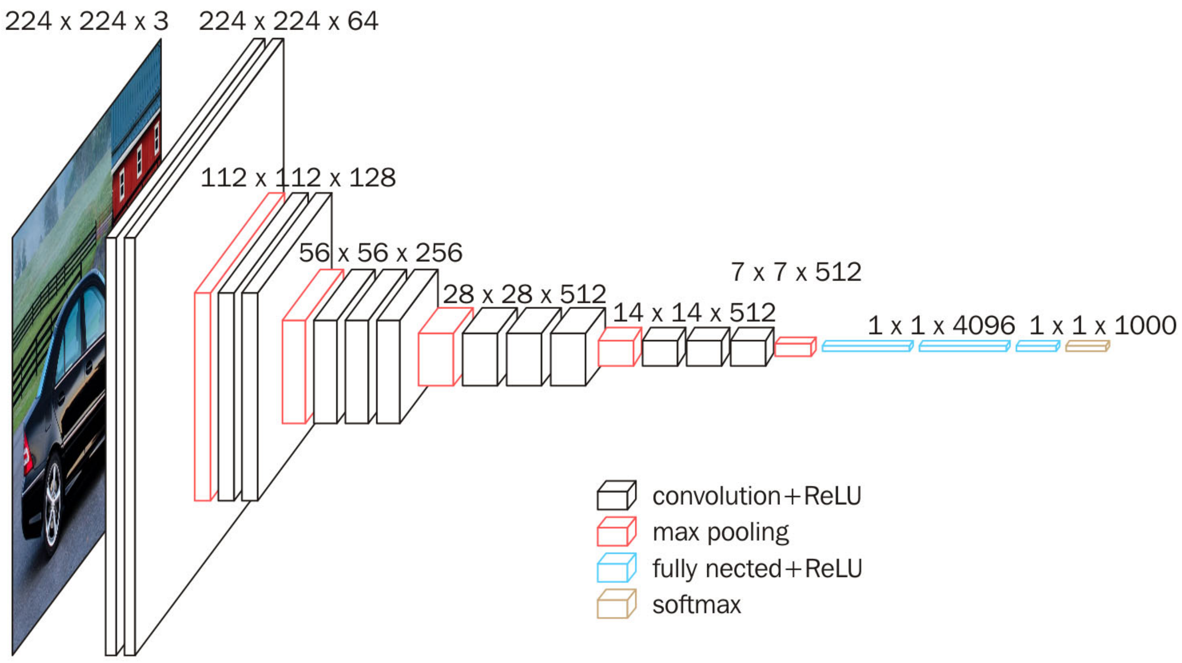


Figure 4.1.1 Network Architecture of modified VGG [6]

4.1.5.2. ResNet50/101

ResNet is a residual neural network [7]. For ResNet50, it means that the number of layers in it is 50. It has improvement on vanishing problem and bursting gradients compared to VGG models. It skips a few layers during the training stage and output directly. The ResNet model scored a score of 96.7 percentage in the picture recognition challenge [7].

We use the ResNet model as the encoder and changed the stride parameter of the 3rd convolutional layers from 2 to 1. The decoder also has two convolution layers.

The neural network graph of ResNet50 is as follows:

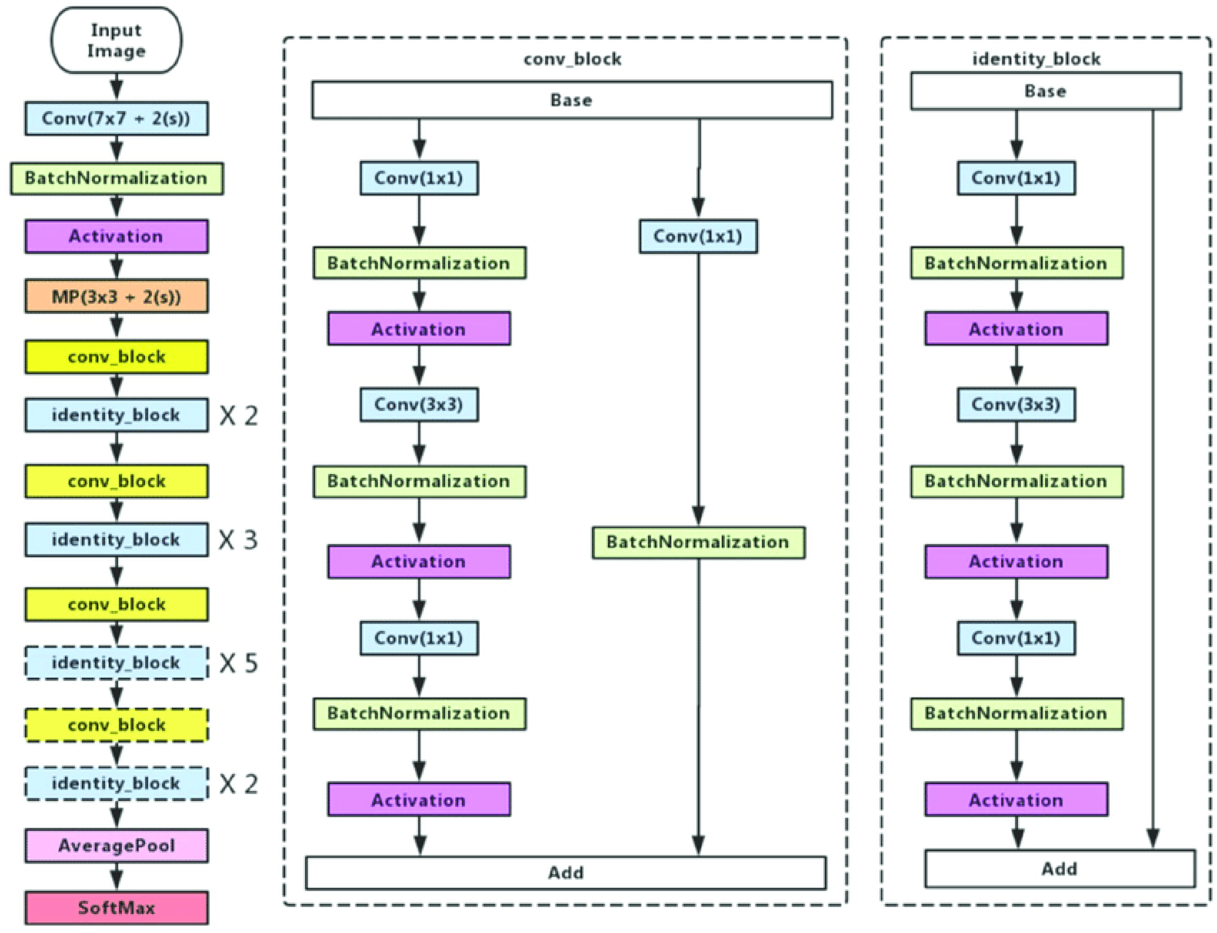


Figure 4.1.2 Network Architecture of modified ResNet50 [8]

4.1.5.3. MCNN

Multi-Column Convolutional Neural Network (MCNN) [9] is one of the deep learning models that perform very well in crowd counting. It is so convenient that the MCNN model has no limitation on the size of the image since utilize multiple CNN with distinct receptive fields of varied sizes to compute the estimated density. Each CNN in MCNN captures various attributes from the input image and the model stack all features in the output

Different from other model, MCNN do not have encoder-decoder designs, instead and it merge the output of each CNN. The following is a diagram of a neural network:

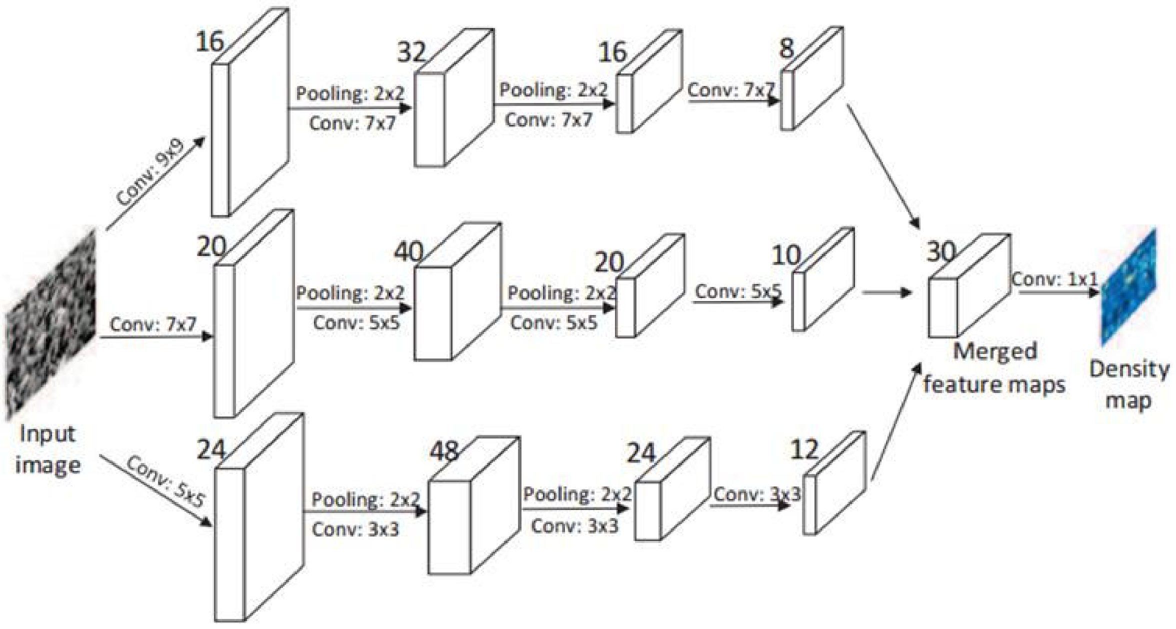


Figure 4.1.3 Network Architecture of modified MCNN [9]

AlexNet

AlexNet [10] is a popular historical CNN model. In the ImageNet Lare Scale Visual Recognition Challenge 2012, AlexNet performed well with a top-5 error of 15.3 %, which is 10% more than the second-place finisher.

AlexNet is the encoder in this experiment. To ensure that the feature maps can be separated, the padding parameters of the convolutional layers are modified. The encoder uses only the first five convolutional layers and down-sampling layer. The decoder also has two convolution layers VGG.

The neural network graph is showed below:

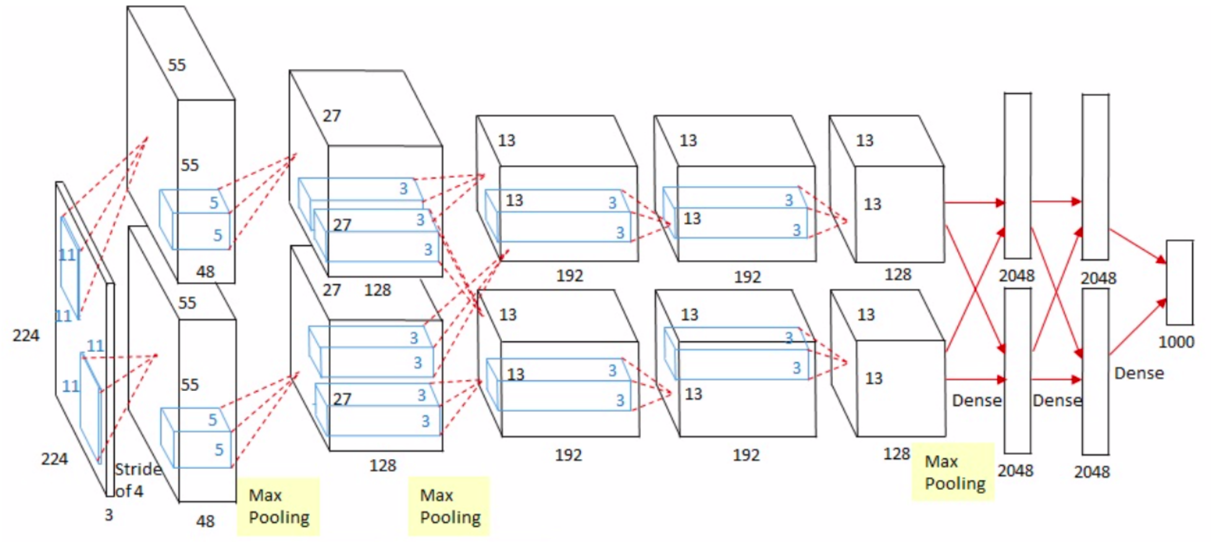
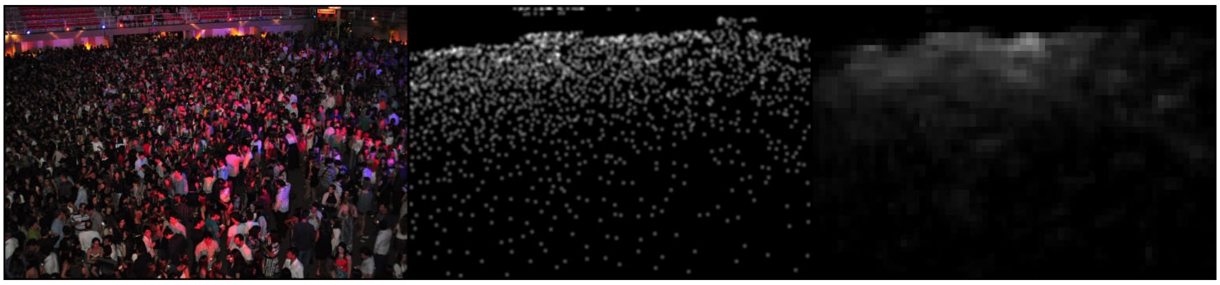


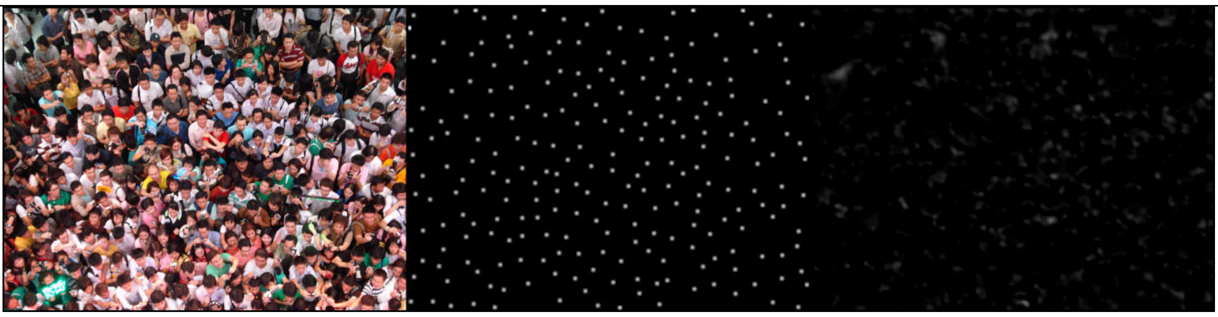
Figure 4.1.4 Network Architecture of modified AlexNet [10]

4.1.6. Result

There are two sections for result. the first section is for visualization, and the second section is for the model assessment results.

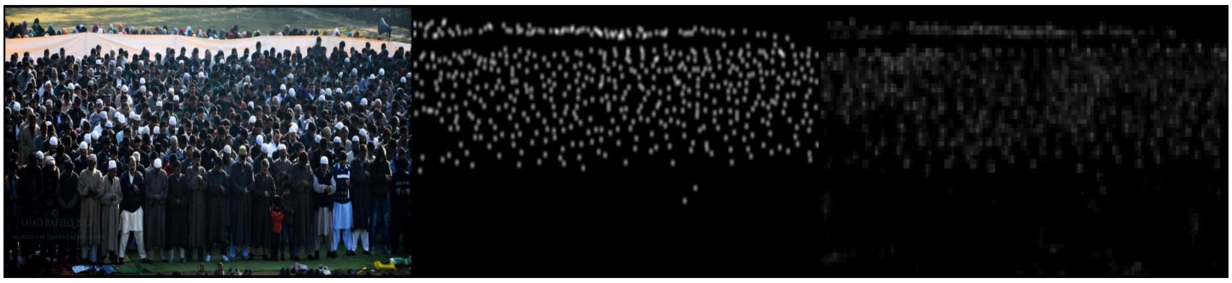
4.1.6.1. QNRF Dataset Visualization

AlexNet  Images Ground-Truth Prediction
MCNN



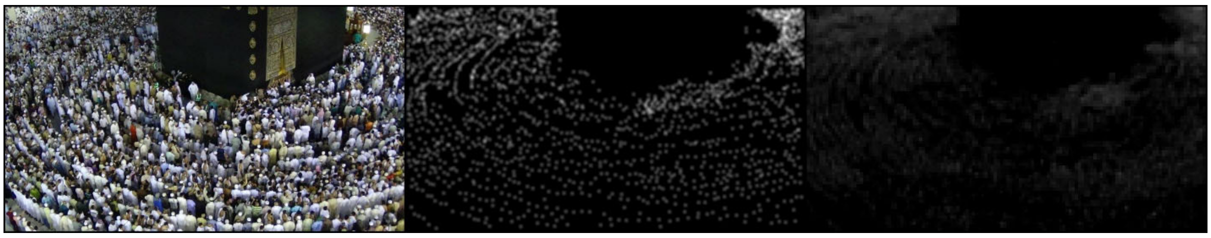
Images Ground-Truth Prediction

ResNet50



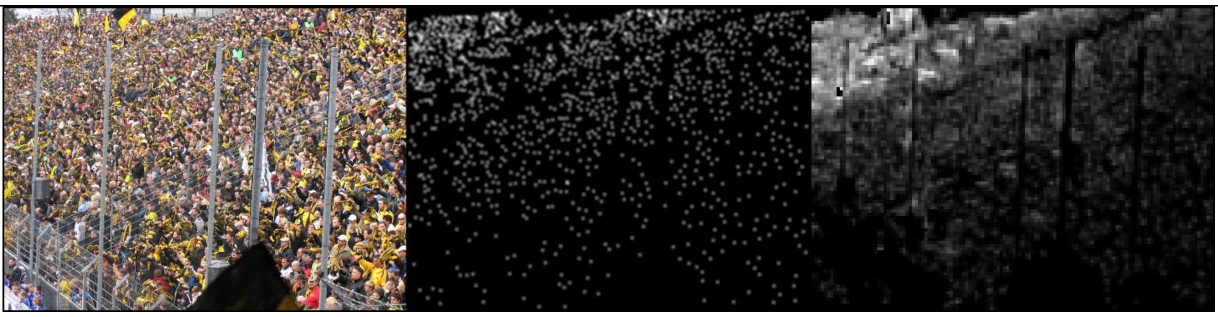
Images Ground-Truth Prediction

ResNet101



Images Ground-Truth Prediction

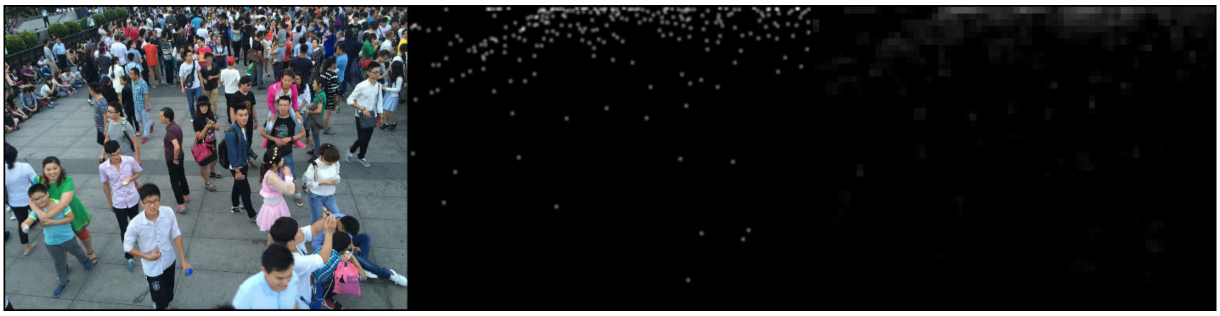
VGG



Images Ground-Truth Prediction

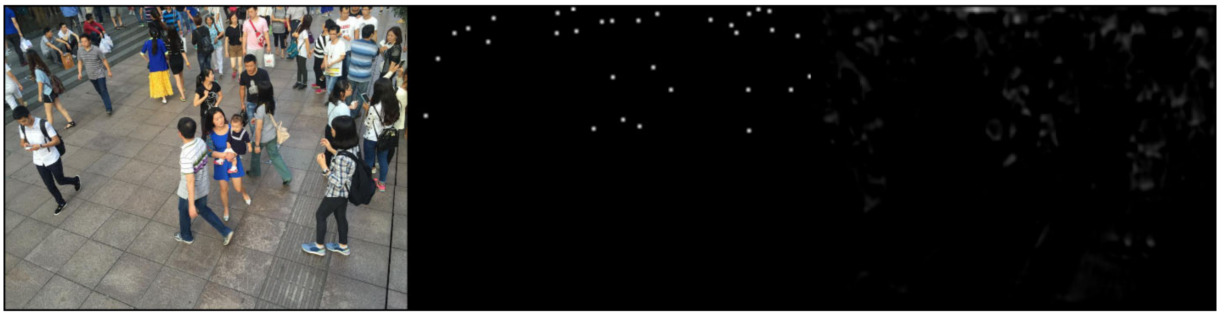
4.1.6.2. SHTB Dataset Visualization

AlexNet



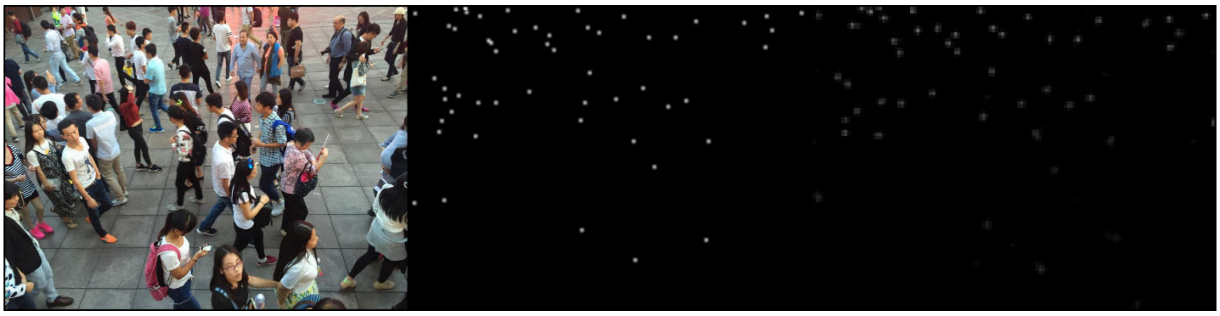
Images Ground-Truth Prediction

MCNN



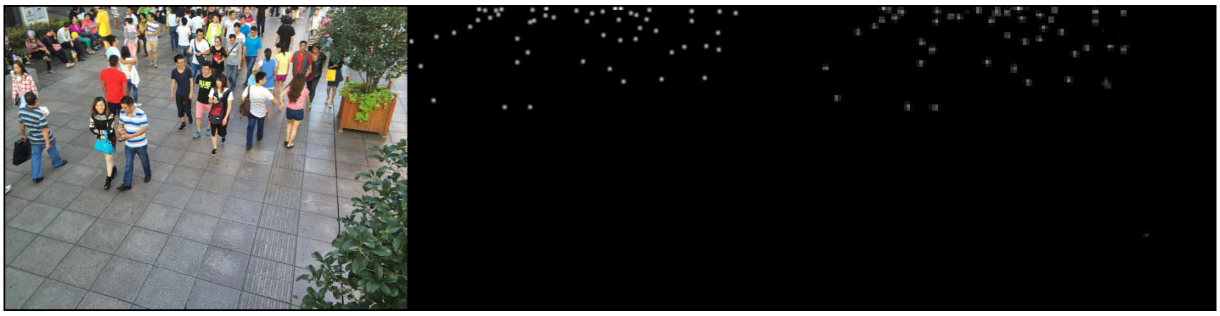
Images Ground-Truth Prediction

ResNet50



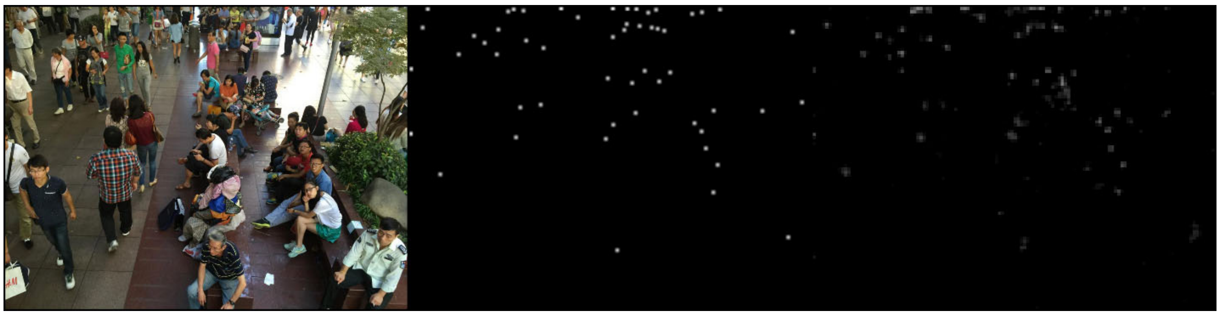
Images Ground-Truth Prediction

ResNet101



Images Ground-Truth Prediction

VGG



Images Ground-Truth Prediction

4.1.6.3. Model Evaluation Results

The focus of the research is to determine which model performs the best in terms of MAE and MSE metrics when it comes to crowd counting. Following is the SHTB result:

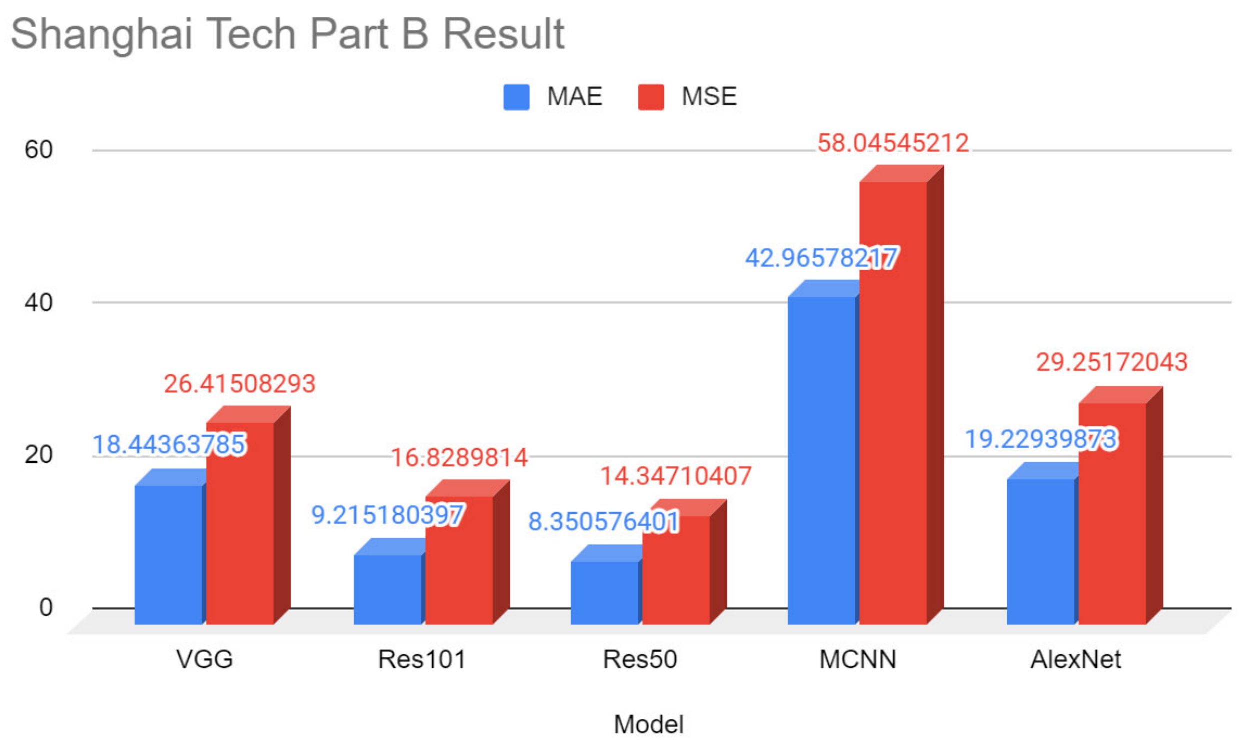


Figure 4.1.5 Result of Classification model for the Shanghai Tech Part B Dataset

The above chart shows that the ResNet50 model perform the best among other models, with an MAE of 8.35 and an MSE of 14.347. The ResNet101 model is the first runner up which has an MAE of 42.96 and an MSE of 58.04. It is clear that the ResNet model series performs better than other series. It is possible that with more epochs, the ResNet101 model can perform even better,

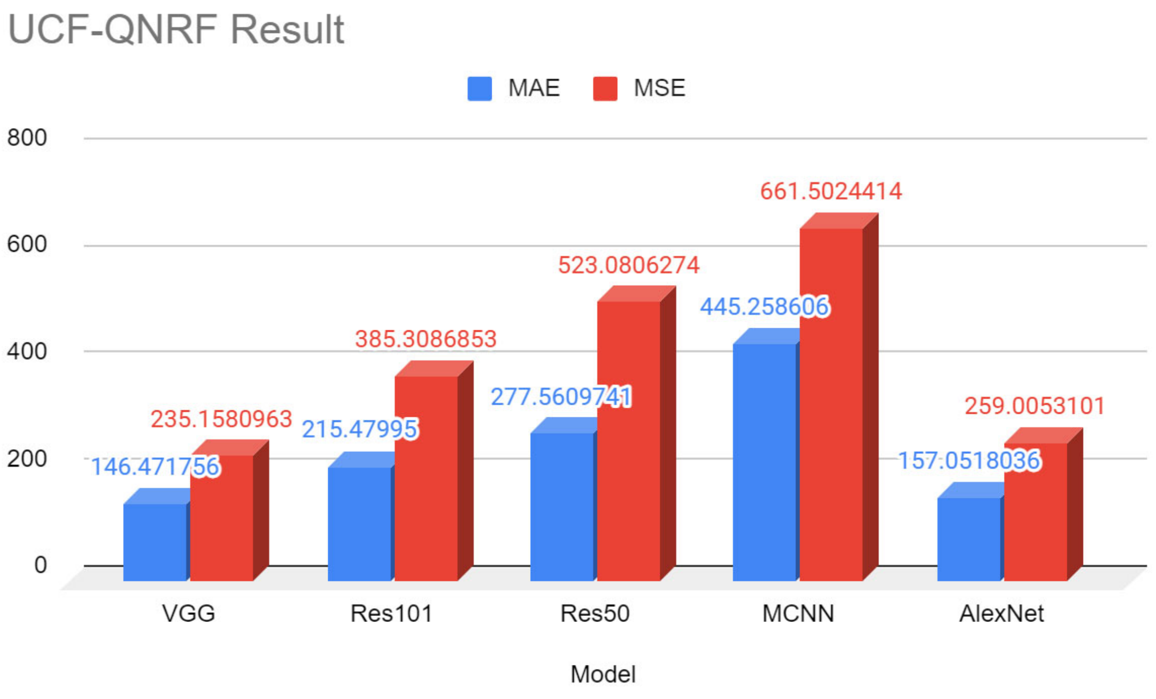


Figure 4.1.6 Result of Classification model for the UCF-QNRF Dataset

For the QNRF result, the VGG model outperform other models with a MAE of 146 and a MSE of 235. The ResNet101 and ResNet50 model come after. While MCNN model perform the worst with a MAE of 445 and a MSE of 661. It is suggested that the large number of humans in QNRF datasets is the main reason resulting significant MAE and MSE on all models. A longer epoch may be beneficial for extracting more characteristics from each image and hence, it may result in better performance.

4.1.7. Summary

In this experiment, density map generation is used to modify the encoder-decoder architecture. Increasing the magnification factor may result in better model accuracy. ResNet family performs better in closer view image for SHTB dataset and VGG performs better in distant view image for QNRF dataset in terms of assessment results. As a result, this study confirms that Resnet50 perform well even if it is closer to the crowd and that VGG is better when it is further away. At this stage, the computational time and the latency are not taken into consideration. The optimization of the model should be done at a later stage.

4.2. Deepstream queuing output accuracy

The purpose of this experiment is to find out the impact of height and distance to the inference result accuracy. It is believed that the result can provide insight on the allowed maximum height of a video source for queue counting (people exiting the queue).

4.2.1. Methodology

The video in the dataset is shot using drones from two community testing centers in Hong Kong with different height settings, i.e. 25m, 30m, 40m and 50m. The resolution is 3840 x 2160.

The model used in this experiment is the PeopleNet model, which is built on one NVIDIA DetectNet v2 detector and ResNet34 as a feature extractor.

For each height variation, we manually count the number of people exit the queue and compare with the inference result to see if there is any different.

4.2.2. Result

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 the drone moved
25	30	39	drone_2_2_ds.mp4	5	6	Overlapping people, blocking tracker
20	26	33	drone_2_3_ds.mp4	4	4	
40	25	47	drone_3_1_cropped.mp4	N/A		Failed to detect/ track to inadequate resolution/overhead angel
50	25	56	drone_3_2_cropped.mp4	N/A		

50	85	99	drone_3_3_cropped.mp4	N/A	Although the camera is still 4K, the image quality is significantly worse

Table 4.2.1 Deepstream queuing output accuracy

The above table shows that the PeopleNet model is able to detect people exiting queue up to 50m height. The key is instead the image quality, even if the pixel count of a human is the same.

Since the video is captured using a drone instead of a static camera, the line drawn in the video may shift following the movement of the drone, which may affect the accuracy. This is a potential area for improvement. In addition, overlapping issue is also very critical, hence, it is ideal to set the camera angle to 45 degrees to the line of the queue exit to minimize this issue.

4.3. Analysis of human size inside a frame

The throughput is determined by the performance of the detection model and the tracker. It is essential that the model can suggest a bounding box in different frame since the tracker cannot track without a bounding box, hence, the model performance is the bottleneck. It would be convenient if we know the number of pixel required to represent a human for the detection model.

The detection model in this experiment is the PeopleNet model, which is built on one NVIDIA DetectNet v2 detector and ResNet34 as a feature extractor. The ResNet model is the best feature extractor, according to our research, and 34 layers are chosen since it has a reasonable inference time.

4.3.1. Methodology

A testing framework for executing image inferences is constructed to determine the entire pixel count. The PeopleNet model is deployed on a machine with an NVIDIA RTX 3090 graphics card, an Intel i7-8700 processor, and 16 GB of RAM.

The downloaded PeopleNet model is pretrained and unpruned. TAO-toolkit was used to optimize the model and make it compatible with the TensorRT engine for the model development pipeline. The PeopleNet model takes in a 960x544x3 dimension input tensors and generates a 60x34x12 bounding box coordinate tensor and a 60x34x3 class confidence tensor. After inferencing, we use the NMS clustering algorithm to give suggestion on the location of the bounding box. Under hyperparameter tweaking, the threshold confidence is set at 0.6, which is the most optimum value.

The screenshot used in this experiment was taken from a drone video of a community testing center in Hong Kong. 3840 x 2160 x 3 is the default frame size. The frame is first downscaled to 960 x 544 x 3, and then 20 images were generated by further scale down the it by 5%. If the frame is smaller than the kernel size, it will be padded with white tile. This experiment aims to determine what pixel count is required to portray humans. Below are the inferenced images.

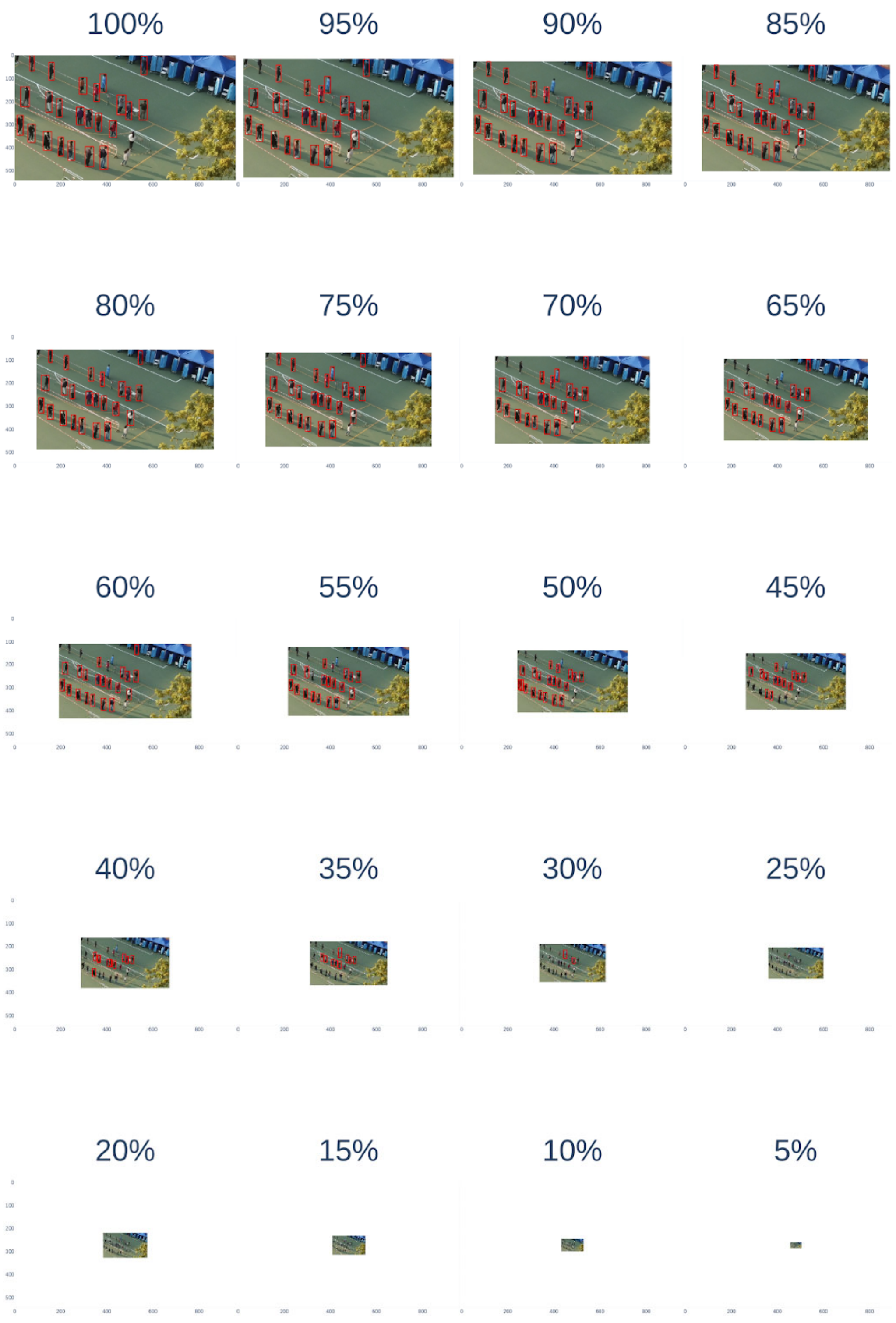


Figure 4.3.1 Gallery of Inferenced Images

4.3.2. Result

This experiment aims to find out the minimum pixel count that is enough to represent humans, and relationship between scaling factor, number of bounding boxes and total pixel count of minimum bounding box inside a frame, where scaling factor refers to the percentage of scaling down of the original image and scaling factor 1 is original image.

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 4.3.1 Result of the Analysis of human size inside a scaled down Image

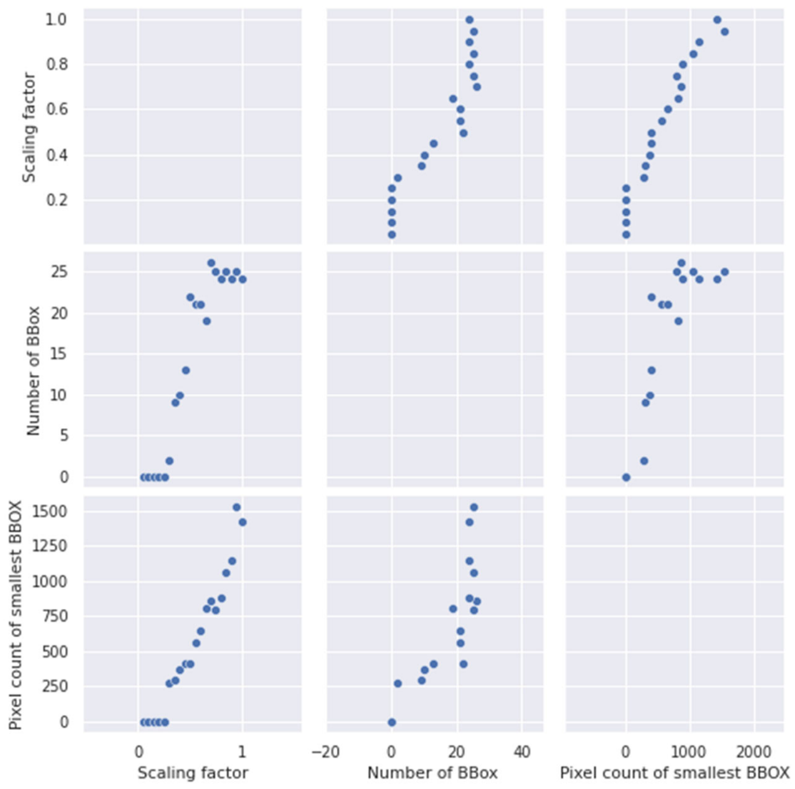


Figure 4.3.2 Pairplot of the Result of the Analysis of human size inside a scaled down Image

In Table 4.3.1 and Figure 4.3.2, there is a positive relationship between the scaling factor is directly proportional to the number of bounding boxes, as well as between the scaling factor and the number of pixel in the smallest bounding box. The smallest bounding box size is formed by 11x25 in rectangular shape which has a total size of 275 pixel, and the largest bounding box size is 20x71 in rectangle shape with total size of 1420 pixel. Besides, the smallest bounding box is in the 0.3 scaled image and more human can be detected in higher scale image. This demonstrates that the PeopleNet model can find more humans if their pixel count is higher.

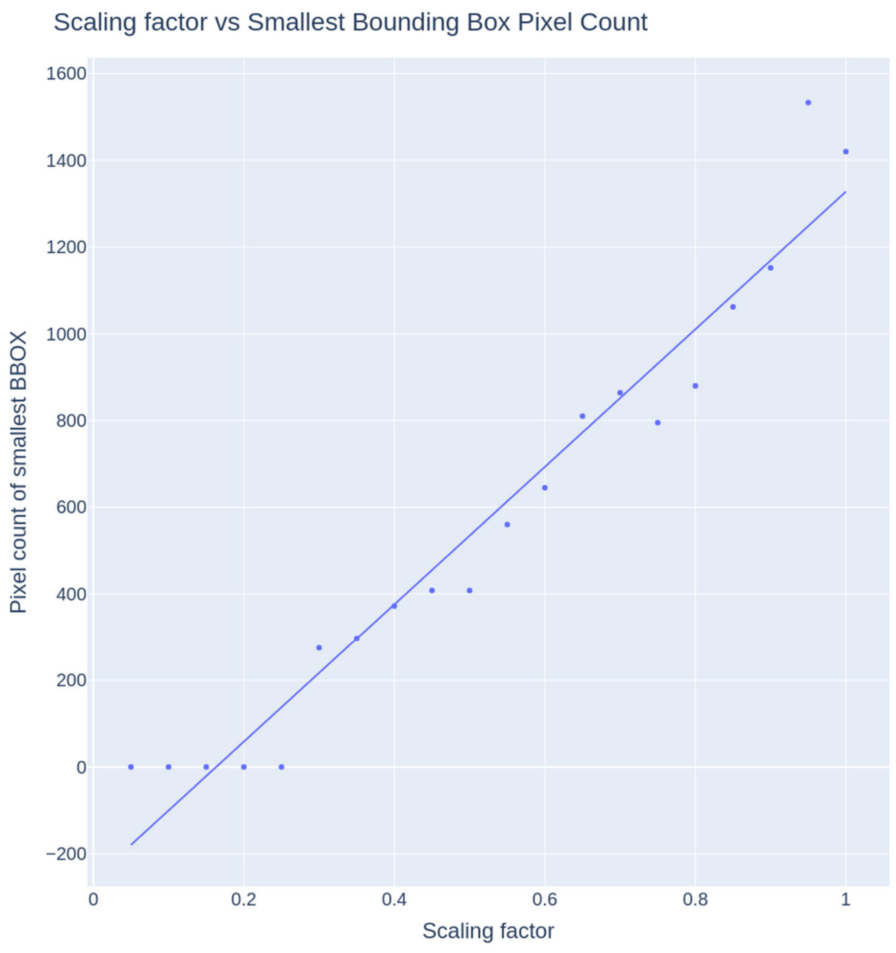


Figure 4.3.3 Scatter plot between scaling factor and the pixel count of smallest bounding box

In Figure 4.3.3, the scaling factor is directly proportional to the total pixel count of the smallest bound box. This result is foreseeable since the bounding box with high confidence level has a higher priority and the bounding box with the low confidence level is usually filtered out. The bounding box's confidence level decreases as the human becomes smaller. When the scaling factor is under 0.25, the suggested bounding box has inadequate confidence level which leads to that the model are not confident enough to detect the humans.

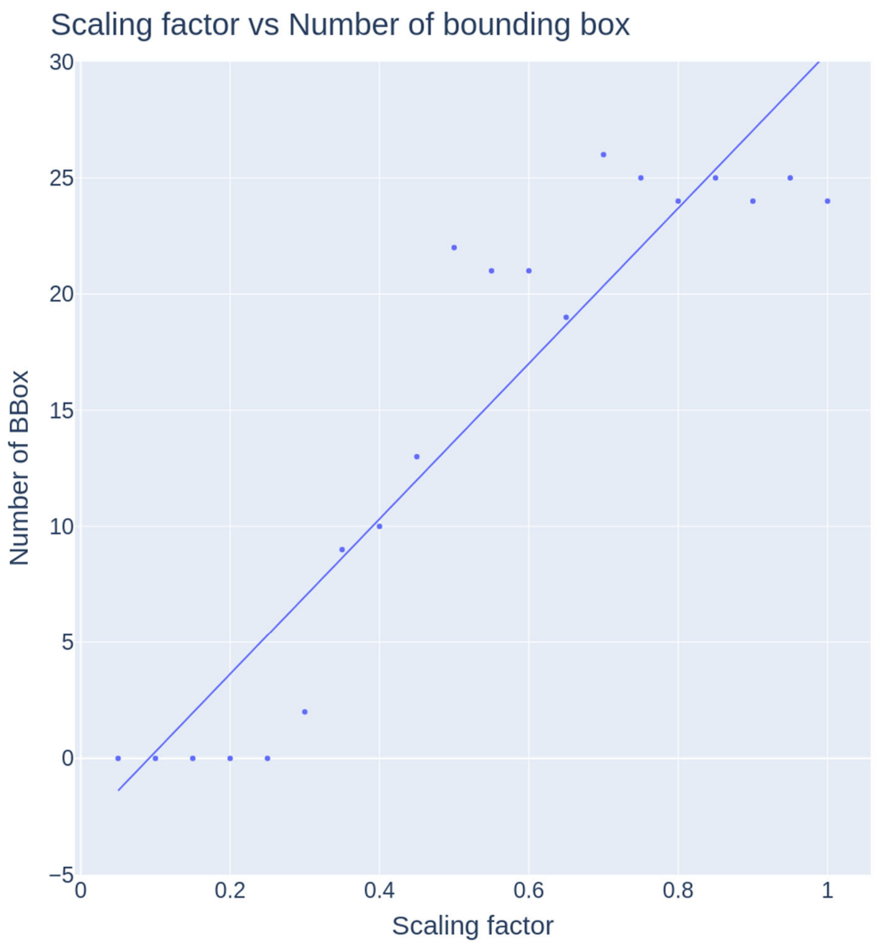


Figure 4.3.4 Scatter plot between scaling factor and the number of bounding boxes

Tn Figure 4.3.4, the scaling factor is directly proportional to the number of bounding boxes. When the image is bigger, the detection model can suggest additional bounding boxes with high confidence. As a result, most of the bounding boxes are kept and are not filtered by the clustering algorithm. However, the bounding box in small images has level confidence level which do not fulfill the requirement of the clustering algorithm, hence is filtered out. The result shows that we need to pay more attention on the threshold of confidence level in clustering algorithms as it may affect the minimum pixel requirement.

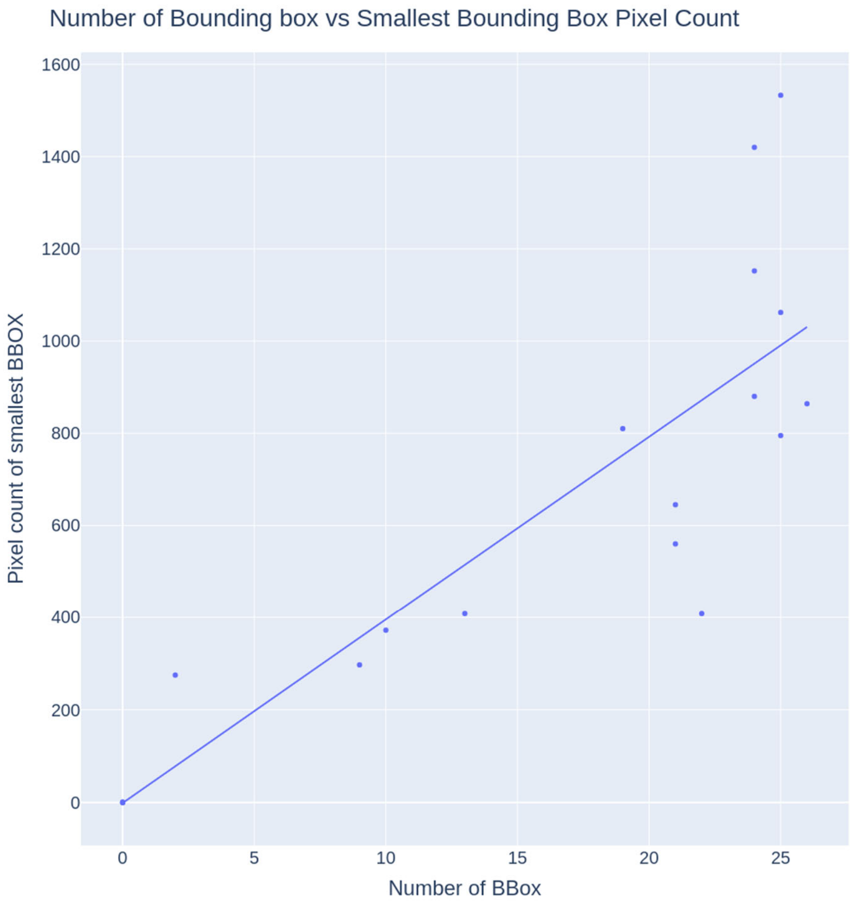


Figure 4.3.5 Scatter plot between number of bounding boxes and the pixel count of smallest bounding box

In Figure 4.3.5, The number of bounding boxes is directly proportional to the total pixel count of the smallest bounding box, but the relationship is not strong. Because these two metrics have a positive association with the scaling factor, there is an indirect mapping between them, and therefore they are linked.

4.3.3. Summary

This experiment investigated the relationship between image size and pixel count of the human bounding box. The result shows that the minimum required pixel count for the PeopleNet model to recognize humans is at least 270 while the NMS threshold is 0.6.

The cropping approach is able to produce the highest number of human bounding box. The result show that the bottleneck is on the bounding box clustering algorithm, but not the detection model. It is believed that a better clustering algorithm may be able to detect human with less pixel count.

4.4. System Design

The system design diagram shown in Figure 4.4.1, comprise of three parts including the MLOps part, Backend and Frontend.

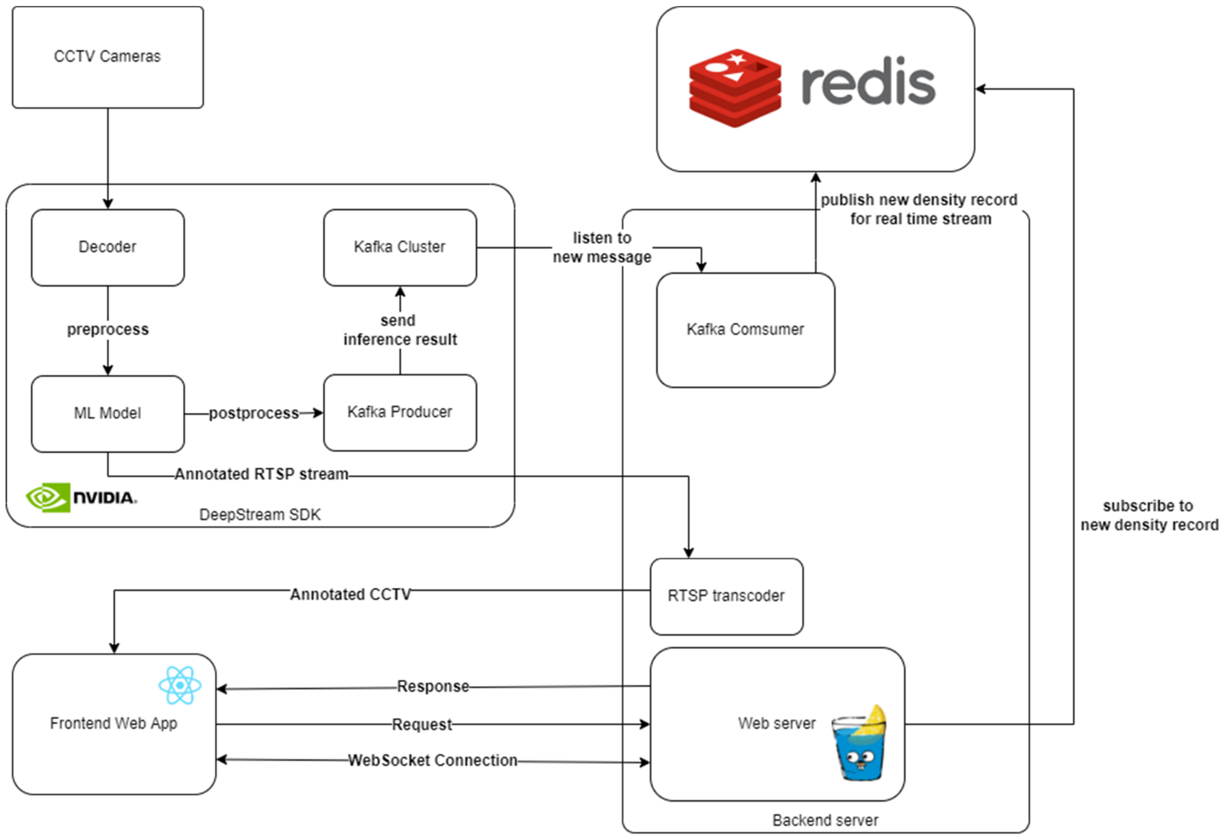


Figure 4.4.1 System Design Diagram

4.4.1. MLOps

We will discuss how to utilize the DeepStream SDK in this section. The DeepStream SDK use the open-source multimedia library Gstreamer and its decoder for decoding the incoming stream with hardware acceleration internally.

We are simulating an RTSP stream incoming from the server with a static file stored locally on the inference machine. For high-performance decoding, the stream will be decoded by hardware. Following decoding, the frames will be dynamically grouped together based on load and transmitted for inference. RestNet 10 is the bounding box regression model that we are currently employing. Because it is utilized on computationally restricted hardware, it was chosen primarily for its low latency. The result is a collection of bounding box suggestions, which are then clustered and tracked by Gst-infer and Gst-nvtracker. Non-maximum Suppression (NMS) is the clustering technique employed, and it matches all relevant ideas to a single result. The Kanade-Lucas-Tomasi (KLT) tracker was utilized, which is capable of identifying distinct objects over several frames inferenced. When the scene is consistent across frames, as is most usually the case with airport CCTV input, this tracker works best.

The ML model's inference result will be post-processed to provide understandable data, which will then be transmitted to the Kafka Producer. Kafka is a high-performance and low-latency distributed event streaming platform. Each Kafka broker may handle numerous topics, and a Kafka cluster can include more than one Kafka broker. Topics keep track of messages. Clients or the backend server can access the result by subscribing to the relevant topic, which Kafka Producer sends as a message to the Kafka cluster. A captioned version of the CCTV broadcast will be accessible through RTSP at the same time.

4.4.2. Backend

For the backend, there are Kafka consumer, gin web server, RTSP transcoder, and Redis.

Kafka Consumer

The Kafka consumer is for actively listening to the inference result from Kafka cluster. Each new inference result received will be published to Redis.

Web Server

Gin, a Golang-based HTTP web server framework, is used for developing the web server. It takes REST API requests from the frontend, searches the database for the relevant data, and returns the result as JSON to the frontend. It also manages the WebSocket connection for streaming real-time data.

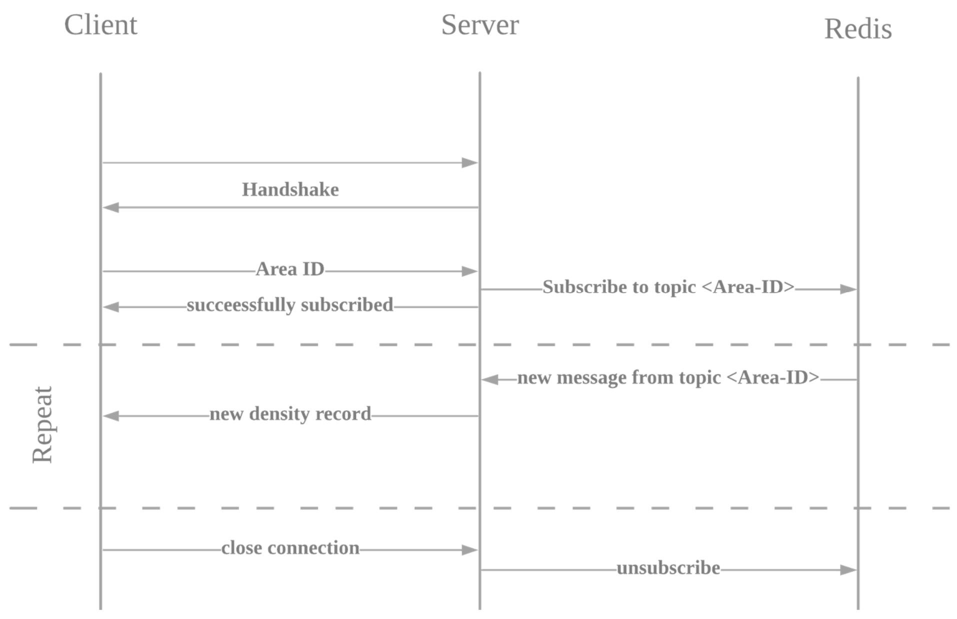


Figure 4.4.2 WebSocket Sequence Diagram

The client and server must first handshake before establishing a connection. Once the connection has been established, the client tells the server the area id of the area it wishes to observe, and the server subscribes to the topic in Redis based on the client's behavior. The client will receive a successful message if the connection is successful. During an open connection, the server will receive fresh inference results as they become available and send them on to the client, who will then update the UI accordingly.

RTSP transcoder

At the modern browser do not support video streaming through RTSP, a RTSP transcoder is used to first transcode the video to MPEG (Moving Picture Experts Group) that can be stream on modern browser, then stream through WebSocket.

4.4.2.1. Frontend

Data visualization is also a key aspect of this project, which requires an understandable and accessible outcome in order for all of the data collecting and inference to be useful. Because native application hardware capabilities are not required, we choose to use a website for user access to the record. As a consequence, the user will find it simple to adopt the system, and the design and functionality may be readily tweaked without the hassle of having to update an application. Practically, we are using React Typescript as the frontend framework.

For the live data ,we would utilize the WebSocket to subscribe to a topic that feeds in live data. This data allows us to see real-time changes in a certain area, such as sudden congestion, or if a specific area is blocked, or if a queue is starting to become long. In the end, it's meant to provide operational management and situation awareness.

Real-time heatmap

The heat map can show and reflect the real time people density data. The map itself is stored as GeoJSON, a format for encoding a variety of geographic data structures. Users can create the GeoJSON file in advance and store it in a server or the cloud, then import it to the Web Application. For the ease of demonstration, the map of Hong Kong (Figure 4.4.3) is selected.

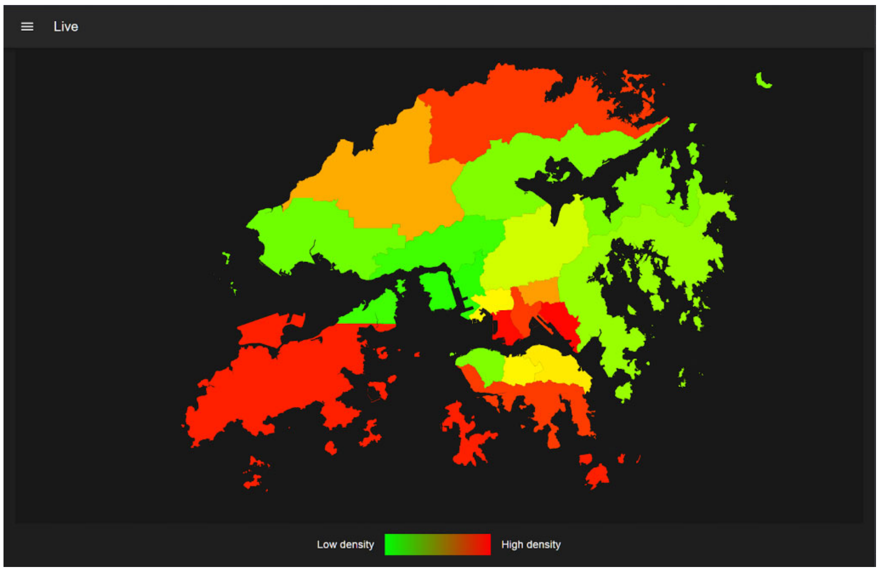


Figure 4.4.3 Density heatmap

The people density data of each region is fed by the backend through WebSocket to reduce the latency.

The heatmap shows the users a brief overview of the occupancy of each region. Green means low density in the region while red means high density in the region. It is believed that the user can infer if a region is becoming overcrowded or blocked from the heatmap.

The heatmap also supports hover action (Figure 4.4.4) which gives the user the number of people in the region.

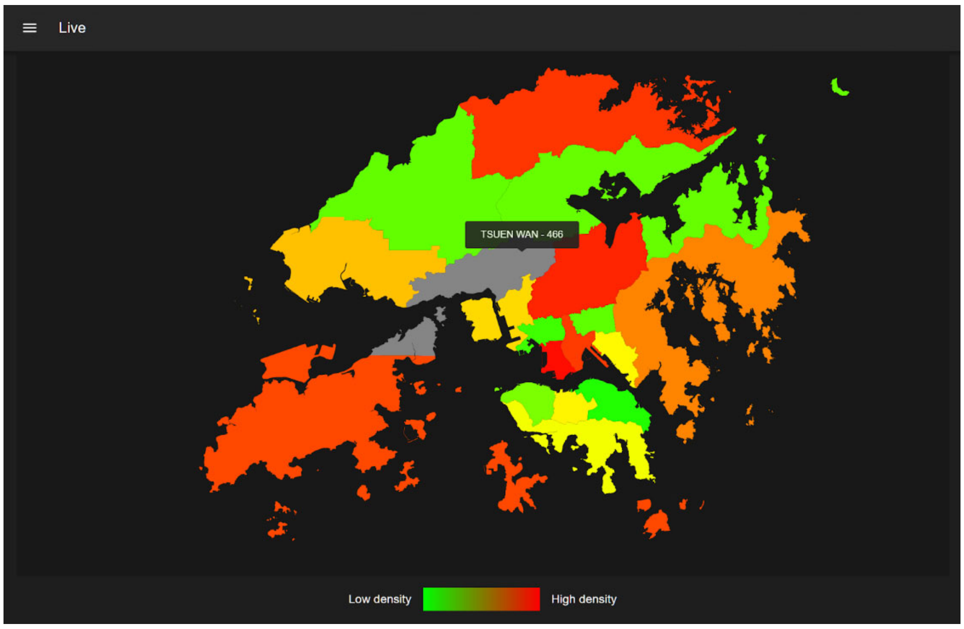


Figure 4.4.4 Density heatmap hover state

Hence, the user can focus on a specific region without being overwhelmed by other unrelated statistics. Ultimately it is designed to give operational management and situation awareness.

RTSP stream and Entry exit data

In order to provide an easy way for the user to view the stream output by the Deepstream, the stream is also embedded in the Web Application. The stream will be as Figure 4.4.5.

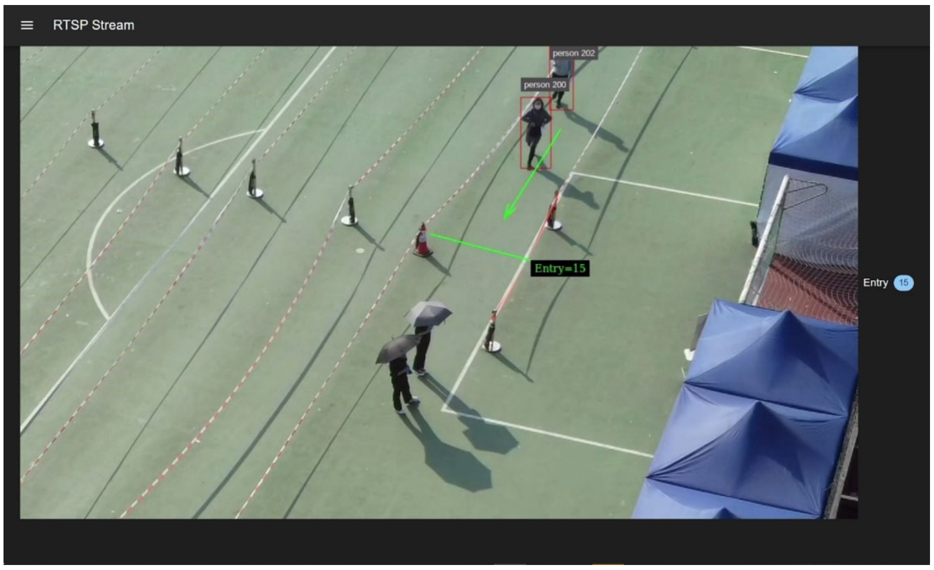


Figure 4.4.5 RTSP Stream

This page tells the users the number of entries in this specific queue which is also fed by WebSocket. Users can also check if the model is functioning or not through cross checking the actual number of entries and the detected number of entries.

As modern browsers do not support RTSP streams natively, the current approach is to re-stream using ffmpeg, an open-source media processing library. The video is transmitted through WebSocket and displayed on canvas.

5. Challenge

In this chapter, we will discuss the challenge we have faced and the solution.

5.1. Data Collection

Because this project necessitates footage from a significant height, yet AAHK does not allow to provide CCTV footage due to privacy concerns, and we do not have the equipment to hoist a camera to such heights, we decided to utilize a drone to film from a great height. We chose two of the community testing centers after analyzing the ideal location, which is outside and has a continual line of people.

5.2. Tracking using density map

In comparison to the bounding box approach, the density map approach is able to locate more humans in a crowded image. However, tracking humans using a density map is difficult, and there is no other option than to use optical flow, which is less precise than the bounding box technique. Furthermore, the density map's inference output is excessively sparse. The method we use is to first zoom in to a level where the bounding box technique can get reasonable inference results, and then use tracking with bounding box. This method provides more information than density map because the passenger flow may be calculated as well.

6. Conclusion

To conclude, this project compared different state-of-art deep learning models and Machine learning algorithms in terms of effectiveness in crowd counting. According to the findings, increasing the magnification factor may enhance model accuracy. The human inside the image must be greater than a certain size and have adequate resolution to be recognized by the model using the bounding box approach. In the best-case situation, PeopleNet can recognize persons with a pixel count of at least 270 and the NMS threshold set to 0.6. After comparing the bounding box and density map approaches, it appears that the bounding box technique may give more information since it allows for better tracking, however the density map approach can recognize more humans in a busy scene.

This project also includes a functioning system design and prototype for a deep learning-based application, which includes Web server, Redis, Kafka, and DeepStream, among other components. In addition, the user may monitor real-time changes in a heatmap, hover for more information, and inspect the model's inference output in the web application.

7. Future Works

In this chapter, we will discuss the future work that can be extended from this project, including prediction on people density through Long short-term memory (LSTM) model, analysis of framerate, image quality and angle impact on tracking, and queue identification approaches.

7.1. Prediction

With the people density data in hand, it is logical to ask if we can anticipate future people density data based on historical patterns. The prediction is anticipated to help with labor and resource allocation, as well as long-term planning. When it comes to time-series data prediction, the LSTM model is one of the best models to consider.

7.1.1. Long short-term memory (LSTM)

LSTM is a kind of Recurrent Neural Network (RNN) in which the network contains loops that allow information to survive. Traditional neural networks, such as CNN, are unable to comprehend temporal data adequately, making them unsuitable for prediction.

LSTMs feature a chain-like structure like regular RNNs, but the repeating module is different. There are four neural network layers instead of one, each of which interacts differently.

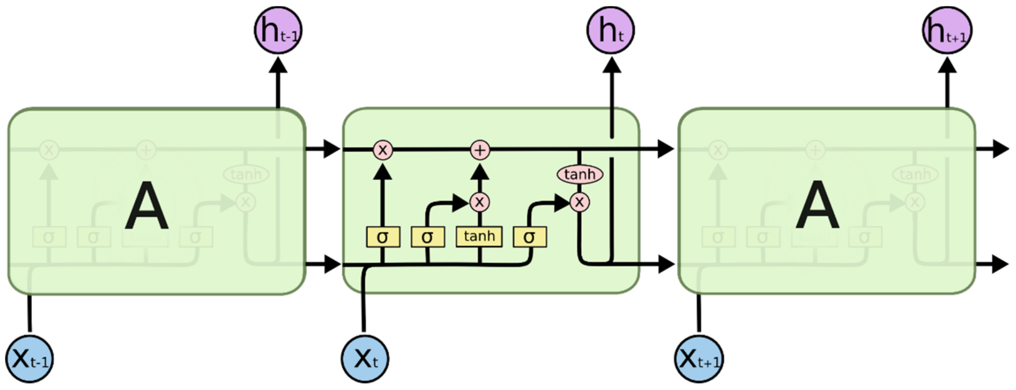


Figure 7.1.1 Repeating Module in an LSTM

In the diagram above, each line transfers a full vector from one node's output to the inputs of other nodes. Pointwise operations, such as vector addition, are represented by pink circles, whereas learned neural network layers are represented by yellow boxes. Concatenation happens when lines combine, whereas forking occurs when the content of a line is duplicated and delivered to many locations.

The key to LSTMs is the cell state, which is represented by the horizontal line running through the top of the picture. The cell is in a similar state to that of a conveyor belt. It flows down the entire chain with only a few minor linear exchanges. It's extremely simple for data to pass through it unchanged. The LSTM has the ability to erase or add information to the cell state, which is tightly regulated via gates. Gates are a device for allowing information to pass through selectively. They're made up of a sigmoid neural net layer and a pointwise multiplication process.

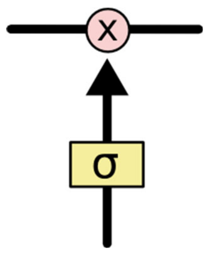


Figure 7.1.2 Structure of the sigmoid neural net layer

The sigmoid layer produces integers ranging from zero to one, indicating how much of each component should be allowed to pass. "Allow nothing through!" signifies a value of zero, whereas "let everything through!" means a value of one.

Three of these gates are present in an LSTM to safeguard and govern the cell state. LSTM can categorize, analyze, and generate predictions based on time series data by controlling the cell state.

7.2. Analysis of framerate/ image quality/ angle impact on tracking

Aside from the human's pixel count, framerate, picture quality, and capture angle can all have an impact on the model's accuracy and tracking.

Experiments with various framerates can be undertaken as part of a larger study to determine the lowest framerate that will result in acceptable tracking performance. Frame interpolation can also be used to compare the framerate to the native framerate to evaluate if frame interpolation improves the accuracy.

Furthermore, preprocessing the video with spatial (e.g. blurring, blocking, ringing, etc.) and temporal (e.g. flickering, mosquito noise, floating, etc.) effects may be used to assess how sensitive the tracker and model are to video quality. Its goal is to imitate a poor video source or a video source with a lot of compression. The experiment might also use AI upscale for lower-resolution video sources to investigate if the inference result from the upscaled video outperforms the original lower-resolution video.

Finally, the extreme scenario of a direct above angle poses a significant challenge for the current approach. Exploring or training a new model would be an intriguing research project that might help with this issue.

7.3. Queue Identification

The approach for counting people in a queue is achieved by cropping the queue from the original video source which is relatively static. It would be more useful if the model could dynamically detect the presence of a queue, allowing it to be applied to queues without a fixed boundary and situations where several queues exist in a single video source.

8. Bibliography

[1] R. Ashmore, R. Calinescu and C. Paterson, "Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges," p. 4, 2019.

[2] "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].

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

[4] J. L. Gao, W. Zhao, D. Bin Wang, C. Gao and J. Wen, "C^3 Framework: An Open-source PyTorch Code for Crowd Counting," *arXiv preprint arXiv:1907.02724,* 2019.

[5] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv 1409.1556,* 2014.

[6] D. HAN, "VGG16 学 习 笔 记 ," 2018. [Online]. Available: http://deanhan.com/2018/07/26/vgg16/. [Accessed 22 January 2022].

[7] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv preprint arXiv:1512.03385,* 2015.

[8] 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.

[9] 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.

[10] 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.

[11] "originlab," 2018. [Online]. Available: https://www.originlab.com/www/products/GraphGallery.aspx?GID=303. [Accessed 29 September 2021].

[12] N. Yeung, "A cool idea: Hong Kong airport cuts energy consumption with new air-conditioning control system," 9 August 2021. [Online]. Available: https://www.scmp.com/news/hong-kong/health-environment/article/3144268/cool-idea-hong-kong-airport-cuts-energy?module=perpetual_scroll&pgtype=article&campaign=3144268. [Accessed 29 September 2021].

[13] "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].

[14] "Hong Kong International Airport Overview," Hong Kong International Airport, [Online]. Available: https://www.hongkongairport.com/en/the-airport/hkia-at-a-glance/fact-figures.page. [Accessed 29 September 2021].

[15] "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].