Pedestrian Tracking and Analyzing

*Group Name: Five knights

Yueying Li
University of New South Wales
z5212833

Kaiying Wu
University of New South Wales
z5251751

Yi Ye
University of New South Wales
z5197130

Qiushi Li
University of New South Wales
z5376373

Jiatao Li
University of New South Wales
z5223756

I. Introduction

Object tracking in real time videos or continues images sequence is a high active area in computer vision. Related applications can be used in traffic monitoring, autonomous driving and social distance rules enforcing. The goal of this group project is to develop and evaluate a method for multiple pedestrians tracking in real-time videos or processed image sequence, and apply motion analysis on a public dataset.

The whole project can be described in three steps:
task1: pedestrians tracking and pedestrians detection by linking the boxes from frame to frame for each pedestrian, and motion analysis from the trajectories individually.
task2: pedestrians counting: total count of pedestrians detected since the start of the video, in the present video frame, and within some manually drawn rectangular region. Numbers should be present the result in the terminal or better on the video frame.
task3: analyze pedestrians: count of groups with two pedestrians or more as well as single walkers, occurrences and destruction of groups(any group member leaves), highlighting occurrences of pedestrians entering or leaving the scene.

The difficulties of this project can be listed as follows. Firstly, detection precision which could be influenced by illumination changes and disturbing objects such as trees and buildings. Moreover, scale and shape changes also make the pedestrian tracking difficult. From Fig 1 it can be seen that some pedestrians would touch and occlude each other which is also the main challenge of this project.

To accomplish the tasks, firstly we pre-process the images sequences by reducing image noise by using Gaussian Blur with kernel size 9*9, and detect pedestrians distinguished from background by using histogram of oriented gradient (HOG) + linear support vector machines (SVM) and then draw the bounding boxes for each pedestrian with different colours. Then we extend the program for task two, where we can count the total number of unique pedestrians by achieving the total number of boxes that we obtained from earlier. Finally, measurement IoU between bounding box can be used to decide whether pedestrians can be considered as a group.

Fig. 1: Step Image Example

Discussions about the experiment results including parameters like min-distance and padding size of Gaussian-Blur and more has lead to conclusions that the efficiency would be affected by many factors, and improvement according to these can be made in the future.
The dataset we used can be found at: https://motchallenge.net/data/STEP-ICCV21/ , the Segmenting and Tracking Every Pixel (STEP) benchmark consists of 2 training videos from the Training Set to train our method, and 2 test videos from the Testing Set to demonstrate on the first video.

II. Literature Review

The main problem of visual tracking, generating an inference about the motion of an object given a sequence of images, is that the wide variety of aspects in tracking circumstances and tracking methods makes it hard to be reconciled in one algorithm. Also, tracking difficulty primarily increases with number of objects, occlusions and tracking length.

In paper [5], aiming for assessing the accuracy and robustness of single object trackers, it selected nineteen trackers to conclude a wide variety of algorithms often cited in literature in recent years and divided them into five groups

according to the method used for tracking:
matching: NCC, KLT, KAT, FRT, MST, LOT; matching with extended appearance model: IVT, TAG, TST; matching with constraints: TMC, ACT, L1T, L10; discriminative classification: FBT, HBT, SPT, MIT, TLD; discriminative classification with constraints: STR;
and took F-score as effective as the object tracking accuracy(OTA) score in the evaluation practice.

Paper [9] also summarizes the achievements and problems of existing algorithms for single-object tracking and reveals that the combination of convolution features and correlation filters greatly improves the performance of trackers, since deep learning improves the features and models of trackers.

The survey in paper[1] also compares tracking methods: subspace matching, constrained optimization, tracking-by-detection and the structured classifier provided by STR and many other new methods.

The data collected for the experiments here includes 315 standard video sequences mainly from YouTube: with most of the sequences are short with an average length of them is 9.2 seconds and a maximum of 35 seconds and nearly 90,000 individual farms are used totally. The current knowledges shows that the best choice of a dataset would be many short videos supplemented with a new very one videos, with a single defined target. Such that in paper [6], it introduced a new benchmark, encompassing two datasets KITTI-STEP and MOTChallenge-STEP(also commonly used for multiple object tracking) [1] that contain long video sequences and challenging examples for studying long-term pixel-precise segmentation and tracking under real-world conditions, then further propose a novel evaluation metric Segmentation and Tracking Quality(STQ).

We can obtain from this survey's results after evaluation that STR, FBT, TST, TLD and L1O performs best in that order and remarkably originate from each of the five groups. Overall, STR performs the best and has an even performance on all aspects of difficulty which is attributed to the use of S-SVM, while only fails on the scale changes of the target. Furthermore, TLD performs remarkably well on camera motion, and FBT on specializing in appearance changes of the target and changes in illumination. Generally, trackers may specialize on one certain circumstance and fail on others.

The paper also states that the locality is a trend that has influenced tracking, due to the object decomposed into segments, super-pixels, parts, patches and structures. It is also acceptable to combine local and global information together to address rapid and significant appearance changes.

In paper [2], trackers are categorized into Correlation Filter based Trackers (CFTs) and Non-CFTs and then further classified into various types based on the architecture and the tracking mechanism. It concludes that Discriminative Correlation Filter (DCF) based trackers perform better than the others, and inclusion of different types of regularizations over DCF often results in boosted tracking performance.

Paper [4] also investigate the current DL-based visual tracking methods, benchmark datasets, and evaluation metrics. It summarized the fundamental characteristics, primary motivations, and contributions of DL-based methods from nine key aspects: network architecture, network exploitation, network training for visual tracking, network objective, network output, exploitation of correlation filter advantages, aerial-view tracking, long-term tracking, and online tracking. Having comprehensively examined on a set of well-established benchmarks of OTB2013, OTB2015, VOT2018, LaSOT, UA V123, UA VDT, and VisDrone2019, and finally summarizes pros and cons of these state-of-the-art DL-based trackers.

In the light of Deep Visual Tracking, surveys like paper [3] concludes that:
The convolutional neural network (CNN) model could significantly improve the tracking performance by distinguish the object from its surrounding background and could get more accurate results, while using the CNN model for template matching is usually faster.

The trackers with deep features perform much better than others, especially the ones using end-to-end networks. The effective combination of deep features usually results in a more robust tracker.

The most suitable network training method for visual tracking is to per-train networks with video information and online fine-tune them with subsequent observations.

III. Methods

A. Preprocessing

Our group pre-process images by using Gaussian Blur with kernel size 9*9 to reduce image noise. [7] Gaussian blur is a type of image-blurring filter that uses a Gaussian function for calculating the transformation to apply to each pixel in the image. Below is the formula for Gaussian function in two dimension

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and sigma is the standard deviation of the Gaussian distribution. Values from this distribution are used to build a convolution matrix which is applied to the original input image. Below is an example of the effect of Gaussian blur. Fig 2 shows that some non-pedestrian object was detected to be pedestrain and bounded as boxes such as the 'woolworth' logo, red bounding box before gaussian blur applied.

B. Task1

1) 1.1: Our group use histogram of oriented gradient (HOG) and linear support vector machines (SVM) to detect pedestrians. The histogram of oriented gradients (HOG) is a feature descriptor for the purpose of object detection used in computer vision and image processing area. It counts occurrences of gradient orientation in localized portions of an image. This method is similar to edge orientation histograms, SIFT, and shape contexts, but differs in that it is computed on

(a) before gaussian blur (b) after gaussian blur

Fig. 2: Gaussian blur effect

a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.

SVM: Default people detector is used to return coefficients of the classifier trained for people detection and these coefficients are set for the linear SVM classifier we initialized. DetectMultiScale function detects pedestrians of different sizes in the input image frame. The detected pedestrians are returned as a list of rectangles and we use different colors to draw them in the frames.

2) 1.2: For pedestrian tracking, method we used to obtain trajectory is to calculate distances between boxes in two successive frames. For each bounding box in current frame, find the closest bounding box in previous frame which is not larger than threshold. The selection of this threshold is discussed in discussion. This bounding box is defined as nearest neighbour nn. Thus nn and current box belong to same pedestrian. The start point of pedestrian is defined as the left bottom corner coordinates. The displacement is defined as trajectory between time stamp t-1 and t and distance is defined as

$$Distance = \frac{1}{2} * ((x1 - x2)^2 + (y1 - y2)^2) \qquad (2)$$

where x and y are the left corner coordinates of current box and its nearest neighbour. A global line list is created to store all trajectories of each pedestrian. For each frame, all previous lines is drawn and new displacement is appended and drawn too. Thus, the bounding boxes are linked over time and the trajectory for each pedestrian is obtained from the time when the pedestrian first appeared up to the current time point.

3) 1.3: For each bounding box, if a nearest neighbour is found, the color of this bounding box as well as the trajectory is set to be his nearest neighbour's color.

C. Task2

1) 2.1: In order to count the total number of pedestrians since the start of video, a dictionary is created to store all the information of pedestrians including color and id.If the nearest neighbour of this pedestrian in last frame is found, total number does not increase and the color of this pedestrian is the color of his nearest neighbour's color. If nearest neighbour is zero, total count was increased and a new dict with new color, id and other information is assigned for him. The number of total count is reported in the top left corner of current frame.

2) 2.2: For the number of pedestrians in current frame, the number of bounding boxes is counted and result is put on top left corner.

3) 2.3: To draw a selection area with mouse, the mouse movement was followed and bind by a tkinter label, the coordinate of motion is returned and updated with x and y accordingly by event listener. When the left mouse button is released, the selected rectangle would be shown in white by clicking the next image button.

Fig. 3: Selected region example

4) 2.4: From fighure 2 it can be seen that the number of pedestrians in current select region is 4. For determine whether pedestrians were in the selected area, we check the x,y,w,h coordinates of each pedestrian. if the pedestrian is inside the selected area or the (IoU) score is over 0.075, this pedestrian is counted to be in this area. The number returned is put in left bottom corner

D. Task3

1) 3.1: The method of detecting the pedestrian belonging to a group or walking alone is to compute the Intersection over Union (IoU) score.

$$IoU = \frac{Area\,of\,Overlap}{Area\,of\,Union} \qquad (3)$$

After that, we tested serial different IoU scores to find the best-matched scores for our model, which is 0.13. Because of this, we define pedestrians as in a group when their IoU score is larger than 0.13 and count them as alone when the IoU score is lower than the value we set. After that, we will update their in-group state in the dictionary to true or false based on the result we computed.

2) 3.2: To find out whether the pedestrian is joining or leaving the group, we will use their unique colour to check their previous frame in-group status that we mentioned in Task

3.1. If their last in-group status is true and their current in-group status is false, we will count this pedestrian as leaving the group. On the other hand, if their previous in-group status is false and now their in-group status is true, we will count them as joining the group. In addition, if the pedestrian is new and detected as in the group, we will also count this new pedestrian as joining the group.

3) 3.3: The number of pedestrians entering is simply counted by the number of pedestrians that do not have nearest neighbours in previous frame, while the number of pedestrians leaving is counted by checking the disappearing colors from last frame. For example, if the colors use in previous frame are red, blue and yellow but there is no yellow bounding box in current frame, the number of pedestrians leaving was increased by 1. Besides, if the pedestrian is leaving, the trajectories of that pedestrian is removed by loop through current line list, if the color of line still exists, that line is drawn, otherwise it would not be shown. The information of pedestrian leaving or entering is shown in the left bottom corner in red.

IV. Experimental Results

A. Experimental Setup

Datasets: STEP-ICCV21-09, STEP-ICCV21-02, STEP-ICCV21-07, STEP-ICCV21-01

Running Equipment: Macbook Pro with M1 8GB RAM

Environment: Python 3.9.1

Python Packages: OpenCV 4.5.5, TKinter, PIL, Numpy 1.22.3

B. Evaluation Metrics

1) Object detection:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y} - Y)^2 \tag{4}$$

where n means total number of step images and $\hat{Y}$ means number of bounding boxes detected in original images and Y indicates the number of boxes in labelled image. So that the performance is better if MSE is smaller.

2) Precision and Recall: Second evaluation method we used is to calculate Precision and Recall for 50 random selected images in each video and manually count the pedestrians within current frame. The definition of these two functions are defined as follows

$$Precision = \frac{tp}{tp + fp} \tag{5}$$

$$Recall = \frac{tp}{tp + fn} \tag{6}$$

tp is true positive which indicates the pedestrians we correctly predicted, fp is false positive which counts the number of objects we predict to be pedestrian but not actually a pedestrian. As shown in Fig 2, the bounded woolworths logo is counted as false positives. According to [8], precision is the fraction of relevant instances among the retrieved instances and recall is the fraction of relevant instances that were retrieved. So if some pedestrians shown in video frame are not detected, the recall rate would be small. And if the number of wrong objects we bounded increase, the precision would be low. Our target is to get higher precision and recall rate in every training and testing video.

TABLE I: Precision and Recall

Table	***train02***	***train09***	***test07***	***test01***
precision	0.84	0.92	0.83	0.75
recall	0.85	0.86	0.75	0.76
f1 score	0.85	0.90	0.89	0.75

From Table 1 it can be seen that the result for test set01 is not as good as others which is because the illumination condition. As shown in Fig 4, by comparing with Fig 2 and Fig 3, it can be seen that the illuminations are relatively dark which makes it more difficult to get a higher f1-score.

Besides, for results in table two, if the total number of pedestrian is larger than the actual number of pedestrian, which means one same pedestrian could be detected as new pedestrian enters in two successive frame. So that the length of total trajectory drawn should be shorter. Our target is to make the difference between ground truth and number we detected becomes smaller. Since we do not have ground truth for test set, only evaluation method two is available now.

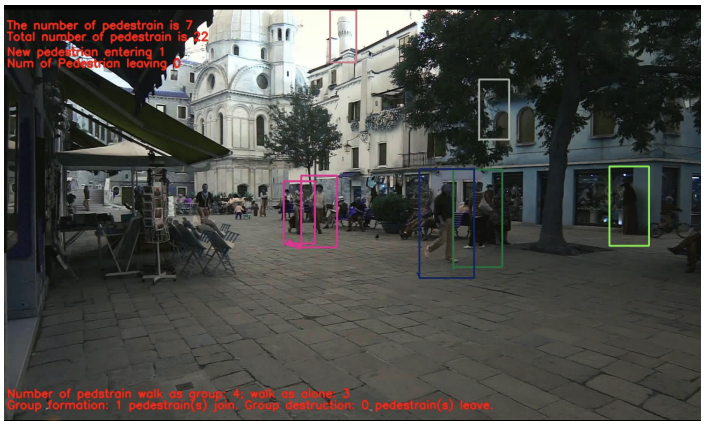


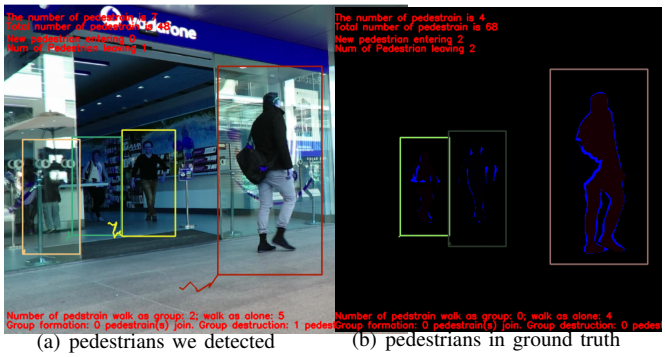
Fig. 4: Test 01 Example

C. Evaluation Example

TABLE II: MSE and total number

Table	***MSE***	***Total Num of Bounding box***	Actual Num
train02	4.846	492	652
train09	5.836	477	459

It can be seen that the number of bounding boxes get in Fig 4(a) is 4 and number in Fig4(b) is 3 so that the difference is 1. Summarize the squared difference in each frame and divide it by n we can get the MSE.



(a) pedestrians we detected (b) pedestrians in ground truth

Fig. 5: Evaluation Example

V. Discussion

Based on the experimental results provided in the section IV, the methods we implemented in this project can complete these three tasks and achieve the expected effects on tracking pedestrians in the videos, counting numbers over time and analyzing the groups with a satisfied performance and accuracy.

Precision is analyzed for our predictions, which shows how many of the tracked pedestrians are actually positive samples. Recall is used for our original dataset which indicates how many tracked pedestrians in the dataset are predicted correctly. High Precision can be regarded as a more cautious model. Although it may be not able to capture specific entities, as long as the object is captured, the result is almost correct and the situation of misjudgments can be avoided. In terms of high Recall model. It sometimes catches the wrong one while it covers almost everything that should be caught.

In general, there is a balance between these values, and it is often necessary to make trade-offs according to specific situation. For the aims of searching, improve the precision rate as much as possible while ensuring the recall rate, that is to reduce the rate of wrong objects such as trees and buildings. For large-scale detection, the recall rate should be improved under the condition of ensuring the precision rate. Reduce the situation of missed detection. As the result, in most cases we need to comprehensively weigh the meaning of Recall and Precision. The F-score can be used as the harmonic value of them.

When we think that both Precision and Recall will have a large impact on the results for analyzing, the β =1, and it is called F1-score. In other cases, if we need to focus on the Precision, the value of β should be set less than 1. On the contrary, the β larger than 1 means the Recall is more important for the final result.

A. Parameters and Kernel Size

During the process of developing, we analyzed result and keep adjusting parameters to improve the accuracy of our methods. In terms of min distance in Find nearest function, a suitable value can avoid the error in judging the number of people gathered. If the value is too large, two pedestrians walking together will be detected as one. And if the value is too small, the pedestrian who has been detected will be judged as new one which leads to a increase of total number of pedestrians. Our min distance is set to be 50 in the end.

A feasible size of kernel makes contribution to box the correct objective. If the parameters blur1 and blur2 are smaller, the pictures with pedestrians are blurred less. Besides, excessive values of blur1 and blur2 not only filter out noise, but also remove useful information in the image. In terms of winstride in HOG, it is the step size determined for the detection of window movement. By adjusting the value of this parameter, we not only improve the detection accuracy, but also reduce the detection time. And compute the intersection over union between bounding boxes can help to improve the judgment of grouping and selecting pedestrians.

Finally, there are some potential problems that make negative influence on the result. In the test set 1, the Light will affect the judgment of the number of people. And the same person is repeatedly detected as unique on because of the surrounding movements makes negative effects on the total number.

VI. Conclusion

In this project, we mainly focused and solved several tasks of multiple pedestrians tracking in real-time videos or processed image sequence, and analyze their motion.

The data-set we used is a public data-set and it is new, which adds much difficulties on getting familiar with data-sets and substitute the data into project.

We used traditional approaches methods such as using Gaussian Blur to reduce image noise, using Hog and SVM to detect pedestrians and using measurement loU to decide whether pedestrians can be considered as a group to accomplish the tasks rather than deep learning.

The main problem of task is showing the motion of an object in a sequence of images. We review litterateurs and used some methods to solve this problem. First of all, we use data prepossessing methods to reduce image noise, and then HOG can help model for object detection and feature descriptor. It uses overlapping local contrast normalization for improving accuracy. SVM is used for people detection and return as a list for future drawing in the frames.

For pedestrians tracking, we calculated the distance between two frames in successive frames. and if the distance is beyond threshold, we just deny these two are belong to same people.

And we can draw rectangular in the sequential pictures by checking previous frame to determine whether show a rectangular in the next picture by Open-CV.

The results of our project can reach a high level which is around 75% to 90% in different test data-sets. But more performance measurements should be implemented in the future.

But there are still issues needs to be amended and some future works need to be implemented: 1. we need to minimize the time complexity of some certain functions because of three

loops. 2. we didn't try deep learning algorithms such as yo-lo v5 or CNN. The yo-lo v5 is a superstar role in item detection and recognition, which is also small but fast. 3. The rectangular can not be drew or shown in videos. 4.Some pedestrians coming form right side cannot be detected due to the weak light and low frame rate. And the label pictures are basically not used in our project, which can improve some performance and accuracy to some extent.

REFERENCES

[1] G. Ciaparrone, F. Luque Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera. Deep learning in video multi-object tracking: A survey. *Neurocomputing*, 381:61–88, 2020.

[2] M. Fiaz, A. Mahmood, S. Javed, and S. K. Jung. Handcrafted and deep trackers: Recent visual object tracking approaches and trends. *ACM Comput. Surv.*, 52(2), apr 2019.

[3] P. Li, D. Wang, L. Wang, and H. Lu. Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, 76:323–338, 2018.

[4] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, and S. Kasaei. Deep learning for visual tracking: A comprehensive survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(5):3943–3968, 2022.

[5] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1442–1468, 2014.

[6] M. Weber, J. Xie, M. Collins, Y. Zhu, P. Voigtlaender, H. Adam, B. Green, A. Geiger, B. Leibe, D. Cremers, A. Ošep, L. Leal-Taixé, and L.-C. Chen. Step: Segmenting and tracking every pixel, 2021.

[7] Wikipedia contributors. Gaussian blur — Wikipedia, the free encyclopedia, 2022. [Online; accessed 4-August-2022].

[8] Wikipedia contributors. Precision and recall — Wikipedia, the free encyclopedia, 2022. [Online; accessed 4-August-2022].

[9] Y. Zhang, T. Wang, K. Liu, B. Zhang, and L. Chen. Recent advances of single-object tracking methods: A brief survey. *Neurocomputing*, 455:1–11, 2021.