Pedestrian Detection and Tracking

Boyun Huang
Computer Science and Engineering
The University of New South Wales
Sydney, Australia
z5342276@ad.unsw.edu.au

Ruijie Ge
Computer Science and Engineering
The University of New South Wales
Sydney, Australia
z5277454@ad.unsw.edu.au

Lorac Zhu
Computer Science and Engineering
The University of New South Wales
Sydney, Australia
z5277143@ad.unsw.edu.au

Shu Yang
Computer Science and Engineering
The University of New South Wales
Sydney, Australia
z5285542@ad.unsw.edu.au

Zequan Ding
Computer Science and Engineering
The University of New South Wales
Sydney, Australia
z5327869@ad.unsw.edu.au

I. INTRODUCTION

A. Main task

This project is mainly asking to complete a series of tasks about the pedestrian in the video, such as detecting pedestrians, tracking and mapping the pedestrian path, counting pedestrian number (including the total number of pedestrians throughout the video, the number of pedestrians every frame, and the number of pedestrians in a rectangular frame of any size) , judging whether the pedestrian walk together and showing the pedestrians in and out of the picture and so on.

When target tracking is applied in practice, the detection results will be affected by many factors, such as the change of lighting in different application scenarios, the complexity of the state change of the target object, and the diversity of the target object morphology. Therefore, pedestrian tracking is still of great significance in the field of computer vision.

B. Dataset

1) Analysis of datasets: In the process of preprocessing the dataset of this project, we found that: the images in the training set were all taken by camera at fixed locations; Objects other than pedestrians remain relatively still; And the overall light change range of the picture was small. Using the frame difference method to detect moving objects in this type of data set has good performance, but for for those pedestrians moving slowly or even stationary, it can easily lead to the same object being almost completely overlapped in the two frames before and after, so that it cannot be detected. Since HOG(Histogram of Oriented Gradients) operates on the local square cell of the image, it can maintain good invariance to both geometric and optical deformations of the image. In addition, OpenCV has a trained HOG pedestrian detection model, which obtains the feature descriptor of pedestrian by calling Cv2.hogdescriptor_getDefaultPeopledector (), and sets SVM classifier in HOG to classify the image. According to the obtained results, the combination of HOG+SVM can present good results, so this project mainly adopts HOG+SVM for pedestrian detection.

2) The challenges: The face of the target (pedestrian) tracked in the dataset STEP-ICCV21-02 is not clear enough, and there are many occlusions (trees and benches in the middle of the road) in the training set, which will cause interference to the recognition of moving pedestrians. Since HOG uses sliding Windows to extract features, after classification and recognition by classifiers, many Windows will be included or mostly crossed with other Windows. The NMS(Non Maximum Suppression) adopted in this project removes redundant target boxes. However, when the two characters are too close to each other or overlap, the optimization will label the two people as one person. And because the call is the built-in default parameters of OpenCV pedestrians, although the overall effect in pedestrian detection is good, there are still some people can not be identified.

II. LITERATURE REVIEW

A. Common methods of object detection at present

Traditional object detection method: region selection (exhaustive strategy: sliding window is adopted to design different window sizes and length-width ratio for image traversal); Feature extraction (SIFT, HOG, etc.; Morphological diversity, light variation diversity and background diversity make the features less robust); Classifiers are used for classification (SVM, Adaboost, etc.).

By extracting candidate regions, the corresponding regions are classified mainly by deep learning methods (R-CNN, SPP-NET, FAST R-CNN, Faster R-CNN, etc.). Modern approaches to human perception are largely based on DL(Deep Learning). Many studies have proven that DL is an effective approach to object detection and segmentation, image classification and natural language processing.[1] DL models [2] contain classes that can learn the feature hierarchy by creating high-level feature from low-level features. In this way, DL architectures represent the data better [3].

Based on DL and other regression methods (YOLO/SSD/Dense Box, combined with RCC detection such as RNN algorithm, combined with Deformable CNN of DPM). At present, the main challenges faced by this project include: multiple redundant calibration boxes in feature extraction, high algorithm time complexity and insufficient code running speed, etc.

B. Previous methods for this project

1) Traditional Object Detection Method (Viola-Jones Classification Algorithm): Haar classifier in OpenCV = Haar-like feature + integral graph method + AdaBoost + cascade. Haar-like rectangle features have been used for detecting humans [4] and faces [5]. Using Haar-like input features: Thresholding the sum or difference of rectangular image regions (can pass the image region containing objects, while allowing some images without objects to pass); The integral image technique accelerates the calculation of the 45 degree rotation value of the rectangular image region. The

Haar-like rectangle features encode the intensity contrast between neighboring regions. Such features are suitable for face detection [6]. This image structure is used to accelerate the calculation of Haar-like input features (also with a low rejection rate). Adaboost is used to create classifier nodes (high pass rate, low rejection rate) for the binary classification problem (face and non-face). The classifier nodes are grouped into filtering cascades.

2) R-CNN: Input an image, search selectively from the image to generate 2000 or so area suggestion boxes. For each area suggestion box, CNN is used to extract features and combine features. A linear SVM classifier is used to classify each area suggestion box. Redefine the target bounding box by bounding box regression algorithm.

Limitations: The overlap of target candidate regions makes CNN feature extraction very redundant.

3) YOLO detection sysetems: The whole image is divided into S*S grid; Then send the whole image to CNN to predict whether there is a target in each grid. If so, predict the bounding box and category of the target. Perform NMS (Non Maximum Suppression) on the predicted bounding box to obtain the final result.

Advantages：

1. The detection speed reaches real-time frame detection, which can detect 45 frames per second on Titan X GPU and 150 frames per second on the fast version.
2.YOLO is a brand-new detection method, which detects the position and category of the target by means of regression on the whole picture.

Disadvantages：

1. Poor positioning accuracy, which is not as high as the region proposal algorithm, is mainly since that the algorithm does image regression instead of sliding window detection.
2. The detection effect is not good for small targets or targets close to each other.

C. Development of object detection algorithms

In the evolution of the target detection algorithm, 2012 is an important dividing line. Before 2012, the target detection algorithm more focused on traditional algorithms, while after 2012, the target detection algorithm evolved into the deep neural network target detection and recognition algorithm.

For the traditional target detection algorithm, Viola and Jones [7] first tried to use Haar-like wavelet features and integral graph method for face detection based on the AdaBoost algorithm in 2001 and designed a cascade of more effective features for face detection and strong classifiers trained by AdaBoost. The innovation of this algorithm is to adopt integral image technology to accelerate the calculation of Haar-like input features, and adopt detection cascade technology to improve the accuracy, allowing the background area of the image to be discarded quickly, to put more calculations on the area that may be the target, reducing the computational overhead. Therefore, this algorithm proposed at that time was called Viola-Jones detector. After that, Rainer Lienhart and Jochen Maydt [8] extended this detector with diagonal features and finally formed OpenCV's current Haar classifier.

In 2005, HOG was proposed by Dr. Dadal [9][10] in the CVPR paper to solve the problem of pedestrian recognition. As Dr. Dadal said in the paper, for human detection, rather coarse spatial sampling, fine orientation sampling and strong local photometric normalization turns out to be the best strategy, presumably because it permits limbs and body segments to change appearance and move from side to side quite a lot provided that they maintain a roughly upright orientation [11]. Later, it gradually became a common feature to describe the local texture of images in the field of computer vision and pattern recognition. HOG is a description operator based on shape edge features, which can detect objects. Its basic idea is to use gradient information to reflect the edge information of image objects well and characterize the local appearance and shape of the image through the size of the local gradient.

Based on the HOG algorithm, Felzenszwalb [12] proposed the DPM algorithm in 2008. DPM algorithm adopts the improved HOG feature, SVM classifier and Sliding Windows detection idea adopts the multi-component strategy for the multi-view problem of the target and adopts the component model strategy based on the Pictorial Structure for the deformation problem of the target itself.

For the target detection and recognition algorithm of a deep neural network, in 2012, Professor Hinton [13]'s team used a convolutional neural network to design AlexNet, which defeated all teams of traditional methods on the ImageNet data set, making CNN the most important tool in the field of computer vision. Target detection algorithms based on DL can be roughly divided into three categories:

1. Algorithms based on regional recommendations, such as R-CNN, Fast R-CNN, and Fast R-CNN.
2. Detection algorithms based on target regression, such as Yolo, SSD, RetinaNet, and EfficientNet.
3. Search-based target detection and recognition algorithm, AttentionNet and reinforcement learning.
4. Anchor-free based algorithms, such as CornerNet, CenterNet, and FCOS.

III. METHOD

A. Pedestrian detection

This project mainly uses HOG+SVM for pedestrian detection.

1) HOG

HOG feature extraction is mainly composed of the following 7 steps:

1. Image gray processing
2. Image normalization

Gamma correction was used to normalize the image in color space before calculating the gradient, and the contrast of the image was adjusted to improve the detector's robustness to illumination.

Gamma formula:

$$I(x,y) = I(x,y)^{gamma} \tag{1}$$

Usually, gamma=0.5 is used for gamma correction.

3. Calculate the horizontal and vertical gradients

By calculating the directional gradient value, the contour, texture, and other information can be captured, and the influence of illumination can be further weakened.

Gradient representation under a function of one variable：

$$\frac{\partial y}{\partial x} = f(x+1) - f(x) \quad (2)$$

The gradient representation of multivariate function $F(x, y)$ is:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\delta y}{\delta x} \\ \frac{\delta f}{\delta x} \end{bmatrix} \quad (3)$$

$$\nabla f = \|\nabla f\|_2 = [Gx^2 + Gy^2]^{\frac{1}{2}} = \left[\left(\frac{\delta y}{\delta x}\right)^2 + \left(\frac{\delta f}{\delta x}\right)^2\right]^{\frac{1}{2}} \quad (4)$$

Due to the large amount of modulus calculation, the gradient calculation can be approximated by the following formula:

$$\nabla f \approx |G_x| + |G_y| \quad (5)$$

Where, G_x and G_y can be respectively calculated by the following formula：

$$G_x(x, y) = H(x+1, y) - H(x-1, y) \quad (6)$$

$$G_y(x, y) = H(x, y+1) - H(x, y-1) \quad (7)$$

G_x , G_y and $H(x, y)$ are the horizontal gradient, vertical gradient and the pixel value of the pixel at the pixel point (x, y) respectively.
The gradient value and gradient direction can be calculated by the following formula:

$$M(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \approx |G_x(x, y) + G_y(x, y)| \quad (8)$$

$$\theta(x, y) = \arctan\left(\frac{G_y(x, y)}{G_x(x, y)}\right) \quad (9)$$

Due to the limited range of the gradient direction, the unsigned range (0-180 degrees) is generally adopted, so the gradient direction can be further expressed as:

$$\theta(x, y) = \begin{cases} \theta(x, y) + \pi, & \theta(x, y) < 0 \\ \theta(x, y), & others \end{cases} \quad (10)$$

4. The image is segmented into several cells.

5. Compute the orientation gradient histogram to generate a descriptor for the cell.

The gradient direction statistics are performed in the divided cells, and the gradient direction ranges from 0 to 180 degrees. The gradient distribution is evenly divided into 9 orientation bins, each of which corresponds to a straight column.

Each pixel in the Cell adopts the weighted projection method, which is projected to the histogram channel based on a certain direction (9 orientation bins).

6. The cells are composed into several blocks and the feature vectors within the blocks are normalized.

The cells in the image are merged into several spatially connected blocks (e.g.,3*3 cells/ blocks), and the information of orientation gradient histogram generated by each cell is synthesized into a vector for representation. Each cell can appear in one or more blocks, so the computation will be evaluated multiple times, and the output of each cell can be applied to the final descriptor multiple times. A block is composed of multiple cells. Concatenated Descriptor for all cells in a block is used to obtain the HOG descriptor for the block.

Normalization of the feature vectors in the block can make the feature vector space robust to illumination, shadow and other factors. The normalization function generally adopted in human detection is L2-Norm:

$$v = \frac{v_i}{\sqrt{\sum_{i=1}^{n} {v_i}^2 + \epsilon^2}} \quad (11)$$

Where, v_i is the vector in the block, ϵ is to prevent the denominator from being equal to 0 and the ϵ value is very small.

7. Collect the HOG characteristics.

We're concatenating the HOG feature Descriptor for all the blocks inside the image to get the HOG feature Descriptor for the image.

2) The general idea of HOG + SVM:

- According to the above HOG feature extraction method, HOG feature extraction is performed on positive and negative samples (formed by the classification of the training sample set) respectively (the number of samples is m, so the number of HOG features is m*3781).
- Both positive and negative samples are assigned labels (number of labels = number of samples), and both labels and HOG features of positive and negative samples are put into the SVM classifier for training, so as to obtain the pedestrian detection model, and the results are saved as text files.
- Multiply the array support vector of one of the resulting text files by the other array alpha to produce a column vector. Then add a floating point number rho in the text file to the end of the column vector to get a new classifier, and replace the default classifier for pedestrian detection in Opencv with this classifier （cv2.HOGDescriptor.setSVMDetector()).
- SVM is a very successful supervised learning technique in terms of speed and accuracy. With SVM, multi-class data can be successfully classified by creating hyperplane. [14]SVM adopts Kernel function to map the data in the input spaces to a high-dimensional feature space. Then, in this high-dimensional space, the generalized optimal classification face is calculated.[15]
- When HOG+SVM is used for pedestrian detection, the final detection method is the linear discriminant function $Wx + B = 0$, where W represents the 3780 dimensional vector, and the one-dimensional B is the default 3781 dimensional detection operator of OpenCV(Cv2.hogdescriptor_getdefaultpeopledetector() is a 3781 dimensional detection operator).

3) Non Maximum Suppression

HOG uses a sliding window to extract features, which may lead to find many bounding boxes that represent the same people in a picture. And it is necessary to distinguish which bounding boxes are useless.

This project adopts NMS to remove redundant bounding boxes. For those with intersection, we define the *IoU* (Intersection over Union, shown by Fig.1, where the red area represents the intersection of the two images, and the green area represents the union of the two images) of the two images. If the calculated *IoU* is greater than the pre-set threshold, the smaller bounding boxes will be removed, and for those without intersection, it is directly retained as the result.

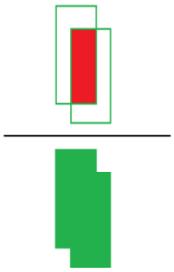

Fig. 1. Intersection over Union

The process of Non Maximum Suppression is as follows:

1. Sort by the height of the bounding box (to determine the largest bounding box).

2. Select the bounding box with the highest height to add to the final output list and delete it from the bounding box list.

3. Calculate the *IoU* of the bounding box with the highest height and other candidate boxes.

4. Delete the bounding box where the *IoU* is greater than the threshold.

5. Repeat the above process until the bounding box list is empty.

B. *Pedestrian tracking*

We store the coordinates (bounding box represented by x, y, w, h) of the detected pedestrians in each frame in the dictionary to track pedestrians (where x, y, w, h correspond to the upper left corner coordinates of the bounding box and the width and height of the bounding box).
The key is the color of each pedestrian's bounding box (because the color of each person's bounding box is independent and unique), and the value corresponding to the key is the coordinate of the center point of each pedestrian's bounding box over time.

After getting the coordinates of all pedestrians in the current frame, calculate the distance between the coordinates of the center point of the bounding box and the coordinates of all center points in the previous frame. We regard the distance between the center points less than 45 as the same pedestrian, and add the coordinates to the value of the corresponding key in the dictionary (update the dictionary).

C. *Pedestrian count*

In this project, we transfer the OpenCV built in default parameters of pedestrian detection to SVM classifier, and remove the redundant bounding box coordinates obtained by HOG+SVM through NMS. And append these coordinates into a temp list (which is initialized to an empty list every time we get a new frame). The number of pedestrians is the length of the temp list.

We record the keys of all pedestrians appearing in the current frame, store them in the list, and pass the values in the list as keys into the dictionary to obtain the corresponding values, so as to obtain the coordinates of the center points of all pedestrians in the current frame. Count the number of coordinates of the center point in the area where the mouse draws the frame on the screen.

D. *Group of walking recognition*

We set criteria for pedestrians walking together if they can satisfy two points. One is that the frames of two people are intersected (Boolean Overlap function), another is frames' size are close (between 0.75 and 1.33 times the size of the other pedestrian's box). If only one or none of these criteria can be met, walkers are judged to be on their own. Because every frame of the picture is in two dimensions, it is difficult to judge the relative position of two pedestrians only using the coordinates of the pixels. The size of frames are close means these two people are relatively standing on the same horizontal line, and frames have intersections means two people are relatively standing on the same vertical line.

E. *Group formation and destruction*

For the recognition of group formation and group destruction, we define the formation that the pedestrian walked alone in the previous frame and walked into the group with others in next frame, we use a big yellow circle to mark the people. And vice versa for the destruction if the people walked with someone in the previous frame and leave the group in the next frame, a big blue circle will appear on that people. We use a dictionary to store pedestrians' position as the key, and store 'flag' as dictionary value in the previous frame. And we use another dictionary to store the same information for the current frame. 'flag = 0' represents the pedestrian is walking alone and 'flag = 1' means this person is in a group. For the same person, from the previous frame to the current one, Thus, the value of 'flag' changes from 0 to 1 means the formation of a group and 1 to 0 means the group is destructed.

F. *Entering and leaving the scene*

In the program, we use the variable '*center_point*' represent each pedestrian's position. And we define a rectangle region which is smaller than 1920*1080. If the *center_point* of a pedestrian appears in this region for the first time, it means this person enter the scene and we use a big red circle to mark him or her in the current frame. If the *center_point* of a pedestrian is in the margin (the area between manually defined rectangle region and the edge of the image) and keeps moving far away from the frame's middle point (960,540), it means this person is going to leaving the scene and we use a small green circle to keep marking him in the current and next few frames until he disappears. We do statistics of the number of pedestrians entering and leaving the scene for each frame on the left upper corner.

IV. Experimental Result

A. Datasets

The datasets used were gathered directly from the Multiple Object Tracking Benchmark website. Specific datasets used for the results are as detailed:

STEP-ICCV21-09 are 'A pedestrian street scene filmed from a low angle', where FPS = 30, Resolution = 1920x1080, Length = 525 (00:18). Because the shooting angle is relatively low, pedestrians are in the upper part of the middle of the picture.

STEP-ICCV21-02 are 'People walking around a large square' where FPS = 30, Resolution = 1920x1080, Length = 600 (00:20). The shooting angle is relatively normal, and the pedestrians are in the middle of the picture. Because the shooting location is the big square, there are many pedestrians, and the pedestrians in the distance are dense and the pedestrian picture is relatively small.

STEP-ICCV21-07 are 'A busy pedestrian street filmed at eye level by a moving camera' where FPS = 30, Resolution = 1920x1080, Length = 500 (00:17). Because this data is captured by the camera at eye height and the photographer is moving. So, most pedestrians are always in the picture.

STEP-ICCV21-01 are 'People walking around a large square' where FPS = 30, Resolution = 1920x1080, Length = 450 (00:15). The pictures of this data set are like the shooting location of the second data set, so the characteristics of the data are basically the same. The shooting location is the big square, so there are many pedestrians.

Fig. 2. Pedestrians in different scenes

B. Setting

For this project, we use jupyter notebook as the main compilation tool. Jupyter notebook puts all resources related to software writing together. When developers open jupyter notebook, we can see the corresponding documents, charts, videos, codes and explanations. We can get all the information of the project by looking at one file. And for this project, jupyter notebook has a great advantage that it can output images and some interactive visual content in the form of web pages.

In order to achieve the project goal this time, we mainly use the python package of OpenCV, numpy and imutils with version 4.5.5, 1.23.3, 0.5.4 and the Python version number is 3.8.5. As the main function of this project, we use the official hog detector in OpenCV (cv2.HOGDescriptor) to detect pedestrians. And the whole project is developed under the windows system.

C. Metric

As a pedestrian tracking project, the main criterions are:

1. All the characters in each picture can be recognized clearly.
2. For the same pedestrian whose position in different pictures moves, each picture can recognize it and the action track is clearly visible.
3. When pedestrians are close to each other, each person can still be identified and marked as a peer (several people walking together).
4. Count the number of all pedestrians in the screen, allowing the user to manually draw a rectangular area in the video window and count the number of pedestrians in the rectangular area.
5. Monitor how many pedestrians are walking together and whether anyone has left the group.
6. Can accurately detect pedestrians entering and leaving the frame.

D. Results Performance

For each task in this project, we have shown it in the results in different ways.

Fig. 3. Performance Results (pedestrian enter or leave the screen)

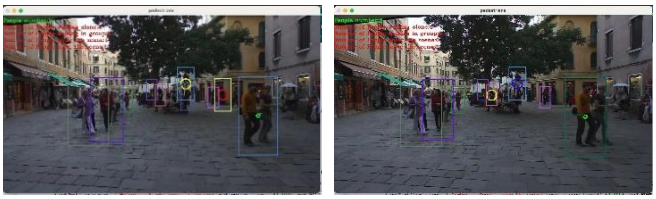

Fig. 4. Performance Results (group formation and group destruction)

As can be seen from Fig.3 and Fig.4, each detected pedestrian is circled in a bounding box with non-repetitive colors. The movement tracking is achieved by connecting the center points frame-by-frame for the same pedestrian if he keeps moving in the scene.

Different colored and sized circles are used to mark people in different states of being. The red circle is for the first time entering the scene, the green one is for the pedestrians are going to leaving the scene. The yellow one is for group formation and the blue one is for group destruction.

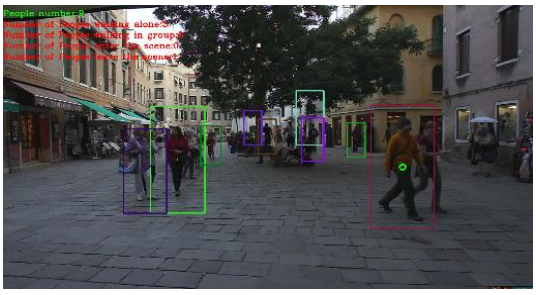

Fig. 5. The Quantity Information of Perdestrians

Meanwhile, the statistics for the number of people in the current frame, the number of people walking in group or alone, the number of pedestrians entering or leaving the scene will be displayed in the upper left corner.

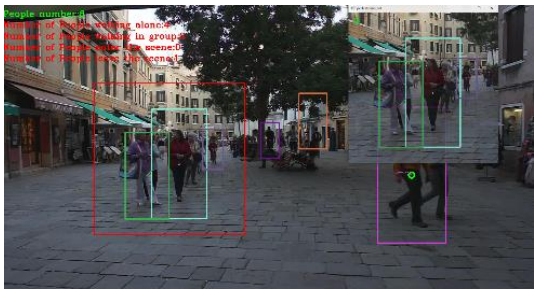

Fig. 6. Information display of perdestrians in produced window

As can be seen from Fig.6, When you click the left button on the image, you can drag a small window whose size is smaller than the original image. The number of pedestrians in the new region will be displayed on the left upper corner of the small window as well.

V. DISCUSSION

A. *error analysis*

1) Detect Pedestrians

From our point of view, failures to recognize pedestrians correctly are due to four aspects of factors.

a) Parameters in HOG

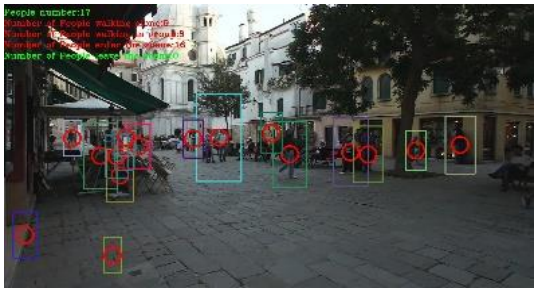

Fig. 7. Detecting result with winStride (1, 1)

Fig. 8. Detecting result with winStride (16, 16)

Fig. 9. Detecting result with winStride (4, 4)

From Fig.7 and 8, we can see that the smaller the stride of window is, the more pedestrians (and objects being mistaken for pedestrians) are. This parameter determines how many pixel the window moves in each step. Smaller window would lead to more HOG descriptors, and thus more computation. So it would take longer time to generate the result of detecting. We finally set this parameter to be (4, 4) to get best performance as we could.

Fig. 10. Detecting result with padding (1, 1)

Fig. 11. Detecting result with padding (16, 16)

As shown in Fig.10 and Fig.11, the padding does not greatly affect the performance. The reason is that it only has impacts on the detecting accuracy of people on the edge of images.

Fig. 12. Detecting result with scale = 1.05

Fig. 13. Detecting result with scale = 1.5

Fig.12 and Fig.13 clearly show how the parameter of scale affects the performance. It determines how many layers a pyramid of images would have. Smaller scale results in more layers, better accuracy and more computing time.

b) SVM Classifier

The classifier we use come from the library of OpenCV, which names *HOGDescriptor_getDefaultPeopleDetector()*. It is an array containing 3781 values. The SVM is trained on INRIA dataset in which contains images of people who are always upright, but with some partial occlusions and a wide

range of variations in pose, appearance, clothing, illumination, and background [11]. There is no people sitting in this dataset, thus the SVM classifier trained might have bias and is likely to mistake sitting people for background in our test.

c) Value of IoU in NMS

Since the NMS adopted in this project removes redundant bounding boxes, the value of *IoU* would affect the number of bounding boxes. If the value is set too large, redundant boxes would not be eliminated effectively. If the value is set too small, boxes of multiple pedestrians would be combined. After plenty of tests, we found the best value was 0.8.

d) The limitation of HOG

HOG is a traditional and relatively old algorithm for object detection. Though it had great influences on computer vision since year 2005, the drawbacks of it has also been found after a long period of time of application. The biggest cons are not being able to detect people with uncommon poses (e.g., sitting) or partly blocked. Also, it cannot cope with changes in orientation of people. Thus, the nature of HOG itself has effect on performance of our model.

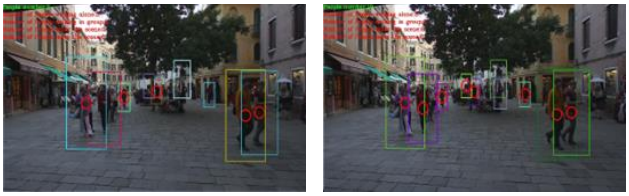

Fig. 14. Pedestrian distance threshold (low)

Fig.14 shows two consecutive frames. If the distance threshold of how long the center point has moved is too low, it is likely to cause the same pedestrian to be recognized as two people (the same pedestrian is surrounded by bounding boxes of different colors in consecutive frame images) during the movement. Please refer to the yellow and the light green box on the right of two frames.

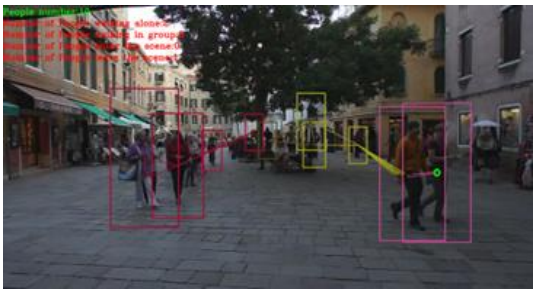

Fig. 15. Pedestrian distance threshold (high)

As shown in the figure Fig.15, if the distance threshold of the central point of the bounding box is too high, it is easy to cause multiple pedestrians to be recognized as the same person (assign the same key to different people).

After several adjustments, we believe the output result was acceptable when the distance threshold = 45.

2) Track Pedestrians

The movement tracking is achieved by connecting the center points frame-by-frame of the bounding box for the same pedestrian if he keeps moving in the scene. And the coordinates of the center point depend on the pedestrian detection model, since the model will affect the bounding box, and we get the center point from the bounding box. Inaccurate detection may cause the change of the central point of the bounding box to be inconsistent with the pedestrian trajectory, thus making the pedestrian trajectory inaccurate.

3) Count Pedestrians

Fig. 16. Statistics in current frame

We count number of pedestrians by calculate the size of a dictionary where stores info of all bounding boxes. It would not go wrong. Thus, the accuracy of this part of project mainly relies on the detecting task earlier.

4) Analyse Pedestrians

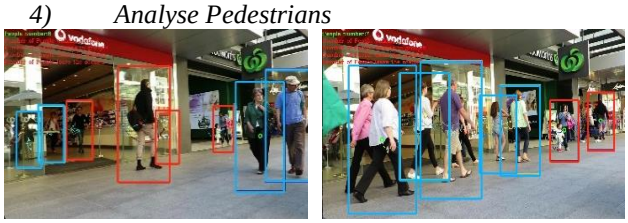

Fig. 17. Pedestrian walk alone or in groups

When detecting how many pedestrians walk in groups or alone, the testing result on our model won't have great ups and down for different scenarios. Our model can catch up people in groups or alone and calculate numbers precisely for each frame of image in most cases. However, when detecting or tracking goes wrong, this part would also be influenced. It is inevitable because parameters we set cannot be just perfect for each frame. Mistakes usually occurs when one bounding box circling multiple pedestrians because incorrect elimination of boxes performed by the NMS algorithm. But generally, our algorithm will count accurately whether a pedestrian is in groups or not, as Fig.17 shows.

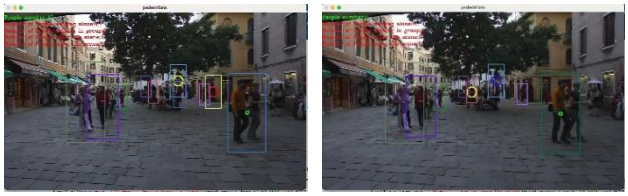

Fig. 18. Group formation and group destruction

For the recognition of group formation and group destruction, sometimes the formation and destruction will be detected correctly, but sometimes the detection won't be recognized. We suppose that the value of distance threshold is still set too small, or the distance that the pedestrian moves exceed the value of distance threshold between each frame. The same person will be falsely recognized as another one, and the formation and destruction of the group won't be marked.

Fig. 19. Pedestrian leave or enter the screen

The detection of pedestrians entering the screen is affected HOG+SVM model. If the distance between the center points of the same pedestrian of two consecutive frames is greater than the distance threshold because of the fast moving of pedestrian, it will be considered as a newly detected person, and it is easy to trigger the occurrences of pedestrians entering the screen. We use the center point of a bounding box to represent the pedestrian. If the center point is in the margin (the area between manually defined rectangle region and the edge of the image) and keeps moving far away from the frame's middle point (960,540), then we regard it as the occurrences of pedestrians leaving the screen. Because the coordinates of the pedestrian center point obtained by the HOG+SVM model are sometimes inaccurate, it will cause the movement of the center point to be inconsistent with the pedestrian trajectory. When the pedestrian moves close to the boundary of manually defined rectangle region, the pedestrian leaving the screen event is prone to false recognition.

B. pros and cons

The way how HOG extract features of edges from an image naturally have a great impact on the results. As shown in the empirical setup in Chapter 4, the results show that some non-pedestrian objects are recognized as pedestrians, which leads to the reduction of the accuracy of the results. This clearly represents the limitation it has. HOG was considered to perform well and fast in 2005, but countless improved algorithms based on it sprang up after that. Thus, HOG is not able to catch up newer algorithms in terms of efficiency and computation speed. And it has weakness recognizing people who are not upright.

However, HOG is good at obtaining information from features of edges by calculating gradients, generating distribution of gradients and use them as a histogram. Moreover, thanks to normalization, it has strong invariance to the change of contrast and illumination, let alone its own nature of being invariance to geometric and photometric transformations (except for object orientation). With classic SVM classifier, it is still capable enough to accomplish basic pedestrian detecting tasks.

VI. CONCLUSION

To sum up, this project adopts the traditional target detection algorithm to extract HOG features for the pedestrians and make recognition. The interference caused by the pedestrian state change can be overcome and the noise of initial image can be resisted to some extent, because HOG is insensitive to the changes of individual pixel values. HOG can describe the local objects' shape as the distribution of light intensity gradient or edge directions. Although, the tracking for the majority of pedestrians can be realized by HOG, some people still cannot be accurately detected such as the person in sitting, the person sheltered by obstacles, the person is too small or not clear. We do guess that the default parameters for the model 'HOGDescripter_getDefaultPeopleDetector()' are trained by the data set with features of someone who stands clearly. Or we haven't done enough to mitigate the noise before we obtain HOG descriptors for each frame, because HOG descriptors are sensitive to the noise. To enhance the detecting effect for the experiment, we need to spend more time on eliminating the noise before the HOG step. We do confirm that non-maximum suppression algorithm (NMS) has better effect than other methods such as 'Meanshift' or the removal of inside bounding boxes to obtain the optimal bounding boxes for pedestrians. We do figure the efficient ways for other tasks such as counting the walkers, collecting the movement information for pedestrians in different state of being.

Nowadays, many pedestrian detection algorithms still continue to develop based on HOG+SVM idea. By doing the experiment on the classic HOG+SVM method of the detection, we get a better understanding for the idea of HOG+SVM method and lay a foundation for the following exploration of object detection in the field of computer vision.

VII. CONTRIBUTION

Boyun Huang(z5342276):
Read reference documents. Code for Task 3. Finish the Method, part D for experimental result, Analyze Pedestrians for the discussion, and conclusion part for the report. Prepare for the demo including the PPT.
Ruijie Ge(z5277454):
Read the reference documents. Code for Task 1-2. Finish the Methods (Pedestrian tracking, Pedestrian Count), Discussion (Track and Count, Pedestrian leave or enter the scree), conclusion part in the report and prepare for the demo.
Lorac Zhu(z5277143):
Read the reference documents. Finish the Introduction, Literature review, Methods and Conclusion parts in the report and prepare the demo and PPT.
Zequan Ding(z5327869):
Read references. Code for Task 1. Analyze and discuss performance of models of all tasks and optimize parameters. Write the Discussion part of the report and prepare for the demo.
Shu Yang(z5285542):
Read the reference documents. Code for Task 1. Finish the literature review, Experimental result and part of Discussion for the report. Prepare the result part of demo and PPT

REFERENCES

[1] A.K. Jaiswal, P. Tiwari, S. Kumar, D. Gupta, A. Khanna, J.J. Rodrigues, Identifying pneumonia in chest X-rays: a deep learning approach, Measurement 145 (2019) 511–518.
[2] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (1998) 2278–2324.
[3] W. Ouyang, X. Chu, X. Wang, Multi-source deep learning for human pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2329–2336.
[4] Paul Viola, Michael Jones, Detecting pedestrians using patterns of motion and appearance, in: Proceedings of the IEEE International Conference on Computer Vision, vol. 2, 2003, pp. 734–741.
[5] Paul Viola, Michael Jones, Robust real-time face detection, International Journal of Computer Vision 57 (2) (2004) 137–154.
[6] Pang Y, Yuan Y, Li X, et al. Efficient HOG human detection[J]. Signal Processing, 2011, 91(4):773-781.
[7] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517.
[8] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," Proceedings. International Conference on Image Processing, 2002, pp. I-I, doi: 10.1109/ICIP.2002.1038171.
[9] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE, 2005, 1: 886-893.
[10] Dalal N. Finding people in images and videos[D]. Institut National Polytechnique de Grenoble-INPG, 2006.

[11] Dalal N, Triggs B. Histograms of Oriented Gradients for Human Detection[C]// IEEE Computer Society Conference on Computer Vision & Pattern Recognition. IEEE, 2005.
[12] P. Felzenszwalb, D. McAllester, Ramanan A Discriminatively Trained, Multiscale, Deformable Part Model IEEEConference on Computer Vision and Pattern Recognition (CVPR), 2008
[13] Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25(2).
[14] Aslan M F, Durdu A, Sabanci K, et al. CNN and HOG Based Comparison Study for Complete Occlusion Handling in Human Tracking[J]. Measurement, 2020, 158.
[15] Zhang, Li Hong. "Human Detection Based on SVM and Improved Histogram of Oriented Gradients." Applied Mechanics and Materials, vol. 380–384, Trans Tech Publications, Ltd., 30 Aug. 2013, pp. 3862–3865.