Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Detecting and Tracking People in Crowded Environments with YOLOv7 and DeepSORT

Devis kabangira[1], Yoosoo Oh[2]

1.*Department of Computer and communication Engineering, Daegu University, Gyeongsan 38453, Republic of Korea* [1] *dvskabanngira001@gmail.com*
2.School of Artificial Intelligence , Daegu University, Gyeongsan 38453, Republic of Korea

Abstract. In recent years, human detection and tracking have become vital in various areas of Artificial intelligence, such as behavior analysis, autonomous driving, and security surveillance. However, detecting human beings in a crowded environment has been challenging due to occlusions and background noise problems. This paper presents an innovative human detection and tracking approach by leveraging the powers of a one stage detector technique YOLO(You Only Look Once)v7 for human detection and DeepSORT for multi-people tracking. DeepSORT uses the Kalman filter and Hungarian algorithm for tracking. The occlusion issue is solved using the Kalman filter. YOLOv7 is among the latest YOLO object detection families and is known for its best accuracy and speed in real-time applications. We achieved significant improvements in detection accuracy by integrating it with DeepSORT, a robust tracking algorithm that combines detection with the ability to track individuals' movements over time. Our model was trained from scratch and evaluated on the public coco dataset. The experimental results show that our model outperforms other detection algorithms regarding precision, recall and mean average precision(mAP).
.

Keywords: YOLOv7, Deep learning, DeepSORT, Human Detection and Tracking.

1. Background

The detection of humans in video surveillance systems is becoming increasingly important due to its numerous applications in abnormal event detection, human gait characterization, person counting in a dense crowd, person identification, gender classification, and fall detection for the elderly [1]. Human detection and tracking have been a core area of research in computer vision for some time. Human detection aims to identify all instances of humans present in an image. Human tracking is associating human detections with a video sequence to generate persistent paths or trajectories of the people[2]. Human detection and tracking are generally considered the first two processes in an AI-based video surveillance pipeline. They can feed into higher-level reasoning modules such as action recognition and dynamic scene analysis. The process involves identifying and monitoring individuals within a given environment, providing valuable information for various applications, including security, surveillance, and human-computer interaction.

Reliable and robust detection algorithms must be developed to accurately detect and track human beings in a crowded setting in real-time. In this research, we adopt a state-of-the-art detection algorithm, YOLOv7 for detection, and DeepSORT for tracking purposes. YOLOv7 is one of the fastest and most accurate real-time detection models for computer vision tasks[3]. It offers a more effective integration approach, precise object detection performance, robust loss function, and improved label assignment and model training efficiency[4].

The number of parameters and the computational complexity of a model are the two primary considerations for constructing efficient layer aggregation networks in the backbone of YOLOV7. YOLOV7 integrates several strategies, including E-ELAN (Extended Efficient Layer Aggregation

Networks) [5, 6], model scaling for concatenation-based models [7], and model reparameterization [8], in order to achieve a balance between detection efficiency and precision. The Yolov7 network comprises four key modules, namely the Input module, the Backbone network, the Head network, and the Prediction network. The Input module encompasses the pre-processing stage of the YOLOv7. At this level, the model employs both mosaic and hybrid data enhancement techniques and leverages the adaptive anchor frame calculation method initiated by YOLOv5 to uniformly scale the input color images to a size of 640 × 640.

The backbone network comprises three major components: Convolution Block series (CBS), E-ELAN, and Max pooling 1x1 (MP1). The CBS comprises convolutions, a batch size, and a siLU activation function. The E-ELAN enhances the network's learning ability by guiding different feature group computational blocks to learn more diverse features while preserving the original gradient path. The MP1 contains CBS and MaxPool. MP1 is divided into an upper and lower part. The upper part uses MaxPool to halve the length and width of the image and CBS with 128 output channels to halve the image channels. The lower part of the network halves the image channels through a CBS with a 1 × 1 kernel and stride, halves the image length and width with a CBS of 3 × 3 kernel and 2 × 2 stride, and finally fuses extracted features from both parts through the concatenation (Cat) operation.

The head network uses a Future Pyramid Network (FPN) architecture. The network comprises several convolutions, batch normalization, and siLU activation blocks, in addition to spatial pyramid pooling and convolution spatial pyramid pooling (SSPCSPC) and Maxpool-2 (MP2). The SSPCSPC structure enhances the perceptual field of the network by incorporating a convolution spatial pyramid (CSP) within the spatial pyramid pooling (SPP) structure. Finally, the prediction network employs a rep structure to adjust the number of image channels for features output from the head network.

The YOLOv7 model also employs a novel loss function, focal loss which makes it effective in detecting small objects. Small objects can be easily detected by down-weighting the loss for all well-classified examples and focusing on the complex examples, that is; objects that are hard to detect. YOLOv7 also employs a strategy known as non-maximal suppression to address the issue of a more significant number of redundant predictions that occur as a result of cells forecasting the same item with different bounding boxes [9]. This strategy ignores the bounding boxes with lower probability values and considers bounding boxes with the highest probability score.

Integrating the capabilities identified in YOLOv7 with DeepSORT enhances the detection mechanism, leading to higher and more accurate detection accuracy. DeepSORT employs a bounding box overlap association metric and frame-by-frame data association using the Hungarian technique. It then applies a Kalman filter in the picture space to conduct filtering. By including the reidentification of identified objects across frames based on a pre-trained CNN to generate bounding box appearance descriptors, DeepSORT enhances resilience against misdetections and occlusion [9].

2. Related works

In the early days, researchers used traditional machine learning object detection methods and extracted features manually. Manual feature extraction is designing and processing features manually by experts in the relevant field through years of accumulation and experience [10]. Papageorigous et al.[11] proposed a feature similar to Haar and applied it to human detection. The designed trainable system for object detection derives much of its power from a representation that describes an object class in terms of an overcomplete dictionary of local, oriented, multiscale intensity differences between adjacent regions, efficiently computable as a Haar wavelet transform [11]. Felzenszwalb et al.[12] designed an object detection system based on mixtures of multiscale deformable part models, a component-based detection method highly robust to pedestrian deformation. This algorithm combined a margin-sensitive approach for data-mining complex negative examples with a latent SVM formalism. Another commonly used traditional detection method was the Oriented Gradient Histogram (HOG), proposed by Wójcikowski et al. [13]. This method counts gradient orientation occurrence in a localized portion of an image. Based on a Support Vector Machine (SVM), it was designed to detect humans. This method involved dividing a rectangular section of the image into

multiple sections resembling a Haar-like feature. However, these methods often exhibited a high false positive rate.

In recent years, convolutional neural networks (CNNs) have been proposed to address the detection-related problems that traditional machine learning algorithms face. A CNN is a deep learning model designed specifically for processing data with a grid-like topology, such as images or time series data. CNNs effectively handle tasks related to image recognition, image classification, object detection, and other visual data-related applications. This deep learning method can learn filters and characteristics after deep training, which is challenging for traditional methods to learn filters and characteristics over time. CNN is a two-stage detector. The initial stage of the method performs preliminary tests, identifies all positive samples, and generates all regions of interest (ROIs). The second stage performs regional classification and location refinement on the RoIs generated in the initial stage [14]. Examples of this method include Faster R-CNN, R-FCN, and FPN. This two-stage detector method employs a variety of algorithms to identify potential regions of interest (ROIs) around an object, which are then classified through a convolutional neural network (CNN). The two-stage approach is computationally expensive and complex. However, the advent of a one-stage algorithm (YOLO) has streamlined the algorithmic flow, making it more concise and straightforward. Unlike two-stage algorithms, YOLO does not generate regional proposals [15]. One-stage algorithms do not require the generation of region proposals in advance, which gives them a speed advantage**.**

3. Methodology

In our research work, we propose using the DeepSORT algorithm with the YOLOv7 model for Human detection and tracking in a crowded environment. This methodology section outlines the dataset used, model training process, hyperparameters, and evaluation metrics.

a) YOLOv7 model

Inspired by Wang Chien-Yao et al. [16] in their paper YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, we adopted YOLOv7 as our detector. We trained our model on a coco dataset on all 80 classes from scratch for 100 epochs. After training our model, we used the YOLO model weights to make detections on only one YOLO class at index 0, which is "person." We created a Python script and initialized the YOLO person detector class with a confidence threshold (conf_th=0.25) and an intersection over the union threshold (iou_th=0.45). Our defined YOLOv7 Detector Wrapper exposes two methods: one loads the model weights from the checkpoints, and the second accepts an image to return a tensor of detections. We also defined another method named detect() to perform the actual person detection using the YOLO model. The method first processes the input RGB image, gets all persons' detections and then, applies a non-maximum suppression technique on the detected bounding boxes to remove redundant bounding boxes, keeping only the most confident detections. Finally, the method returns an array of detected people in the format [x_min, y_min, x_max, y_max, confidence, class].

b) DeepSORT Model

After performing YOLO human detections, we feed the results into the DeepSORT model. DeepSORT is an extension of the SORT algorithm[17]. This algorithm contains four significant components: detection, estimation, association, and tracking identity and destruction. The detection model identifies the class of the target and generates bounding boxes around these targets, providing their corresponding positions and labels. The estimation component utilizes the framework of the Kalman filter to conduct predictions better, even for occluded objects. The bounding box overlap association metric and frame-by-frame data association are achieved by minimizing costs, such as using the Hungarian algorithm. DeepSORT also uses a prediction mechanism to maintain track of occluded or temporarily undetected objects until they reappear, reducing occlusion occurrence. For

every target found in a video, the DeepSORT algorithm presents a continuous set of tracks related to their motion. Also, the generated tracks contain the unique identities of targets across frames, which creates room for in-depth movement analysis. Therefore, using YOLOv7 as input, the DeepSORT algorithm provides an effective tracking mechanism for humans within a crowded environment.

4. Evaluation

A. Experimental setup.

We conducted our experiments on Ubuntu 20.04.6 LTS, NVIDIA-SMI 470.239.06, driver version 470.239.06, and CUDA version 11.4 with 2 GPUs. We also used torch 1.11.0+cu113. Since the tuning and definition of hyperparameters are crucial during the training phase, we defined our hyperparameters for the YOLOV7 model as shown in the table below;

Table 1.Used hyperparameters for training our proposed yolov7 model and other baseline models

Hyperparameter	value	Hyperparameter	value
lr0(initial learning rate)	0.01	fl_gamma(focal loss)	0.0
lrf(one cycle learning rate)	0.1	hsv_h(hue-augementation	0.015
momentum	0.937	hsv_s(img saturation aug)	0.7
Weight decay	0.0005	hsv_v(value augementation)	0.4
Warmup epochs	3.0	Degrees(image rotation)	0.0
Warmup momentum	0.8	Translate(img translation)	0.2
Warmup bias lr	0.1	Scale(image scale)	0.5
box	0.05	Shear(image shear)	0.0
cls(loss gain)	0.3	Perspective(image)	0.0
cls_pw(BCELoss positive_weight)	1.0	Flipud(img flip up-down)	0.0
Box(loss gain)	0.05	Fliplr(img flip left-right)	0.5
anchor_t(anchor multiple threshold)	4.0	Mosaic(image mosaic)	1.0
iou_t(IOU training threshold	0.20	Mixup(image mixup)	0.0
paste_in	0.0		
copy_paste(img copy pate probability)	0.0		
loss_ota(ComputeLossOTA)	0.0		

B. Evaluation metrics

In this section, we present the quantitative metrics used to evaluate the performance and effectiveness of our model. These metrics provide insight into the performance of the detection model and help to compare different models or algorithms with our model. These metrics include Precision, Recall, Mean Average Precision (mAP)

Precision: This metric is critical to model evaluation as it quantifies the accuracy of the positive predictions made by the model. Specifically, it assesses how well the model distinguishes true objects from false positives. Mathematically, it can be calculated as;

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

where TP stands for True positives and FP False positives.

Recall: Recall measures the model's capability to capture all relevant objects in the image. In essence, recall assesses the model's completeness in identifying objects of interest. A high recall score indicates that the model effectively identifies most of the relevant objects in the data. It can be computed as;

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

where, FN stands for False positives.

Mean average Precision(mAP): This metric measures a model's ability to identify and accurately localize objects in images, which are critical tasks in applications such as autonomous driving and security surveillance. Mathematically, it can be calculated as;

$$\text{mAP} = \frac{1}{\text{N}} \sum_{\text{i}=1}^{\text{N}} \text{AP}_{\text{n}}$$

where, N is the total number of images used in the inference phase.

5. Evaluation

This section presents the results of the experiments conducted using the DeepSORT algorithm for tracking and proposed YOLOv7 detection model. We also provide a detailed analysis of the performance of our proposed model compared to other state-of-the-art baseline detection models

A. Detection performance

We carried out our detection performance experiments on the proposed model to efficiently and accurately detect and identify people with occlusions in a crowded setting. We also compared the performance of our proposed YOLOv7 model with other one-stage detection models regarding Precision, Recall, and Mean Average Precision (mAP). The comparison of model performance is shown in Table 2, which shows a selection of experimental results of the baseline models we used based on detection performance, confidence values, and inference speed. From the results shown in the table, the YOLOv7 model outperformed other models based on the provided results of the detection metrics. From the table, it can also be seen that the inference time for YOLOv3-tiny is fast compared YOLOv7 and YOLOR. This is due to reduced convolution layers of the model. YOLOv3-tiny uses Darknet19 network structure with only seven convolutional layers and small number of 1×1 and 3×3 convolutional layers making it light-weight

Figure 1 shows the detection performance of our selected YOLOv7 model, the generated loss and the behavior at each epoch during training. The training and validation behavior of the model is described in detail in terms of metrics which include; Recall, Precision, mAP@0.5, mAP@0.5:0.95, as well as abjectness, classification, and box losses. Our model demonstrates its effectiveness by continuously recording the loss, Precision, Recall, and mAP metrics during the training and validation. Figure 2 demonstrates the training behavior of our baseline model YOLOvR and figure 3 also depicts YOLOv3-tiny's training behavior over a period of time. From the training results, YOLOv3-tiny recorded the lowest performance in terms of precision, Recall, mAP@0.5 , mAP@0.50:0.95. This lower performance can be attributed to fewer convolutional layers and parameters. This reduction in complexity allows for faster inference but may also limit its ability to capture intricate features and patterns in the data, leading to lower detection accuracy.

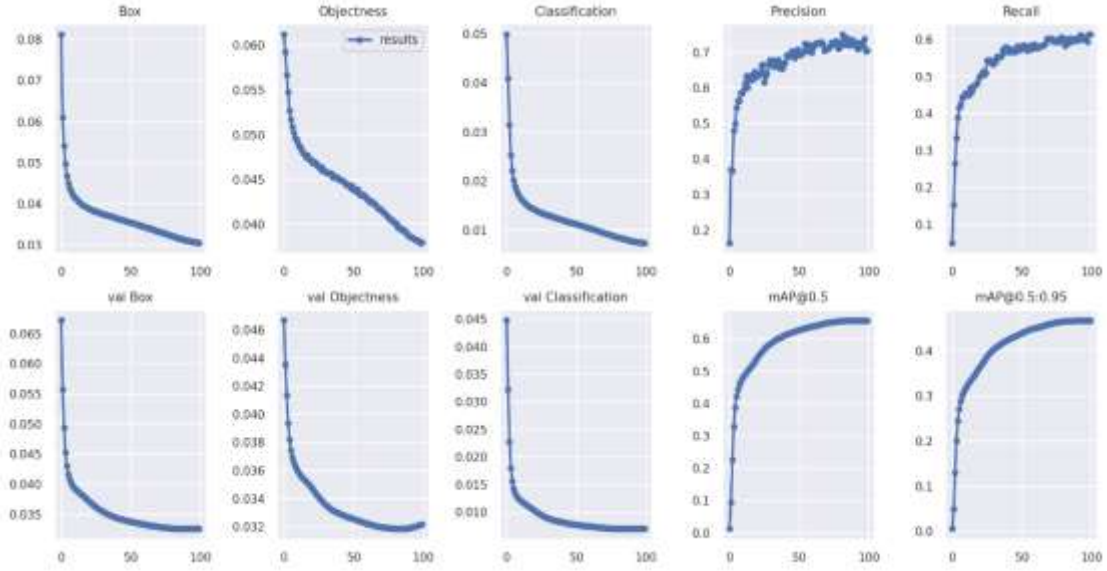


Fig 1. Performance behavior of YOLOv7 model

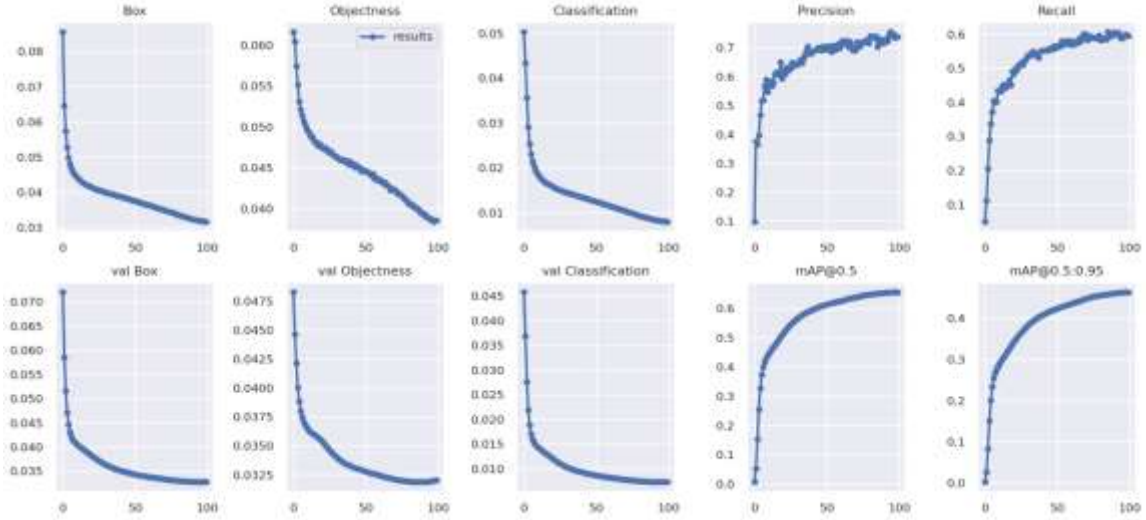


Fig 2.Performance behavior of YOLOvR model

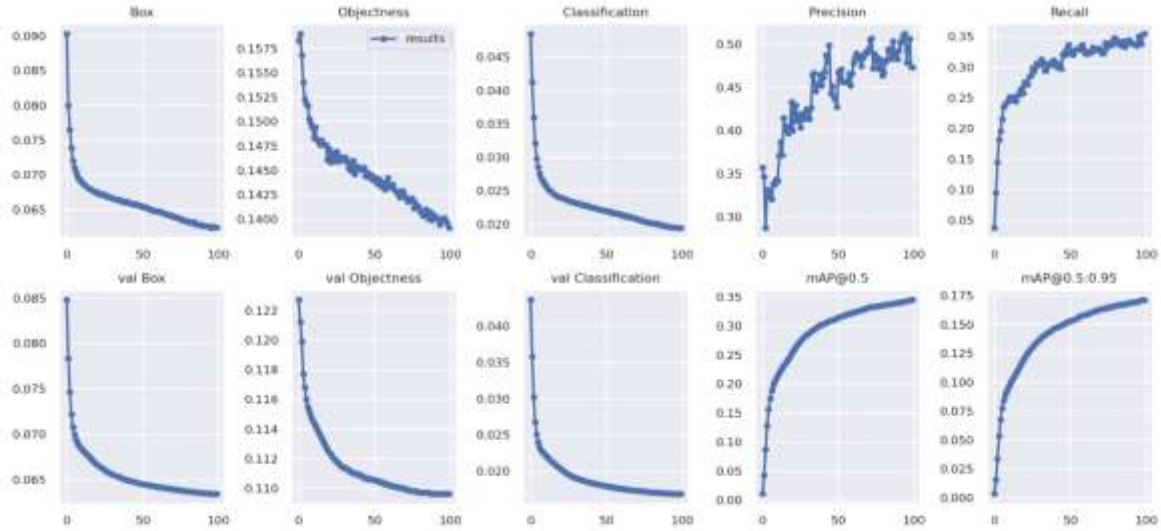


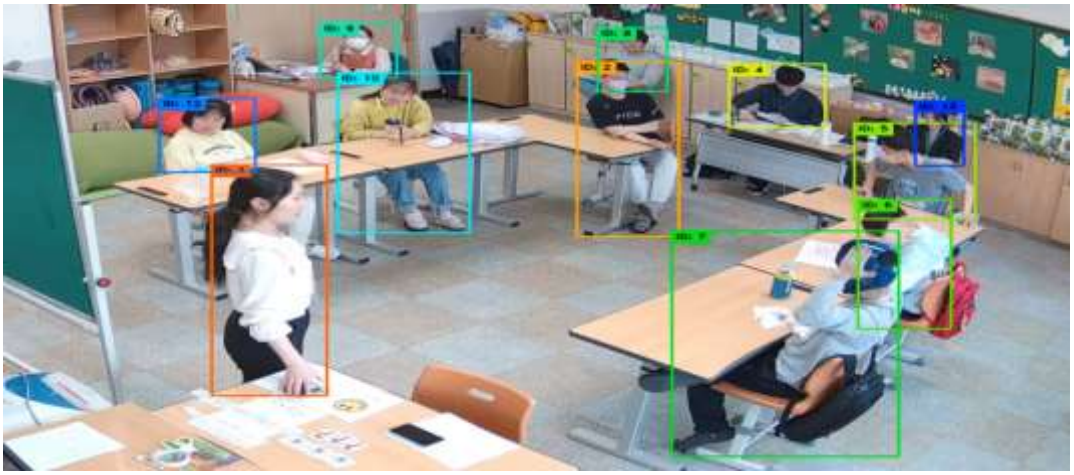
Fig 3.Performance behavior of YOLOv3-tiny model

Table 2. Experimental results of detection models

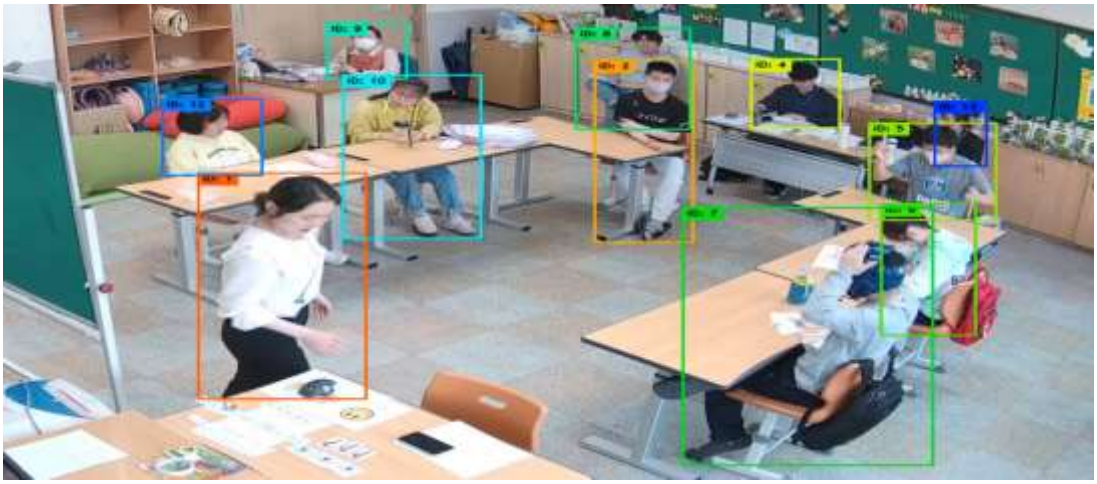
Model	Test Size	P	R	AP^{test}	AP_{50}^{test}	AP_{75}^{test}	batch 32 average time
YOLOv7	640	71.1%	60.6%	48.0%	66.1%	52.2%	4.0ms
YOLOvR	640	70.4%	59.6%	47.6%	66.0%	51.8%	4.4ms
YOLOv3-tiny	640	49.7	33.2%	18.2%	34.6%	16.8%	1.7ms

B. Detection and tracking multiple persons under occlusion

In the midst of detecting and tracking multiple people in a crowded environment, there are times when individuals are occluded by others, potentially causing the other people to disappear from view. However, occlusion is not an inherent problem for our proposed model as it can detect occlusions and overcome them by applying the Kalman filter to the object. The occurrence of occlusion and the ability of the model to overcome it and continuously track the individual can be seen in (a) and (b) of Figure 4.



(a)



(b)

Fig 4. (a) and (b) above show the Detection and tracking results of our proposed model in a crowded setting with occluded persons in motion

6. Conclusion

In this research work, we propose the use of YOLOv7 with DeepSORT algorithms to detect and track people in a crowded environment with occlusion. From the experimental results, our proposed model outperforms other baseline detection algorithms and could easily detect and track multiple persons under occlusions accurately. With significant advancement in computer vision and artificial intelligence, employing this model in detecting and tracking multiple people in a crowded environment can be a crucial task in applications related to maintaining security in places with high rates and behavioural analysis. The proposed model can solve the occlusion problem that most researchers have been facing in applications involving detection. Occlusion can significantly reduce the accuracy of human detection algorithms. When people are partially or fully occluded, their appearance can be altered or obscured, making it difficult for detection systems to recognize and locate them correctly which can lead to missed detections or false positives. This model was also able to continue tracking people even when occlusion occurred in sequence. In the future, we intend to incorporate more detection algorithms and develop a surveillance system.

Acknowledgment

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea(NRF-2022S1A5C2A07091326)

References

[1] M. Paul, S. M. E. Haque, and S. Chakraborty, "Human detection in surveillance videos and its applications - a review," *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no. 1, p. 176, Nov. 2013, doi: 10.1186/1687-6180-2013-176.

[2] J. W. Davis, V. Sharma, A. Tyagi, and M. Keck, "Human Detection and Tracking," in *Encyclopedia of Biometrics*, S. Z. Li and A. Jain, Eds., Boston, MA: Springer US, 2009, pp. 708–712. doi: 10.1007/978-0-387-73003-5_35.

[3] G. Boesch, "YOLOv7: A Powerful Object Detection Algorithm (2024 Guide)," viso.ai. Accessed: Apr. 30, 2024. [Online]. Available: https://viso.ai/deep-learning/yolov7-guide/

[4] V. K. Yadav, D. P. Yadav, and D. S. Sharma, "An Efficient Yolov7 and Deep Sort are Used in a Deep Learning Model for Tracking Vehicle and Detection," vol. 18, no. 11, 2022.

[5] K. Liu, Q. Sun, D. Sun, L. Peng, M. Yang, and N. Wang, "Underwater Target Detection Based on Improved YOLOv7," *Journal of Marine Science and Engineering*, vol. 11, no. 3, Art. no. 3, Mar. 2023, doi: 10.3390/jmse11030677.

[6] P. Gao, J. Lu, H. Li, R. Mottaghi, and A. Kembhavi, "Container: Context Aggregation Network." arXiv, Oct. 18, 2021. Accessed: May 07, 2024. [Online]. Available: http://arxiv.org/abs/2106.01401

[7] P. Dollar, M. Singh, and R. Girshick, "Fast and Accurate Model Scaling," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 924–932. doi: 10.1109/CVPR46437.2021.00098.

[8] P. K. A. Vasu, J. Gabriel, J. Zhu, O. Tuzel, and A. Ranjan, "MobileOne: An Improved One millisecond Mobile Backbone." arXiv, Mar. 28, 2023. Accessed: May 07, 2024. [Online]. Available: http://arxiv.org/abs/2206.04040

[9] A. Pujara and M. Bhamare, "DeepSORT: Real Time & Multi-Object Detection and Tracking with YOLO and TensorFlow," in *2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, Nov. 2022, pp. 456–460. doi: 10.1109/ICAISS55157.2022.10011018.

[10] J. Zhang, Y. Yang, D. Gao, R. Wang, and J. Wang, "Deep Learning Based Cross-View Human Detection System," *J. Phys.: Conf. Ser.*, vol. 2504, no. 1, p. 012027, May 2023, doi: 10.1088/1742-6596/2504/1/012027.

[11] "A Trainable System for Object Detection | International Journal of Computer Vision." Accessed: May 08, 2024. [Online]. Available: https://link.springer.com/article/10.1023/A:1008162616689

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010, doi: 10.1109/TPAMI.2009.167.
[13] M. Wójcikowski, "Histogram of Oriented Gradients with Cell Average Brightness for Human Detection," *Metrology and Measurement Systems*, vol. 23, no. 1, pp. 27–36, Mar. 2016, doi: 10.1515/mms-2016-0012.
[14] L. Du, R. Zhang, and X. Wang, "Overview of two-stage object detection algorithms," *J. Phys.: Conf. Ser.*, vol. 1544, no. 1, p. 012033, May 2020, doi: 10.1088/1742-6596/1544/1/012033.
[15] "A survey: object detection methods from CNN to transformer | Multimedia Tools and Applications." Accessed: May 12, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s11042-022-13801-3
[16] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." arXiv, Jul. 06, 2022. Accessed: May 09, 2024. [Online]. Available: http://arxiv.org/abs/2207.02696
[17] M. I. H. Azhar, F. H. K. Zaman, N. Md. Tahir, and H. Hashim, "People Tracking System Using DeepSORT," in *2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Penang, Malaysia: IEEE, Aug. 2020, pp. 137–141. doi: 10.1109/ICCSCE50387.2020.9204956