University of Nottingham
UK | CHINA | MALAYSIA

COMP3065 Computer Vison Coursework

A processing platform for pedestrian surveillance with integrated object detection, tracking, and pose estimation

Guankai Fu
20413130
ssygf1@nottingham.edu.cn

word:2400

code: https://github.com/GuankaiFU1109/comp3065-cw

School of Computer Science

University of Nottingham Ningbo China

1 Introduction

People detection and tracking [1], as a key task in the field of computer vision, has a wide range of application needs in many important fields [2] such as traffic monitoring and public safety. However, traditional people tracking systems often face challenges in practical applications.

The goal of this project is to build an efficient and robust video processing system to enhance intelligence and safety in surveillance applications. The system should not only achieve detection and tracking of people in complex real-world scenarios with high accuracy but also analyze the pose of the people in various aspects. Specifically, the system will provide a comprehensive and useful set of features. These include 1. accurate bounding boxes for recognized objects such as persons and cars, clearly identifying the position of the object in the video frame. 2. accurate track drawing to go to a complete presentation of the movement path. 3.reasonable speed estimation. 4. reliable fall detection to find potential safety risks in a timely manner.5. effective intuitive information including counts and alerts.

2 Methodology

2.1 Object Detection

YOLO (You Only Look Once), with its end-to-end approach, regards the object detection task as a regression problem [3]. It can directly predict the bounding boxes and class probabilities of objects through a single forward propagation, thus greatly improving the detection speed and accuracy.

2.2 Multi Object Tracking (MOT)

MOT aims at continuous and accurate tracking of multiple objects in a video sequence. The Simple Online and Realtime Tracking (SORT) algorithm is widely used for its simplicity and efficiency [4]. SORT consists of two key components, the Kalman filter and the Hungarian algorithm. SORT consists of two key components, the Kalman filter and the Hungarian algorithm. Kalman filtering, as a powerful state estimation method, can accurately predict the future state of a object based on its historical state and observation data. The main task of the Hungarian algorithm [5] is to match the detection results in the current frame with the objects that have been tracked in the previous frames to determine the identity and track of the object. This simple but effective design allows the SORT algorithm to perform well in real-time and meet the needs of most practical applications.

2.3 Fall Detection

Most fall detection methods rely on pose estimation techniques to analyze human pose changes [6]. By detecting and tracking the joint points of the key parts of the human body (e.g., hips, knees, shoulders, etc.), it is possible to obtain pose information of the human body. Then, by setting reasonable fall judgment rules, such as the relative positional relationship between the joint points and the tilt angle of the body, to determine whether a fall event has occurred.

3 System design and implementation

3.1 User Interface (Streamlit)

The user interface of this system is developed using Streamlit [7], which is simple and easy to use, allowing developers to quickly build a functional and interactive UI. In this system, the user interface has the following key functions: firstly, video uploading function, which supports common video formats, such as MP4, AVI, MOV, so it is convenient for users to upload their video files to the system for processing. Second is the real-time preview function, users can view the original video and the video processed by the system in real time on the interface, intuitively understand the processing effect of the system. In addition, the interface also provides interactive control buttons, including the Clear Output button, which allows users to clear the previous processing results by clicking on the button to reopen the operation. Finally, to facilitate the user to save the processed video, the interface also provides a download option, which allows users to download the processed video locally for further viewing and analyzing.For details, please refer to Appendix B

3.2 Object Detection (YOLO)

We chose YOLOv11 [8] as the detection model and optimized the settings. Specifically, we loaded the model in half-precision (FP16) format with eval mode to ensure the accuracy and efficiency of the model in the inference process. After the model was loaded, we also loaded the category labels of the COCO dataset, which enabled the system to accurately recognize different categories of objects such as persons and cars.

Firstly, during pre-processing, YOLO requires the input frames to be resized to 640x640. In the resizing, a suitable scaling algorithm is used to maintain the aspect ratio of the frames and to avoid distortion of the image, and a padding operation is performed on the image if necessary. Secondly, for image tensor dimension and color conversion operations, we need to convert HWC to CHW and BGR to RGB format. Next, we input the frames into the YOLOv11 model for forward propagation, and the model

will output a series of bounding box prediction results, which contain information such as the location, size, and class probability of the object. We adopt the Non-Maximum Suppression (NMS) algorithm to select high-quality detections from these predictions, and the basic idea of the NMS algorithm is to compare the Intersection over Union (IoU) between different bounding boxes and suppress those with high overlap and lower confidence scores ,so as to retain those that are most likely to be the real objects.In this system, we set a low confidence and a high IoU threshold, which can effectively filter out low-quality detection results and improve the accuracy and reliability of detection.

3.3 Multi Object Tracking (SORT)

In the MOT module, we employ Kalman filtering in the SORT algorithm to model and predict the state of the object. In this system, we use a 7-dimensional state vector [x, y, s, r, vx, vy, vs] to represent the state of the object where (x, y) represents the center-of-mass coordinates, s represents the area of the bounding box , r represents the aspect ratio, and (vx, vy, vs) represents the velocity components of the object in the x-axis, y-axis, and area directions. The future state of the object is estimated by the prediction and updating steps of Kalman filtering. The Hungarian algorithm is an efficient combinatorial optimization algorithm. In this system, we measure the similarity between the detection results and the tracking object by calculating the IoU between them, and then we use the similarity matrix as an input to the Hungarian algorithm to obtain the optimal matching results. In this way, we can effectively match the detection results with the tracking objects to ensure that the identity and tracks of the objects are accurately tracked in the video sequences.

3.4 Fall Detection

We use YOLOv11-POSE [9] to detect 17 key joint points of the human body. By analyzing the positions and relative relationships of these joints, we can determine whether the human body's pose is abnormal and thus detect whether a fall has occurred. The fall detection logic is as follows: we first calculate the horizontal and vertical distances between the hip and knee joints. When the horizontal distance exceeds the vertical distance which is shown in the following formula (1), it indicates that the human body may be in a horizontal fall pose. We also considered the tilt of the torso, which may also indicate that a fall has occurred when the angle of tilt of the torso exceeds 45°. In addition, if a fall is detected as a faller, the system will textually indicate falldown in its vicinity to visualize the output directly as an image.

$$|\text{hip}_x - \text{knee}_x| > |\text{hip}_y - \text{knee}_y| \tag{1}$$

3.5 Speed estimation

Speed estimation allows for real-time assessment of an object's motion, providing crucial information for subsequent behavior analysis and safety alerts. In this system, we first calculate the centroid coordinates of the object across two consecutive frames and compute the Euclidean distance between these two centroids. Next, we convert the pixel distances to actual physical distances based on the fps of the video and a pre-calibrated pixel-to-meter scale factor of 1.7, as we use the average height of a Chinese man as a benchmark. Finally, the velocity of the object is obtained by dividing the physical distance by the time interval and the velocity value is displayed as m/s on the video frame.

$$v = \frac{p_d}{h_{\text{avg}}} \cdot H_{\text{real}} \cdot fps \tag{2}$$

Where:

- p_d: the pixel distance between the centroids of the same object across two consecutive frames;
- h_{avg}: the average height (in pixels) of all human detection boxes in the current frame;
- H_{real}: the average real-world height of a human (in meters), set to 1.7 in this project;
- fps: the frame rate of the video (frames per second).

4 Results

We evaluated the system on several videos representing different scenarios: (1) a mixed traffic, pedestrian crowded scene was used to test multi objects detection and tracking; (2) a road video focusing on car movement and a video of person movement to evaluate the speed estimation functionality (3) a video of a simulated fall was used to validate the fall detection capability. The remaining results are presented in Appendix A.

For object detection and track, the YOLO detector successfully identified most persons and cars in each frame. However, false detections also occurred, as in Figure 3, where the dummy model in the window was also detected as a person. The SORT tracker was able to maintain IDs for moving persons and cars with a high degree of consistency. However, other results suggest that when two pedestrians in proximity are merged or one of them is missed resulting in overlapping detections, their IDs may be swapped. However, the actual impact is minimal, and the fact that the IDs are swapped and then tracking continues effectively means that the identities of both individuals are lost within a second.

For fall detection, as in Fig. 4 for a standard fall, we can quickly and accurately detect the falldown state. However, as in Fig. 1, the pedestrian is riding an electric bicycle and wearing a raincoat, it is difficult for us to analyze the location of the key point through pose estimation, and misjudgment will occur through aspect ratio.

For speed estimation, since we do not have the verification of physical devices, we cannot get the accurate speed just by prediction, but through qualitative analysis, it is reasonable that the speed of the person is about 1m/ is reasonable. One limitation we noticed is that since the camera is moving, all the backgrounds such as trees move in the frame such that any misdetection of the background occurs leading to confusing speeds.

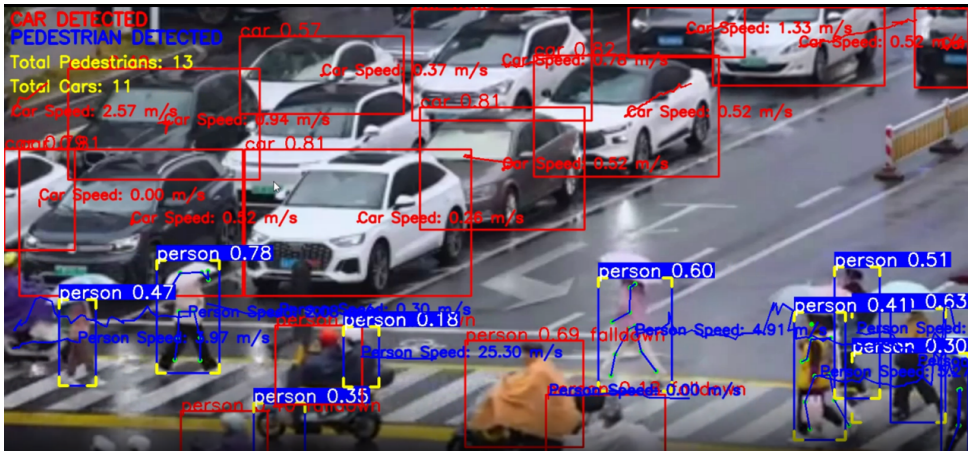


Figure 1: a mixed traffic, pedestrian crowded scen

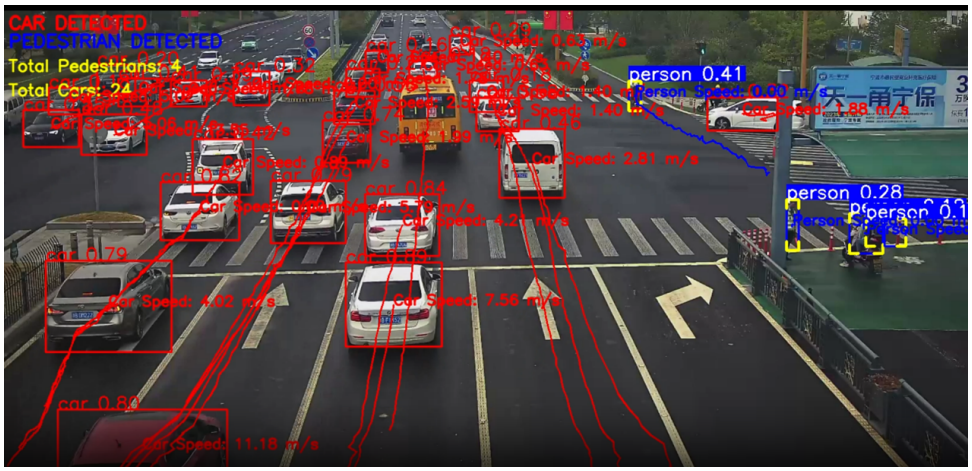


Figure 2: a road video focusing on car movement

Figure 3: a video of person movement

Figure 4: a video of a simulated fall

5 Discussion

In this section, we discuss the advantages and disadvantages of the algorithms in this system as well as the improvement points.

UI usability and performance: using Streamlit is an advantage in terms of development speed and ease of creating controls and layouts. The interface is user-friendly and simple to use, allowing users to accomplish tasks with the click of a button.

Detection accuracy and responsiveness: the use of the YOLO model provides robust real-time detection capabilities [8]. A major advantage is that YOLO can detect multiple object classes in a single detection at high speed, and the pre-trained model showed good generalization on our test videos. The confidence threshold setting and NMS ensure that we get mostly correct bounding boxes. However, one drawback is that small or fast moving objects are occasionally missed and for two pedestrians with a high overlap, the detection frame is momentarily lost. Another is the failure to make correct judgments about dummy models in windows and multiple vehicles that appear in complex realities, such as motorized bicycles. These can be improved in terms of accuracy through enhanced training or fine-tuning for specific scenarios. Overall, the system demonstrates high accuracy and speed for different categories of detection capabilities.

Tracking robustness: the SORT tracker handles the data correlation problem well across multiple classifications [10]. Due to the use of IoU matching strategy, it rarely confuses objects from different classes. The simplicity of the Kalman filter and the Hungarian algorithm approach allows the system to achieve real-time performance with minimal overhead. In addition, the predictive power of the Kalman filter helps to compensate for transient missing detections due to occlusions.

Reliability of fall detection: we start with a simple prediction by detecting the aspect ratio of the frame and then determine whether it is a falldown state or not based on abnormal changes in the key points. However, in the real world, human fall pose may be caused by a variety of reasons, and we have to introduce a variety of factors to make a real judgment. For example, disabled people, children's body pose is different from normal people, we need to strengthen the reliability through multiple algorithms and training.

Speed estimation and correction: an obvious weakness is that without knowing the actual pixel-to-meter ratio or the camera's angle of view, we just make a speed estimation. We use the average human height of 1.7m as a baseline, which is unreliable to get the true speed in real different human heights. In the real world, we can incorporate physical devices such as high-speed cameras for calibration training [11], including combining motion and perspective to improve the accuracy of the system.

6 Conclusion

In this project, a video analytics system integrating object detection, multi-target tracking, human pose estimation, fall detection and speed estimation is constructed. By introducing the YOLOv11 detector and the SORT tracker, the system achieves high precision object detection and tracking in complex scenarios, while combining with pose estimation to achieve effective fall behavior recognition. In addition, the system performs speed estimation, which provides a practical reference for intelligent monitoring and risk warning. The user interface realizes the functions through the Streamlit platform with simple operation and intuitive interaction.

In future, we should optimize the model accuracy and improve the detection robustness for small targets, occluded targets and complex backgrounds. Through model pruning, improving network structure and other lightweighting techniques [12], the system will have stronger real-time processing capability and adapt to edge device deployment. On basis of fall detection, LSTM [13] is introduced to further train the system in more complex human behaviors such as abnormal behavior recognition to enhance reliability.

References

[1] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *2008 IEEE Conference on computer vision and pattern recognition*. IEEE, 2008, pp. 1–8.

[2] A. H. Ahamad, N. Zaini, and M. F. A. Latip, "Person detection for social distancing and safety violation alert based on segmented roi," in *2020 10th IEEE international conference on control system, computing and engineering (ICCSCE)*. IEEE, 2020, pp. 113–118.

[3] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia computer science*, vol. 199, pp. 1066–1073, 2022.

[4] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-centric sort: Rethinking sort for robust multi-object tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 9686–9696.

[5] B. Sahbani and W. Adiprawita, "Kalman filter and iterative-hungarian algorithm implementation for low complexity point tracking as part of fast multiple object tracking system," in *2016 6th international conference on system engineering and technology (ICSET)*. IEEE, 2016, pp. 109–115.

[6] X. Wang, J. Ellul, and G. Azzopardi, "Elderly fall detection systems: A literature survey," *Frontiers in Robotics and AI*, vol. 7, p. 71, 2020.

[7] M. Khorasani, M. Abdou, and J. H. Fernández, "Web application development with streamlit," *Software Development*, pp. 498–507, 2022.

[8] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725*, 2024.

[9] D. Maji, S. Nagori, M. Mathew, and D. Poddar, "Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 2637–2646.

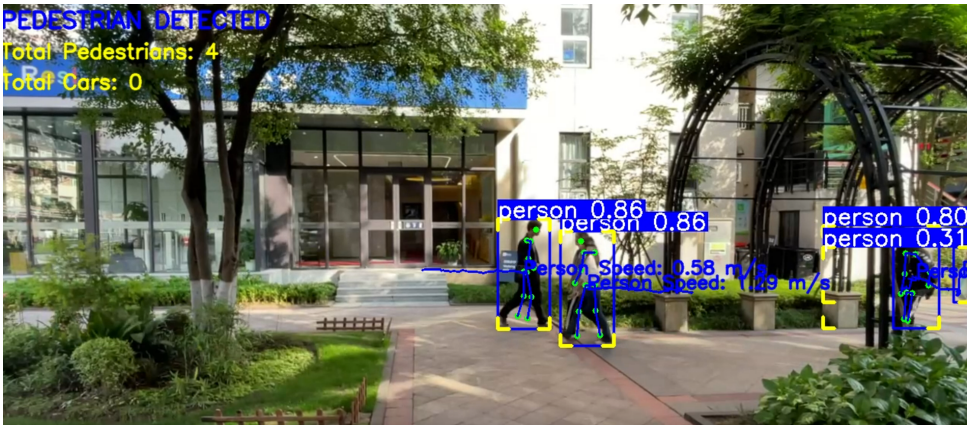
[10] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, "Strongsort: Make deepsort great again," *IEEE Transactions on Multimedia*, vol. 25, pp. 8725–8737, 2023.

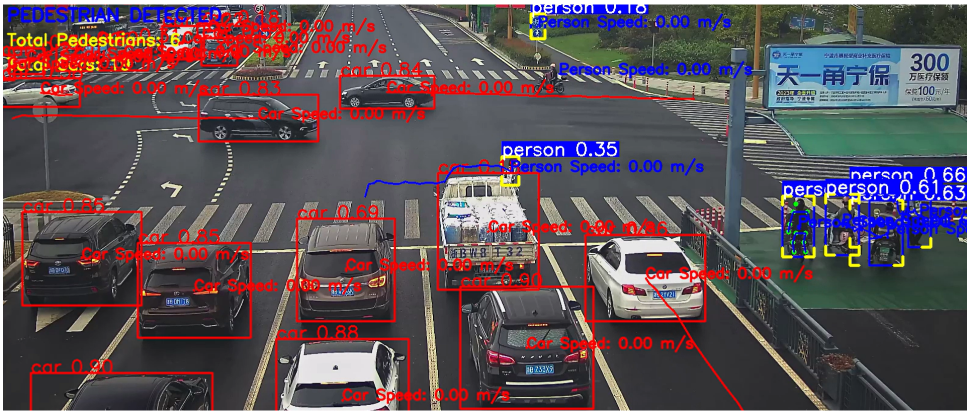
[11] D. Fernández Llorca, A. Hernández Martínez, and I. García Daza, "Vision-based vehicle speed estimation: A survey," *IET Intelligent Transport Systems*, vol. 15, no. 8, pp. 987–1005, 2021.

[12] C.-H. Wang, K.-Y. Huang, Y. Yao, J.-C. Chen, H.-H. Shuai, and W.-H. Cheng, "Lightweight deep learning: An overview," *IEEE consumer electronics magazine*, vol. 13, no. 4, pp. 51–64, 2022.

[13] Y. Luo, J. Ren, Z. Wang, W. Sun, J. Pan, J. Liu, J. Pang, and L. Lin, "Lstm pose machines," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5207–5215.

Appendix A: Additional Results




PEDESTRIAN DETECTED
Total Pedestrians: 6
person 0.18
Person Speed: 0.00 m/s
person 0.35
car 0.84
car 0.83
car 0.85
car 0.88
car 0.90
Car Speed: 0.00 m/s
person 0.66
person 0.61
天一甬宁保
300
万医疗保额

Appendix B: UI design

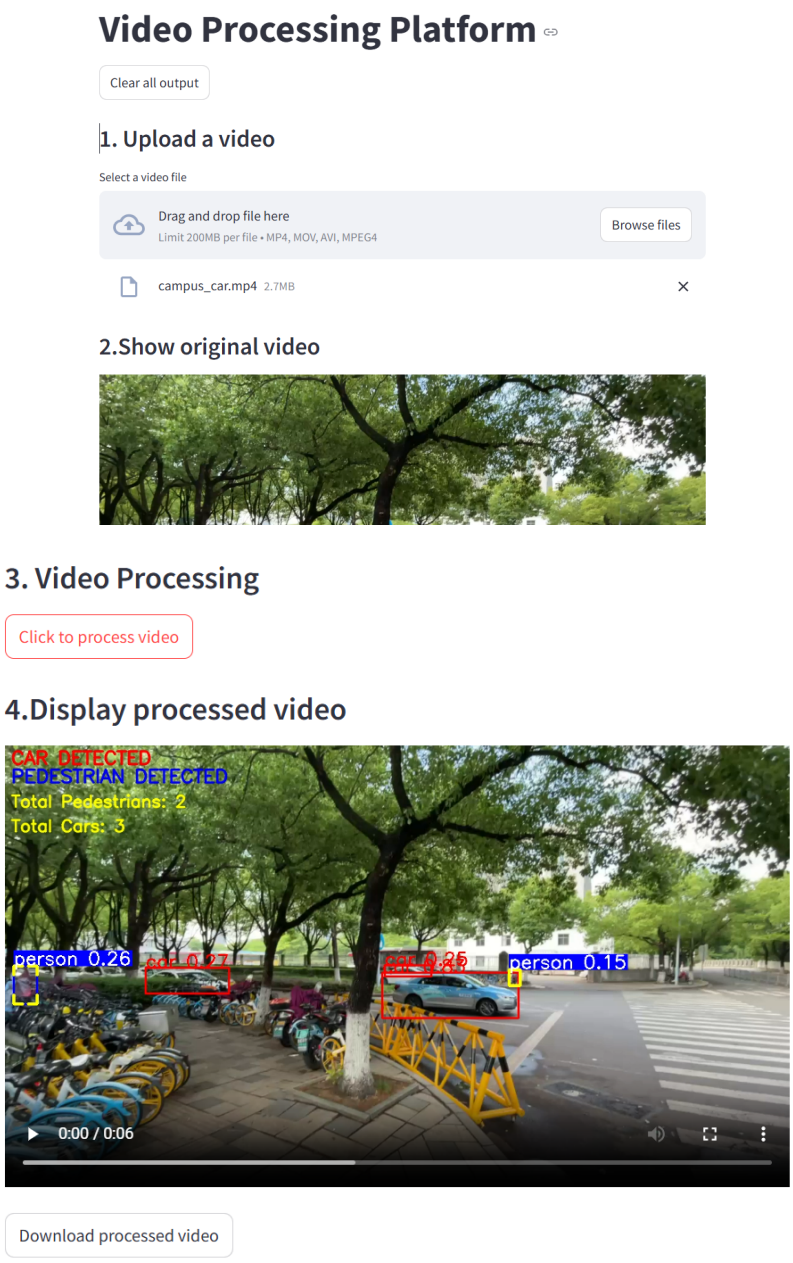


Figure 5: User Interface