Received: 7 September 2022 | Revised: 27 September 2022 | Accepted: 11 October 2022 | Published online: 11 October 2022

RESEARCH ARTICLE

Artificial Intelligence and Applications
2023, Vol. 1(4) 221–229
DOI: 10.47852/bonviewAIA2202391

BON VIEW PUBLISHING

Real-Time Human Detection and Counting System Using Deep Learning Computer Vision Techniques

Hamam Mokayed[1,*], Tee Zhen Quan[2], Lama Alkhaled[1] and V. Sivakumar[2]

[1]*Department of Computer Science, Electrical and Space Engineering, Lulea University of Technology, Sweden*

[2]*Faculty of Computing, Asia Pacific University, Malaysia*

Abstract: Targeting the current Covid 19 pandemic situation, this paper identifies the need of crowd management. Thus, it proposes an effective and efficient real-time human detection and counting solution specifically for shopping malls by producing a system with graphical user interface and management functionalities. Besides, it comprehensively reviews and compares the existing techniques and similar systems to select the ideal solution for this scenario. Specifically, advanced deep learning computer vision techniques are decided by using YOLOv3 for detecting and classifying the human objects with DeepSORT tracking algorithm to track each detected human object and perform counting using intrusion line judgment. Additionally, it converts the pretrained YOLOv3 into TensorFlow format for better and faster real-time computation using graphical processing unit instead of using central processing unit as the traditional target machine. The experimental results have proven this implementation combination to be 91.07% accurate and real-time capable with testing videos from the internet to simulate the shopping mall entrance scenario.

Keywords: detection, tracking, counting, deep learning, computer vision, Covid 19

1. Introduction

The booming event industry has faced its biggest disruption due to the challenges of Covid 19, causing many events to be cancelled, postponed, relocated, and transformed into virtual events (Congrex Team, 2020). However, the demand of crowd management does not reduce as it is still significant to prevent virus spreads by controlling the crowd density in a specific environment. In this case, object detection is the ideal solution for crowd management in public areas like malls, shops, restaurants, parks, subway station, and more. Unlike the complicated and traditional manual crowd management which requires planning, risk assessment, and communication, object detection is far more efficient and effective without involving any human or manual operations.

In the current situation, a huge crowd could be especially dangerous where the virus is mainly spread in the air when people are close to each other. Furthermore, countries are implementing strict crowd management and social distancing rules in public places to minimize the virus spread. In Kuala Lumpur, Malaysia alone, 150 Covid 19 hotspot locations were identified on 8 May 2021 in shopping malls, hypermarkets, and grocery stores by the government with big data analytic and artificial intelligent tool (The Straits Times, 2020). There are a significant need to control and manage the crowd in these malls with the biggest crowd potential at this time.

In Malaysia, the government has established a series of standard operating procedure (SOP) for enterprises like limiting the store's crowd size, maintaining a safe social distance between individuals, and many more. The government has even implemented an emergency law to fine enterprises which have violated or fail to perform the SOP. Even more, the fine amount has been enormously increased to 50,000 Malaysian Ringgit for companies and corporations effective from 11 March 2021 (Bernama, 2021). This has caused countless fears to enterprises, and manually monitoring its crowd all time is difficult and requires a heavy investment in human resources. Undoubtedly, most of these affected enterprises are shopping malls, restaurants, and stores due to high customer flow. For example, shopping malls will have to assign a dedicated staff at the entrance just to ensure that the mall is not overly crowded, and the current in-house customers do not exceed the limit stated in SOP. Hence, it is almost impossible to keep track of the crowd size accurately and continuously with manual operations.

Moreover, human object detection and counting system is also useful for preventing abnormal behaviors. Whenever there is an event, a crowd regardless of size will be usually present and this could lead to unpleasant situations caused by unpredicted human behaviors. Besides, the crowd can often be cacophonous causing the atmosphere and aura to be unpleasant which soon could be

*Corresponding author: Hamam Mokayed, Department of Computer Science, Electrical and Space Engineering, Lulea University of Technology, Sweden. E-mail: hamam.mokayed@ltu.se

escalated to violence. Therefore, effective crowd management should always be performed to also ensure the safety of a human interaction event which could lead to injuries and other worse circumstances.

This paper proposes a real-time human detection and counting system specifically for public places with entrance like shopping malls. It helps to detect individual human objects within the camera coverage area to estimate the crowd size. Technically, a surveillance camera will be installed at the entrance and send real-time video footage to the system installed in a server. The system should then perform human detections on the video received and keep track of the number of human objects who have entered and left the environment. In addition, this system implements image processing technique and lightweight deep learning algorithms for object detection, classification, and tracking which requires less computational power. The main contribution of the work is to allow human detection and counting in almost real-time for concurrent crowd management in real-life environment. This is achieved by converting the information into TensorFlow format which is also known as graph visualizations for faster processing using the more powerful graphical processing unit (GPU) as target machine.

2. Literature Review

2.1. Other importance of human detection and counting

Human detection and counting with computer vision not only can solve crowd management problems but also help to study people behavior in a specific environment or scenario. In Velastin et al. (2020), the same approach is used to perform passenger traffic management for reducing delays in public transport systems (Mokayed et al., 2021). Especially in railway system where traveling at optimized and highest speed at their own track, passenger transit time is the only critical factor affecting the system's effectiveness and efficiency. For example, detecting and counting how many passengers can enter or exit the train in each transit. The challenge is related not only to human detection but also to other object detection like plate numbers, vehicles, and other objects that add value to the field of the intelligent transportation system (Velastin et al., 2020). Thus, an intelligent model is developed to determine the best dwell time, suitable physical conditions such as door width, gap between the train and platform, allocation of train interior, and train and platform floor material while ensuring passengers can be boarded and exited easily, quickly, and safely.

Another popular application is security event analysis. It promptly and automatically detects any events that are abnormal or suspicious with great possibility to cause disruptions. Mokayed et al. (2021) and Piciarelli and Foresti (2011) have categorized this operation into anomaly detection and explicit event recognition with both approaches implemented. Explicit event recognition requires explicit knowledge of the events to be detected with semantic description. Technically, the system captures video data and compares it with the knowledge base to determine whether it is a recognizable event and classify the event sequence. On the other hand, no need for pre-developed knowledge base for anomaly detection. Usually, unsupervised learning is applied detect almost all activities in the scene with a probability indicating how anomaly is each event. Anomaly detection is more dynamic to adapt new anomaly events yet requiring human evaluation to extract events that are really abnormal.

The last application to be discussed in this section is pedestrian detection to prevent one of the major traffic accidents such as pedestrian collision. Liu and Sun (2012) proposed a fast in-vehicle pedestrian detection system for driver assistance. In this paper, pedestrian detection is still currently a hard problem that requires prompt detection yet complex algorithms as pedestrians could have major differences in appearance and background environment. To advance the parallel computation of the system, the researchers use a C4 algorithm to extract CENTRIST features for pedestrian detection on a NVIDIA GPU. Both Internet protocol (IP) and infrared cameras are used to capture quality input video data for greater detection accuracy. The system basically uses predetermined contours information to detect humans and assign a threshold to determine whether the detected object is a human and has resulted in a respectable accuracy of 80%.

2.2. Challenges and difficulties in human detection and counting

In a comparative research paper (Raghavachari et al., 2015), the main difficulties in vision-based human object detection and counting applications are its accuracy in different scenarios and conditions. Different camera orientations will obtain distinctive angles of the human objects which require a different type of implementation to perform the detection and counting. Moreover, the accuracy of detection commonly changes with different people density as crowd size can be increased or decreased throughout the day. Occlusion happens when human objects are too concentrated with overlaying, causing difficulty to distinctly detect each object separately. In addition, lighting or weather conditions might also decrease the quality of images sent for detection such as blurred caused by the camera glare. Therefore, different types of algorithms or techniques should overcome these four common issues to be compatible and suitable for various experimental environments. Another concern in real-time human detection and counting is the high computational efficiency to execute multiple image frames at one time.

To resolve the main occlusion problem, Reis (2014) has mentioned three methodologies in a similar research paper. First, trajectory clustering tracks and identifies human object over a period of time. This allows a further and more in-depth clustering on overlaying objects but might not be suitable for real-time due to high computation. Second, feature-based regression is the most commonly used. After background subtraction and feature extraction are performed, this information is sent to a regression model or function for further and more accurate evaluation. Third, individual pedestrian detection performs detection individually and counting is performed based on the total number of detected human objects. This method typically requires detection of a full human body which results in lower computational performance. Besides, this paper further identifies another difficulty to separate objects from uninterested region which can be simply resolved with advanced and more expensive hardware like stereo-vision camera to separately extract different image properties using the two lenses and perform background subtraction straight away.

2.3. Computer vision-based approaches

Jalled and Voronkov (2016) perform human body and vehicle detections using image processing to perform background subtraction and obtain the foreground mask using corresponding shape features to detect human and vehicle objects. The initial result shows many false detections for human object due to

commonly shared shapes with other objects. Therefore, the face detected from human object is sent to the Haar classifier to further determine the human object using a series of sequential classifiers where the object must pass all the classifiers to obtain a true result. Based on the conducted experiment, Haar classifier only works for face detection while the performance drops on vehicle detection due to its characteristics. The tracking on face or human object is performed using simple constant facial features like color which does not change when the object rotates or moves. However, these limited tracking features can suffer through light and camera issues.

Al-Zaydi et al. (2018) proposed another image processing-based technique using regression model, low-level feature representation, and perspective normalization with frame-to-frame analysis. First, the Gaussian mixture model algorithm is used to remove unwanted noises and background. Then, the Gaussian process regression (GPR) model is trained with identified features and crowd size like foreground segment, texture, edge, and key point features which could be extracted using image processing through translation and rotation. Besides, prospective normalization helps to resolve the issue of people size changes from different camera angles by allocating different weights to the pixels at different locations. With these human-based features extracted and a trained GPR model, human counts in a particular frame are estimated with little processing. The result errors were measured using mean absolute error and mean squared error matrixes and resulted in an acceptable margin with slight difference between the estimated count and true count.

Alternatively, Al-Zaydi et al. (2016) have proposed another image processing method using multiplexer cascade model and multiple independent people detectors to enhance true positive detections. Different detectors are equipped with independent feature extractions, deep learning method, and human models. Furthermore, the confidence level of each detector is used to classify the detected window into four levels where only the highest level will be considered for counting. Different from normal cascade Haar classifier, the multiplexer cascade solution is more advanced and fuses the different confidence levels of each detector with three defined quality thresholds. When experimenting the two different detectors separately, Haar-like detector resulted in higher miss rate because of low confidence level while full-body detector resulted in much lesser miss rate. Ultimately, the paper has proven that applying deep learning and combining both detector obtains the best accuracy, highest true positive detections and lowest missing rate compared to using single people detector. However, this approach requires extensive cascading processes and might lack real-time processing capability although a pipeline is used.

In another paper (Li et al., 2014), a human counting method using head detection and tracking is proposed using both image processing and deep learning algorithms. First, the foreground region of moving human is extracted and optimized using the VIBE algorithm. As the next step, the extracted foreground region is sent to the local binary pattern based and pretrained Adaboost classifier for head detection. Third, the detected head objects are tracked around the crossline local area using a local mean-shift tracking algorithm. With the head object's position information, the final counting is performed using crossline judgment where number of people is determined when a head object intrudes the line from different moving direction. The approach is also proven with best real-time performance and accuracy compared to other histogram of oriented gradients (HOG)-based Adaboost classifier and SVM method that are higher level and more computational. This paper has also shown the importance of training classifier with quality data to outperform advanced methods. Although head detection is fast, its limited and commonly shared features with other objects could cause high false positive detection.

Distinctively, Nikouei et al. (2018) introduced another full deep learning method using lightweight convolutional neural network (L-CNN) that is both accurate and real-time capable even on edge device like Raspberry PI. Single shot multi-object detection (SSD) has been applied to architecture design which has less layers and complexity while remaining low computation and memory characteristic of L-CNN. The proposed L-CNN algorithm consists of 26 layers separating different depthwise separable convolutions and pointwise separable convolutions. The conventional convolution begins in the first layer while the rest layers are followed by separated network depthwise and pointwise convolutions. The final pooling layer performs downsizing and parameter striding during filtering. The excluded final classifier, softmax, and regression layers are then used to assign bounding box to detect objects with probability labels. The comparative experiments with other deep learning algorithms like SSD, GoogLeNet, Haar-Cascade, and HOG + SVM have shown that the L-CNN approach has the best balance between accuracy and computation efficiency compared to deep neural network approaches.

In addition, Sumit et al. (2020) performed a detailed comparison on another two fast deep learning algorithms known as Mask region-based CNN (R-CNN) and "You Only Look Once" (YOLO). R-CNN integrates region proposals with CNN. It can detect objects with deep neural networks and train high-density model with minimum annotated data. Mask R-CNN is an extension from L-CNN, R-CNN, and Faster R-CNN that performs pixel-to-pixel alignment in the bounding box known as segmentation masks. YOLO is a single-stage object detector equipped with a specific CNN network that detects and locates objects at first glance. The result shows that YOLO can detect all human objects in the first stage with lesser time complexity, computational requirements, and surprisingly better accuracy. Oppositely, Mask R-CNN only obtains certain objects in each stage although new object could be detected when the stage increases. Despite Mask R-CNN is more trained, it cannot detect certain tiny human objects where YOLO did. Lastly, the study did mention that Mask R-CNN can still be further enhanced with more convolution layers and training, but computational efficiency will be gradually sacrificed causing real-time human detection to be impractical.

2.4. Similar system comparisons

This section performs comprehensive analysis and comparison between commercial systems that are already deployed in the market which are offering similar features as the proposed system in this paper. The four popular and reliable similar systems or solutions selected from different Information Technology companies are Prodco, Vivotek (Wiangtong et al., 2012), Xovis (Jens et al., 2021) and Hikvision (Mudongo, 2021).

Tables 1 and 2 have comprehensively shown the comparisons of different criteria on the four popular similar systems in the existing market. All the systems have similar deployment environment like being ceiling mounted on a shopping mall entrance and use computer vision-based approach to handle huge crowds that require detection and counting on larger and faster scale. Similar to the approaches reviewed in research papers, these systems use image processing techniques or fast deep learning algorithms for preprocessing, human detection, and counting to reduce computational resources. Based on the research, almost all commercialized systems use GPU as the target machine due to the

Table 1
Comparison table of Prodco and Vivotek similar systems

Criteria	Prodco	Vivotek
Hardware requirements	(1) 3D stereo camera (2) Bluetooth tag	(1) 3D stereo camera
Implementation methods	Not mentioned	Not mentioned
Accuracy	High	High
Number of functionalities	9	5
Target user	Only retail stores	Only retail stores
Installation and configuration	Ceiling mount	Ceiling mount
Coverage area	Moderate	Moderate
Cost	High	High

Table 2
Comparison table of Xovis and Hikvision similar systems

Criteria	Xovis	Hikvision
Hardware requirements	(1) Dual wide lens 3D stereo camera	(1) Dual-lens 3D stereo camera
Implementation methods	(1) Image processing (2) Neural network (3) Deep learning	(1) Deep learning
Accuracy	High	High
Number of functionalities	11	4
Target user	Airports transportations retail stores	Retail stores and shopping malls
Installation and configuration	Ceiling mount	Ceiling mount
Coverage area	High due to wide lens	Moderate
Cost	High	Low

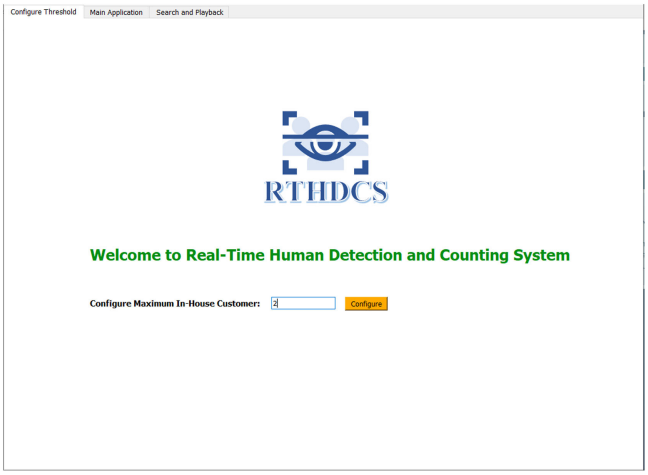
superiority in computational power and real-time processing capabilities. Specifically, these systems focus on criteria such as cost, efficiency, detection and counting speed, accuracy, space occupation, and most importantly in-store analysis for business intelligence. The major focus of the current conducted work is the mainly the cost space occupation, accuracy, and performance without stereoscopic camera for affordable hardware requirements. To remain certain return of investment and management features in business perspective, the proposed system develops a front-end management system for human detection and counting visualization as well as crowd prediction for rent estimation.

3. Methodology

3.1. Overall system architecture

The proposed system has three primary counting features such as to count the number of people who have left, entered, and are present in the shopping malls. To achieve these, only the people crossing the intrusion line needs to be counted by determining whether it is an up to down direction or the opposite. For visualization, detected human objects are drawn with a bounding box and a unique identification number for sequence indication. A secondary warning feature is also implemented to alert user when crowd size has exceeded the pre-set threshold. Special features

Figure 1
Proposed system threshold configuration tab



like setting the shopping mall's crowd threshold, parallel video recording for future search and playback, selection of normal detection or faster detection module for speed, and accuracy trade-off are included. Ultimately, all these functions are centralized into one single graphical user interface (GUI) platform with three embedded tabs such as configure threshold, main application, search and playback to enhance convenience, and data integrity as shown in Figures 1, 2, and 3. The system architecture is designed in a cascade approach where a human object is first detected then being tracked throughout the frames. The approach performs multi-object tracking and effective counting and, moreover, handles cases where a new human object appears in the frame and assigns new different detection accordingly. Therefore, this requires two important generic elements including an individual human detector that performs human detection and classification, and a tracker that performs human tracking on detected object which are further elaborated in the following sections.

3.2. Human object detection and classification

The first element is to perform object detection and only returns human objects for tracking by filtering using object classification. To achieve real-time capability, YOLOv3 is selected which is a deep learning convolutional neural network algorithm that performs both detection and classification in a single stage. The YOLOv3 model used is trained on the advance and famous COCO dataset with the target's weight and height set to 416. This configuration has a respectable accuracy and performance with 55.3 mean average performance and 35 frame per second (FPS) compared to other fast detection algorithms claimed by the model's official website (Redmon & Farhadi, 2021). However, the experimented machine is not capable to achieve the claimed FPS due to low hardware specification and the addition of advanced DeepSORT tracking algorithm. Therefore, another YOLOv3-tiny model with the same target width and height is added to as a detection module option giving users to choose the trade-off between speed and accuracy. In fact, another fast algorithm known as MobileNet SSD was also experimented, but its speed and accuracy are both lower than the YOLOv3-tiny model. To ensure the YOLOv3 models run efficiently on GPU with better performance, it is converted into TensorFlow format which basically turns the data into graph visualizations. The load of using central processing

Figure 2
Proposed system main application tab

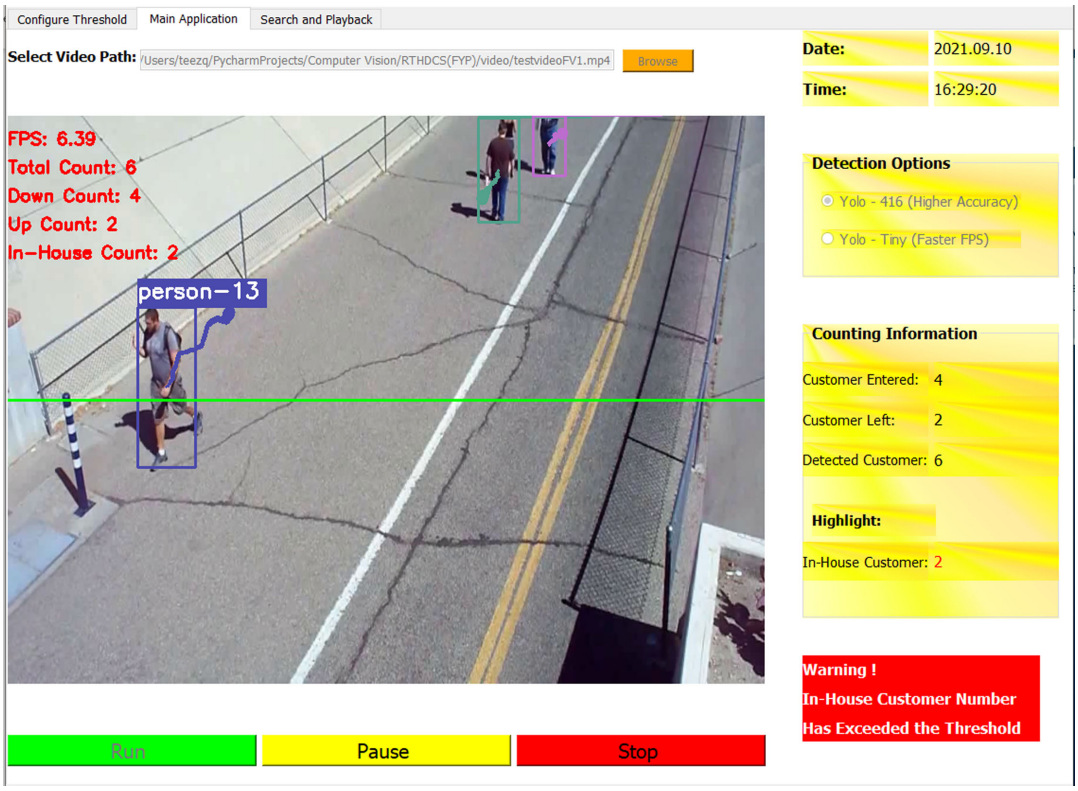


Figure 3
Proposed system search and playback tab

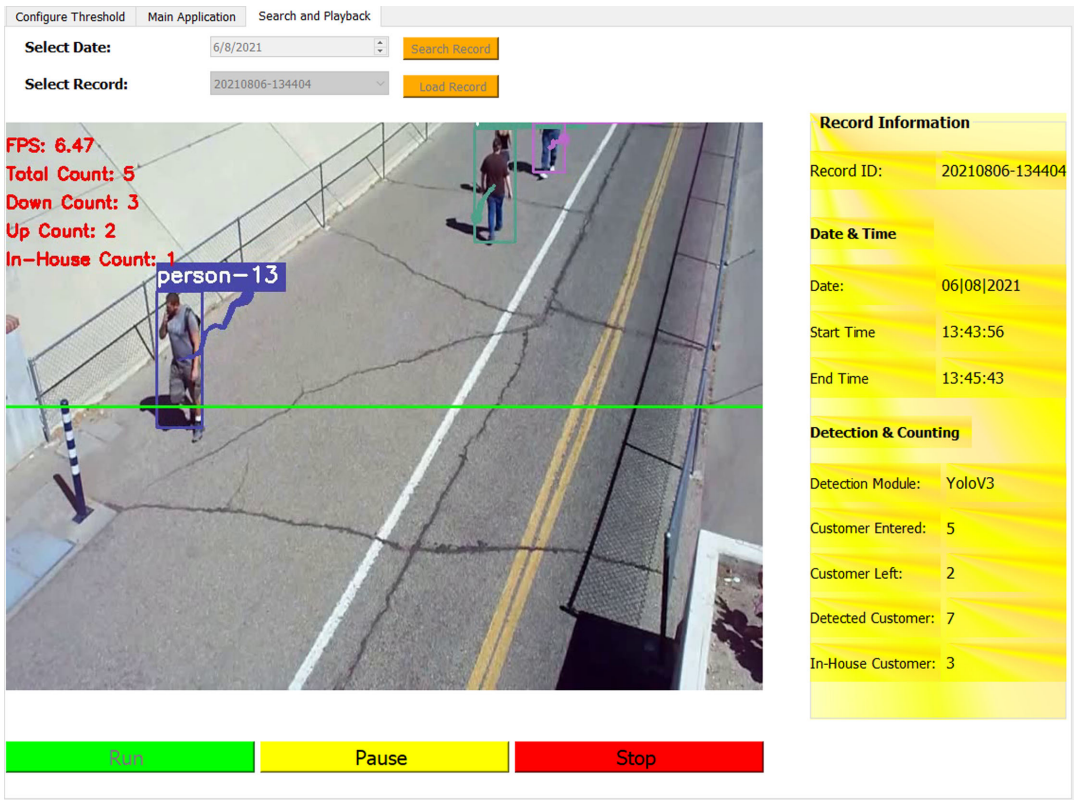
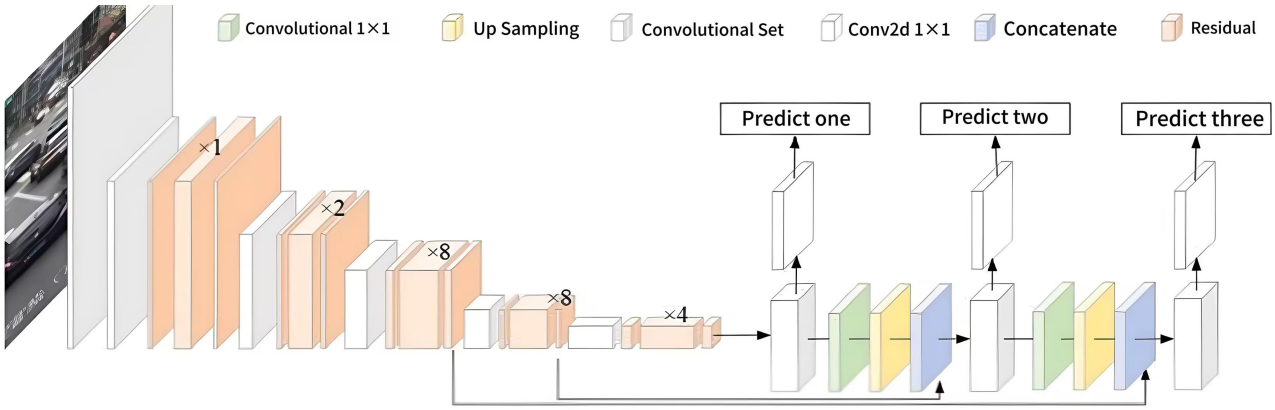


Figure 4
YOLOv3 neural network convolutional layers from Mao et al. (2019)



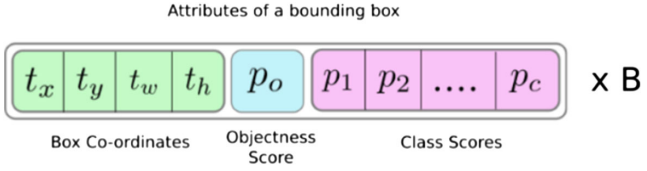
unit (CPU) is optimized by using open-source Google's machine learning technology (javaTpoint, 2021). For both selected YOLOv3 models, the result evaluation will be performed at the last three network layers which return the object with a list of possible classes and its probability as illustrated in Figure 4.

As a result, the detector should return four information such as the bounding box's top-left coordinate, bottom right coordinate, class label, and confidence level from the attributes shown in Figure 5. Both "t_x" and "t_y" are the detected or targeted object center x and y coordinates while "t_w" and "t_h" are the width and height. These raw information can be used to extract the mentioned top-left and bottom right coordinates for bounding box drawing. Only the class with the highest probability and confidence layer will be finalized and returned which is referred as the "P_0" or "objectness score" in Figure 5 while the remaining "P_1", "P_2" until "P_c" are just probability scores for other individual classes in the pretrained YOLOv3 model with COCO dataset.

3.3. Human object tracking

Tracking is another important element for all the human counting features using the intrusion line to judge the object movement from a particular direction. In general, object tracking is the process to assign a unique identification to the detected human object and perform constant tracking when it moves throughout the frames until the object is disappeared. Detection cannot be solely run in every frame as it cannot connect the same object being detected in the previous frame especially when handling multiple objects. Furthermore, when an object exits the detection and reappears, it cannot be determined that whether it is the same object that appeared in previous frames or a completely new object. Also, comparing the current object's movement with the past one is not possible. Hence, tracking is mandatory for computer vision applications including the proposed system that run on real-time video data. DeepSORT is selected as the system's tracker which is an extension of Simple Online and Real-Time Tracking (SORT) algorithm. The simple SORT technique first uses Kalman filtering to estimate the object's motion prediction on the next frame. The bounding box's Intersection Over Union (IoU) is then calculated to determine the similarity between the tracked object and the detection object. According to the IoU distance, the final assignment is solved using Hungarian method to locate the tracked object new location. SORT is proved to be simple and fast while having respectable accuracy. However, most computer vision projects or systems do not adapt the simple SORT technique as it cannot resolve occlusion when two or more objects overlap with each other. Also, new identification no will be assigned to the same object because of combining multiple object bounding boxes that are overlapping. Therefore, the "Deep" characteristic has to be added to improve the tracking accuracy with the use of objects' appearance features rather than just motion prediction. Hence, this is how the method name "DeepSORT" is formed. It uses an additional visual appearance descriptor to perform feature generation on the detected object using its pretrained convolutional neural network. In other words, the algorithm already remembers the particular object once it firstly appears through feature extraction techniques. When an object is occluded with another, DeepSORT algorithm can help to justify and differentiate the objects through different features obtained, thus maintaining a better constant track of an object and even when it disappears and reappears in the frames. Furthermore, the historical path of each object will also be stored to justify its moving direction for counting purposes. For example, if an object moves from the top to bottom, it indicates that a human or customer has entered the shopping mall, thus add one count to the entered customer count and in-house customer count. The concept works opposite when an object moves from the bottom to the top when crossing the intrusion line.

Figure 5
YOLOv3 neural network convolutional layers from Mao et al. (2019)



4. Experimental Results

4.1. Experimental setup

In this study, the proposed system is experimented on a local computer's GPU with selected dataset. In fact, the system is also

experimented with the CPU as the targeting machine but is quickly unconsidered as GPU has shown way much superiority in performance. The five individual testing videos ranging from 20 s to 2 min are taken from the internet with different camera orientations simulating the shopping mall entrance scenario. The computer hardware specifications and system software requirements are listed as below.

(1) Hardware:
   (a) CPU – Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz
   (b) GPU – Nvidia GTX 1050 4GB with 6.1 CUDA compute compatibility
   (c) RAM – 16GB
(2) Software:
   (a) Python 3.7
   (b) CUDA Toolkit Version 10

4.2. Accuracy evaluation

As mentioned, only the up count and down count of line intrusions are important for counting purposes. Tables 3 and 4 record the expected or truth count evaluated through precise manual counting and the actual count estimated by the proposed system for the five testing videos with different camera orientations. To obtain the final accuracy percentage, the error count should first be calculated which is basically the differences between the expected count and actual count. The final accuracy is 91.07% for the normal more accurate YOLOv3 model and 76.8% for the faster YOLOv3-tiny model; both calculations are detailly and individually demonstrated below.

$$\begin{aligned}\textbf{Error rate} &= \text{sum of error count}/(\text{sum of expected up count and down count}) \times 100\% \\ &= (0+1+1+1+2)/(1+\%+7+4+12+12+3+7+3+5) \times 100\% \\ &= (5/56) \times 100\% \\ &= 0.0893 \times 100\% \\ &= 8.93\%\end{aligned}$$

Table 3
Accuracy result of normal YOLOv3 model with higher accuracy

No.	Camera orientation	Expected up count	Expected down count	Actual up count	Actual down count
1	Front view	2	5	2	5
2	Front view	7	4	8	4
3	Overhead view	12	12	12	13
4	Overhead view	3	7	3	6
5	Overhead view	3	5	2	4

Table 4
Accuracy result of YOLOv3-Tiny with faster performance

No.	Camera orientation	Expected up count	Expected down count	Actual up count	Actual down count
1	Front view	2	5	3	4
2	Front view	7	4	4	2
3	Overhead view	12	12	11	10
4	Overhead View	3	7	3	7
5	Overhead view	3	5	1	6

$$\begin{aligned}\textbf{Overall accuracy} &= \text{total expected count} - \text{error rate} \\ &= 100\% - 8.93\% \\ &= 91.07\%\end{aligned}$$

$$\begin{aligned}\textbf{Error rate} &= \text{sum of error count}/(\text{sum of expected up count and down count}) \times 100\% \\ &= (2+5+3+0+3)/(1+5+7+4+12+12+3+7+3+5) \times 100\% \\ &= (13/56) \times 100\% \\ &= 0.232 \times 100\% \\ &= 23.2\%\end{aligned}$$

$$\begin{aligned}\textbf{Overall accuracy} &= \text{total expected count} - \text{error rate} \\ &= 100\% - 23.2\% \\ &= 76.8\%\end{aligned}$$

4.3. Performance evaluation

As shown in Tables 5 and 6 and Figures 6, 7, the performance of each model's testing videos is evaluated through the average FPS

Figure 6
FPS performance screenshot on experiment using normal YOLOv3 model with higher accuracy

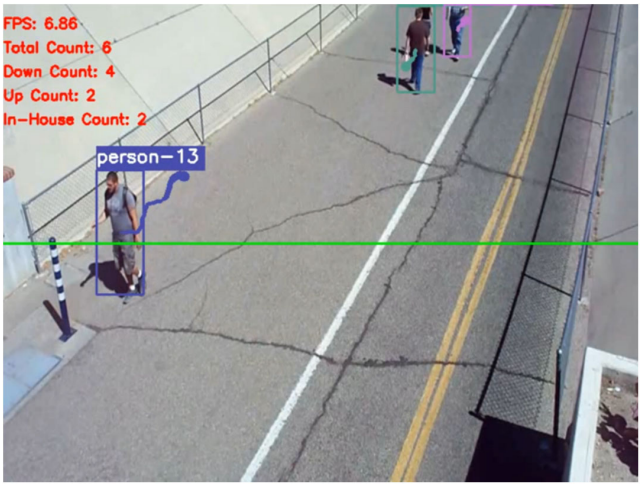


Figure 7
FPS performance screenshot on experiment using YOLOv3-tiny with faster performance

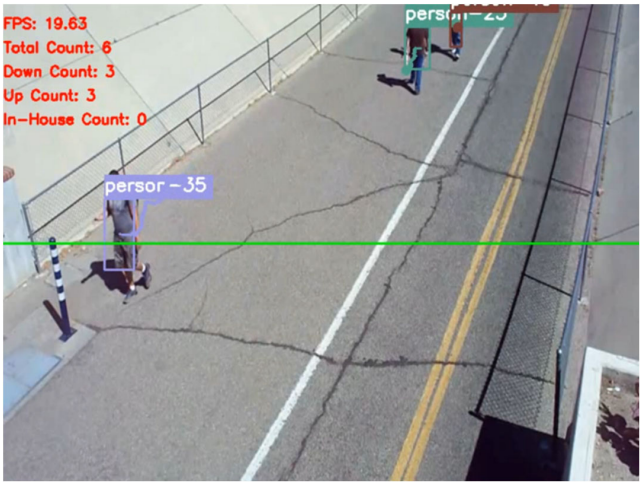


Table 5
Performance result of normal YOLOv3 model with higher accuracy

No.	Camera orientation	Average performance
1	Front view	7 FPS
2	Front view	5 FPS
3	Overhead view	7 FPS
4	Overhead view	7 FPS
5	Overhead view	7 FPS

Table 6
Performance result of YOLOv3-tiny with faster performance

No.	Camera orientation	Average performance
1	Front view	30 FPS
2	Front view	25 FPS
3	Overhead view	34 FPS
4	Overhead view	34 FPS
5	Overhead view	37 FPS

observed during the runtime. In other words, how many frames or images was the proposed system capable to process when switching between different testing videos and model. Ultimately, the final average FPS performance of the normal more accurate YOLOv3 model is 6.6 and 32 FPS for the faster YOLOv3-tiny model. Again, both calculations are individually shown below.

$$\begin{aligned}\textbf{FPS performance} &= \text{sum of average performance/total amount of testing videos}\\ &= (7+5+7+7+7)/5\\ &= 33/5\\ &= 6.6\,\text{FPS}\end{aligned}$$

$$\begin{aligned}\textbf{FPS performance} &= \text{sum of average performance/total amount of testing videos}\\ &= (30+25+34+34+37)/5\\ &= 160/5\\ &= 32\,\text{FPS}\end{aligned}$$

4.4. Summary

The experimental results have clearly shown the accuracy and performance trade-off between the normal YOLOv3 and YOLOv3-tiny model. Undeniably, the normal YOLOv3 model has a more promising average accuracy where there are little situations where the human object is not properly detected. On the other hand, the YOLOv3-tiny model has advantages in smaller size or architecture and faster inference speed which helps it to achieve better real-time processing performance. Unexpectedly, this study discovered that the YOLOv3-tiny model actually has similar error counts as the more accurate normal YOLOv3 model when testing with overhead view videos. In fact, surveillance cameras in shopping malls are usually ceiling mounted which simulates overhead view and make the YOLOv3-tiny model a more practical solution especially with lower computational requirements. In fact, only the YOLOv3-tiny model has met the actual real-time performance with 32 FPS as the original average FPS of the five testing videos is around 24–30 FPS. Oppositely, running the normal and more accurate YOLOv3 model either takes way more time than the original video duration or incapable to process all the frames in every seconds.

5. Conclusion

In conclusion, the automation of the proposed system has proven to be a more efficient, effective, and accurate crowd management solution as compared to manual operations. Despite the hardware limitations, this study has achieved the actual real-time performance with an additional faster YOLOv3-tiny model. Furthermore, it achieves a high topping accuracy of 91.07% with the normal YOLOv3 model. The demonstration of accurate deep learning YOLOv3 and DeepSORT algorithms has resolved common challenges in human detection and counting like different camera orientation, people density, lighting, and occlusion. The unique contribution of this proposed system compared to existing and similar systems is the direct compatibility with existing surveillance camera without new hardware or space occupation requirements. In addition, a user-friendly and novice GUI is used to configure, run, search, and playback the human detection and counting. Besides, the strong advantages of using GPU for computer vision applications and converting pretrained YOLOv3 model into TensorFlow format from original weight and configuration files for even faster processing are proven. The main limitation of the proposed system is low resolution in certain testing videos where human objects are unclear and difficult to be detected especially when using the YOLOv3-tiny model. In addition, the Google TensorFlow format is only currently supported by Nvidia GPU machines with Python implementation. Ultimately, the proposed system still has big room of improvements where an even more accurate YOLOv3 detection could be added with the target width and height set at 608 if a better GPU specification is available. In fact, GPU is no longer an expensive requirement with Nvidia releasing their affordable Jetson Nano series, which is basically a small but powerful computer with dedicated GPU specialized for processing in computer vision applications. This technology availability does not only save the cost of purchasing a complete computer set for hosting but also greatly reduces the space occupation for such system to be easily deployed in real environment. Moreover, an additional machine learning algorithm can also be included to constantly adapt and learn new human objects detected in the deployment environment; hence, it further improves the accuracy.

Conflict of Interest

The authors declare that they have no conflicts of interest to this work.

References

Al-Zaydi, Z., Vuksanovic, B., & Habeeb, I. (2018). Image processing based ambient context-aware people. *International Journal of Machine Learning and Computing*, *8*(3), 268–273. https://doi.org/10.18178/ijmlc.2018.8.3.698

Al-Zaydi, Z., Ndzi, D., & Sanders, D. (2016). Cascade method for image processing based people detection and counting. In *Proceedings of International Conference on Image Processing, Production and Computer Science*, 30–36.

Bernama. (2021). *RM10,000 fine for SOP violations beginning March 11*. Retrieved from: https://www.nst.com.my/news/nation/2021/02/669100/rm10000-fine-sop-violations-beginning-march-11

Congrex Team. (2020). *Disruption in the business events industry: rising to the challenges of Covid-19*. Retrieved from: https://congrex.com/blog/disruption-business-events-industry-challenges-covid-19/

Jalled, F., & Voronkov, I. (2016). Object detection using image processing. *arXiv Preprint:1611.07791*

Jens, K., & Gregg, J. S. (2021). The impact on human behaviour in shared building spaces as a result of COVID-19 restrictions. *Building Research & Information*, *49*(8), 827–841. https://doi.org/10.1080/09613218.2021.1926217

javaTpoint. (2021). *Advantage and disadvantage of TensorFlow*. Retrieved from: https://www.javatpoint.com/advantage-and-disadvantage-of-tensorflow

Li, B., Zhang, J., Zhang, Z., & Xu, Y. (2014). A people counting method based on head detection and tracking. In *2014 International Conference on Smart Computing*, 136–141.

Liu, G., & Sun, Y. (2012). An in-vehicle system for pedestrian detection. In *2012 11th International Symposium on Distributed Computing and Applications to Business, Engineering & Science*, 328–331.

Mao, Q.-C., Sun, H.-M., Liu, Y.-B., & Jia, R.-S. (2019). Mini-YOLOv3: Real-time object detector for embedded applications. *IEEE Access*, *7*, 1–9. https://doi.org/10.1109/ACCESS.2019.2941547

Mokayed, H., Meng, L. K., Woon, H. H., & Sin, N. H. (2014). Car plate detection engine based on conventional edge detection technique. In *Proceedings of the International Conference on Computer Graphics, Multimedia and Image Processing*, 101–106.

Mokayed, H., Shivakumara, P., Woon, H. H., Kankanhalli, M., Lu, T., & Pal, U. (2021). A new DCT-PCM method for license plate number detection in drone images. *Pattern Recognition Letters*, *148*, 45–53. https://doi.org/10.1016/j.patrec.2021.05.002

Mudongo, O. (2021). *Work in progress in computer vision and AI surveillance in Africa*. Retrieved from: https://africaportal.org/publication/work-progress-computer-vision-and-ai-surveillance-africa/

Nikouei, S. Y., Chen, Y., Song, S., Xu, R., Choi, B.-Y., & Faughnan, S. Y. (2018). Real-time human detection as an edge service enable by a lighweight CNN. In *2018 IEEE International Conference on Edge Computing*, 125–129.

Piciarelli, C., & Foresti, G. L. (2011). Surveillance-oriented event detection in video streams. *IEEE Intelligent Systems*, *26*(3), 32–41. https://doi.org/10.1109/MIS.2010.38

Raghavachari, C., Aparna, V., Chithira, S., & Balasubramanian, V. (2015). A comparative study of vision based human detection techniques in people counting applications. *Procedia Computer Science, 58*, 461–469. https://doi.org/10.1016/j.procs.2015.08.064.

Redmon, J., & Farhadi, A. (2021). *YOLO: Real-time object detection*. Retrieved from: https://pjreddie.com/darknet/yolo/

Reis, J. V. (2014). *Image descriptors for counting people with uncalibrated cameras.* Master's Thesis, University of Porto.

Sumit, S. S., Watada, J., Roy, A., & Rambli, D. R. A. (2020). In object detection deep learning methods, YOLO shows supremum to Mask R-CNN. In *The 2nd Joint International Conference on Emerging Computing Technology and Sports*, 1–8.

The Straits Time. (2020). *Malaysia issues list of malls, hypermarkets seen as potential Covid-19 hot spots using new AI system*. Retrieved from: https://www.straitstimes.com/asia/se-asia/malaysia-lists-names-of-malls-hypermarkets-seen-as-covid-hotspots-using-new-ai-system

Velastin, S. A., Fernández, R., Espinosa, J. E., & Bay, A. (2020). Detecting, tracking and counting people getting on/off a metropolitan train using a standard video camera. *Sensors*, *20*(21), 6251. https://doi.org/10.3390/s20216251

Wiangtong, T., & Prongnuch, S. (2012). Computer vision framework for object monitoring. In *2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 1–4. https://doi.org/10.1109/ECTICon.2012.6254291

How to Cite: Mokayed, H., Quan, T. Z., Alkhaled, L., & Sivakumar, V. (2023). Real-Time Human Detection and Counting System Using Deep Learning Computer Vision Techniques. *Artificial Intelligence and Applications*, *1*(4), 221–229, https://doi.org/10.47852/bonviewAIA2202391