

# Pedestrian Detection in Small Device with Cluster based Hog

Kazuki Amakawa

November 18, 2018

## **Abstract**

**Pedstrain Detection, DBSCAN Cluster, HOG Descriptor, SVM, Small Device**

# 1 Introduction

Pedestrian detection is an essential and significant task in any intelligent video surveillance system, as it provides the fundamental information for semantic understanding of the video footages. It is special kind of problem in object detection, which have wildly applications in autonomous vehicles, surveillance camera early-warning and robot design. Normally, a pedestrian detection processing may include these steps: [2]

- 1) Pre-treatment of image
- 2) Extracting candidate regions may include human (or just include object)
- 3) Describing these region(or object)
- 4) Clasify all descriptions
- 5) Find pedestrian

In this article, we aim at pedestrian detection in a small surveillance camera and try to build a early-warning system. In this kind of program, there are several problem we have to face:

- 1) Less memory and CPU speed.
- 2) Distort image, because of fish eye camera.
- 3) Infrared model image during the night.

Most articles in pedestrian detection are care about the accuracy and speed, but they forgot a important thing, detection in night. Most data set in human detection even human re-identification just have image in daytime. So to solve this problem. We also build a new image set based on fish eye camera and infrared model in night.

In the section 2, it will showed the basic fundenmental of the method as well as the reason we don't using Hog only or DNN. It will also introduced DBSCAN, Hog, and SVM identity. In the section 3, it will introduced the development we did in Hog and DBSCAN. The section 4 is talking about IFECF dataset. The section 5 will showed the result of the method in different dataset with different model. In the end, the address of the project has been gived. It is also showed a sample model which can be used in wildly situation.

## 2 Related work

Here is the main processing of the method.

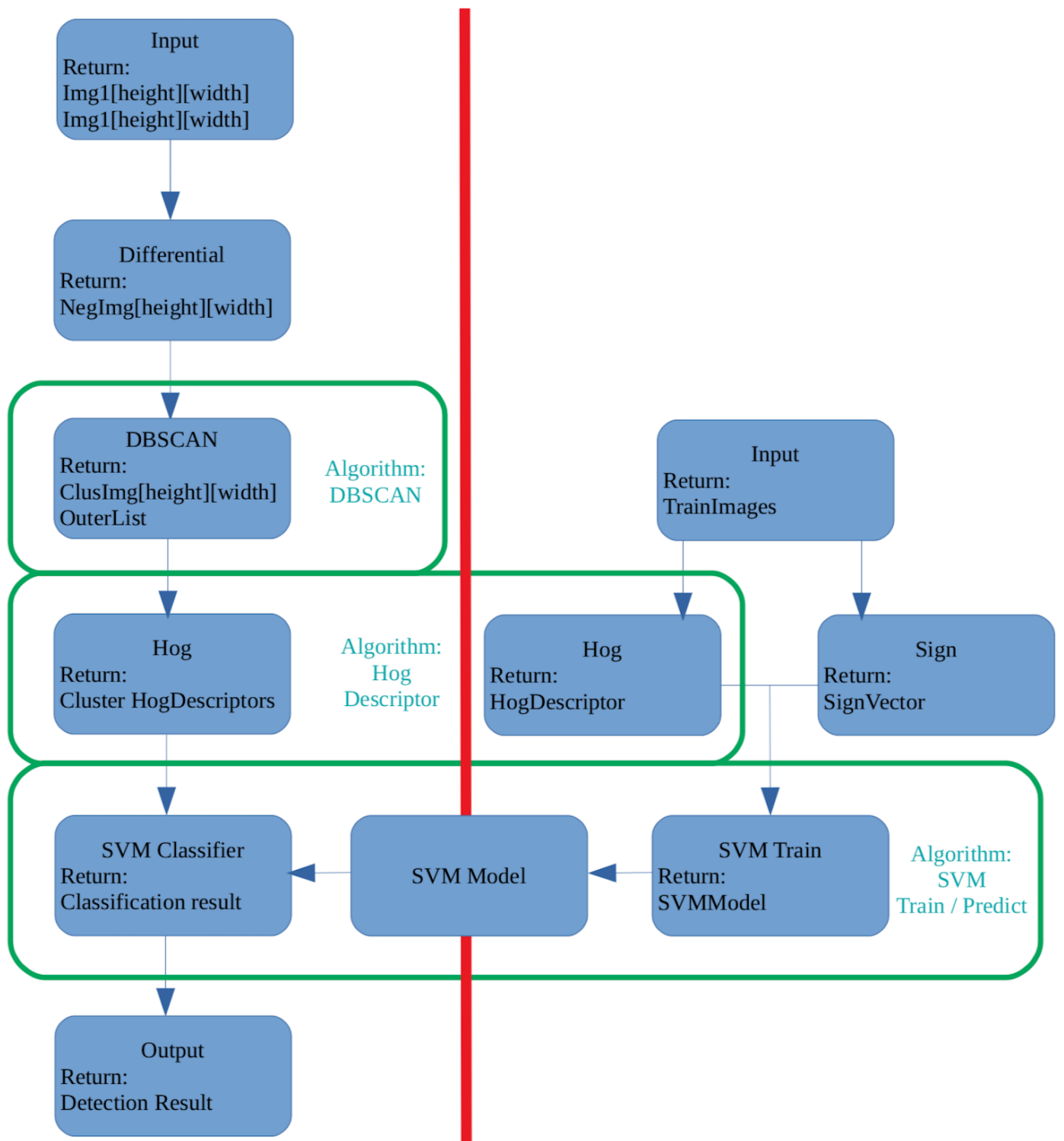


Figure 1: Fast Hog main processing

## 2.1 Pre-treatment

Different from normal human detection problem, surveillance camera have less moving and it is easy to partial background and fountground.

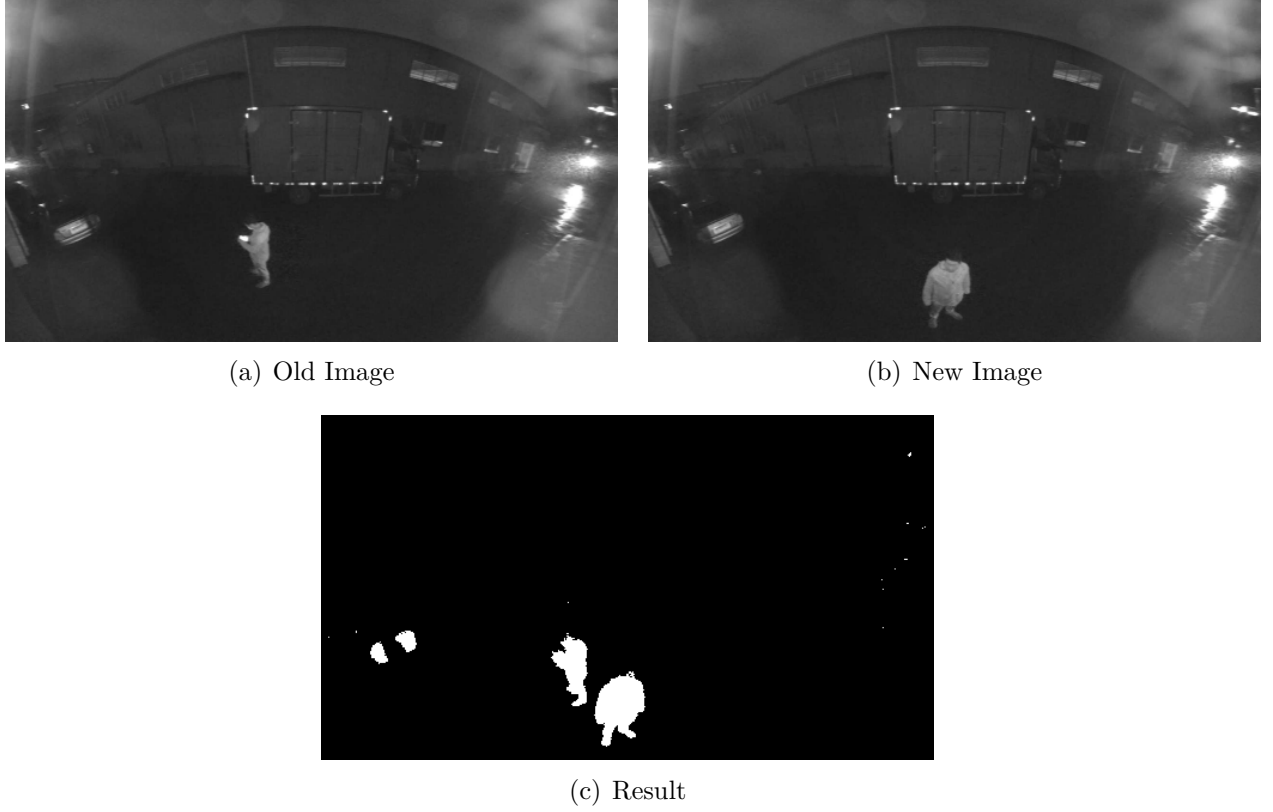


Figure 2: Differential between two images

After differential, we need to get the region may include object. And we will use DBSCAN to cluster all region connected.

## 2.2 DBSCAN method

Density-based spatial clustering of applications with noise (DBSCAN) is a density-based clustering algorithm: given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). Before the algorithm, it is necessary to give some definition

**Definition 2.1** *A point  $p$  is a inner point if at least  $minPts$  points are within distance  $\epsilon$ .*

**Definition 2.2** *A point  $q$  is directly reachable from  $p$  if point  $q$  is within distance  $\epsilon$  from point  $p$  and  $p$  must be a inner point.*

**Definition 2.3** *A point  $q$  is reachable from  $p$  if there is a path  $p_1, \dots, p_n$  with  $p_1 = p$  and  $p_n = q$ , where each  $p_{i+1}$  is directly reachable from  $p_i$ .*

**Definition 2.4** *All points not reachable from any other point are outliers.*

For example, as a data set showed in 3(a),  $A$  is a inner node and  $B, C, N$  are outer node.  $A, B, C$  are reachable from each other. And we can build a cluster with  $A, B$  and  $C$ . The node  $N$  is a noisy node will not join any cluster. The processing of cluster for all node in data set has been showed in 3(b).

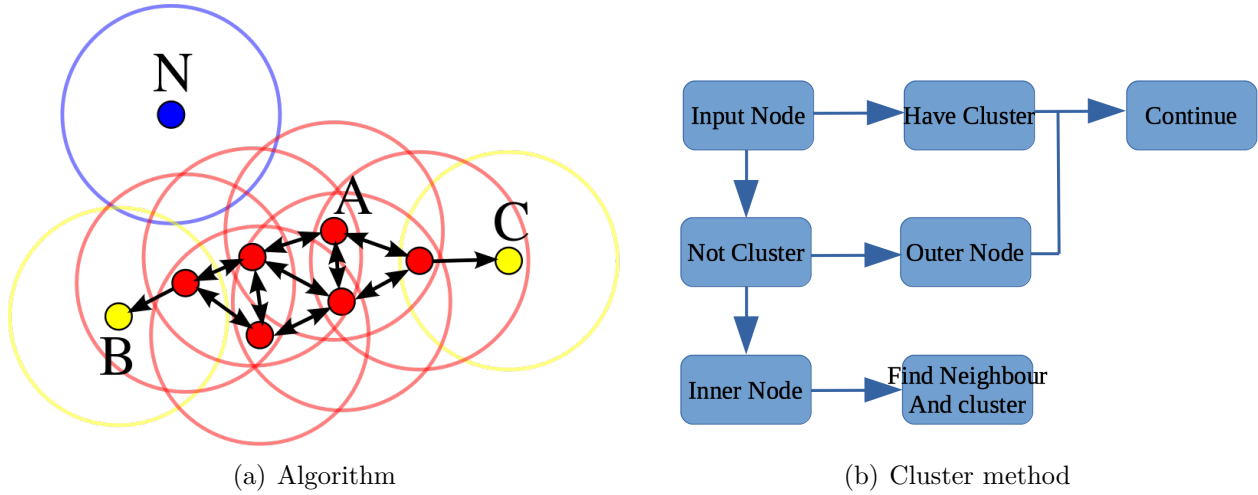


Figure 3: DBSCAN

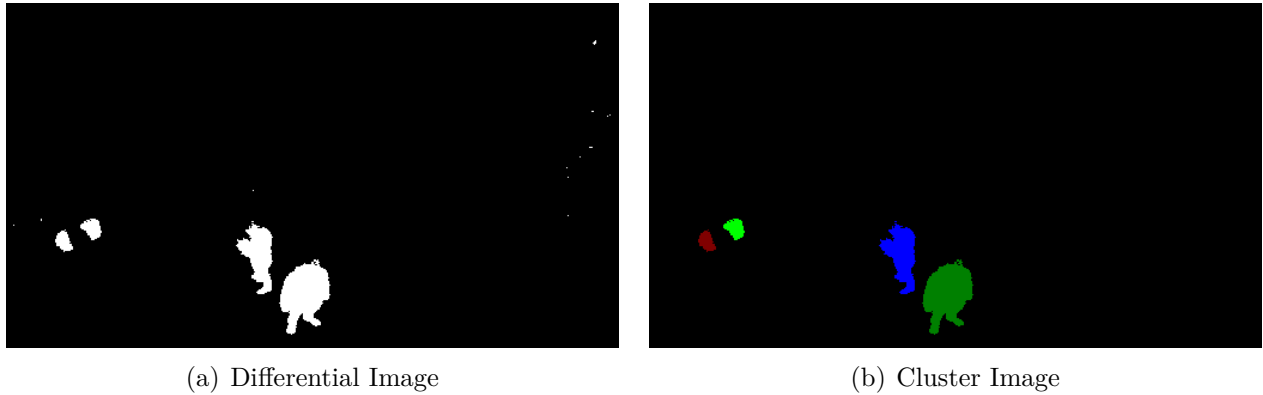


Figure 4: DBSCAN

Different from other cluster method, DBSCAN have these advantage:

- 1) Less noisy
- 2) Faster
- 3) Needn't care about the total cluster value

## 2.3 Hog Description

The main processing of Hog Description is

---

**Algorithm 1** Hog Descriptor

---

Input:  $Image[Oriheight][Oriwidth]$   
STEP1: Down Sample  $Image = \text{resize}(Image, (height, width))$   
STEP2: Gamma transform:  $Image[i][j] = \text{GammaTable}[Image[i][j]]$   
STEP3: Get the *Gradient* and *Angle* image with Sobel operator  
STEP4:  $BlockImgs = [Block\ New\ Image\ with\ BlockX, BlockY, StrideX, StrideY]$   
**for** Block in BlockImgs **do**  
     $Cells = [Get\ the\ Cell\ with\ CellX, CellY\ in\ every\ Block]$   
    **for** Cell in CellImgs **do**  
        Statistic Histogram of gradient Hisrogram[nbins]  
        Normalize the Hisrogram  
    **end for**  
    Get the *Hog descriptor in the Block*  
**end for**  
Output: *Hog descriptor of the image*

---

In STEP2 and STEP3, we have  
Gamma transform:

$$Img_{output} = A \cdot Img_{Input}^\gamma$$

Get gradient and angle:

$$SobelX = [-1, 0, 1] \quad SobelY = [-1, 0, 1]^T$$

Convolution

$$GradientX = Image * SobelX$$

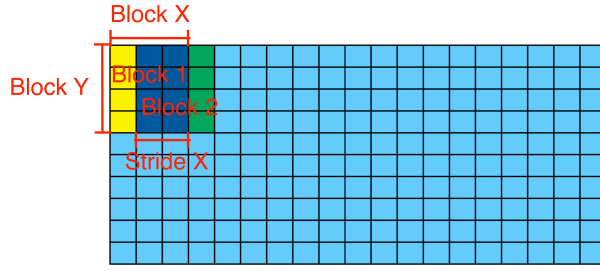
$$GradientY = Image * SobelY$$

So we have

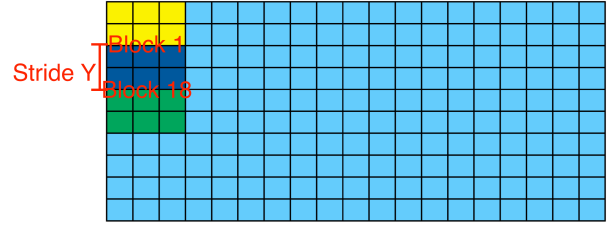
$$Gradient = \sqrt{GradientX^2 + GradientY^2}$$

$$Angle = \arctan \left[ \frac{GradientY}{GradientX} \right]$$

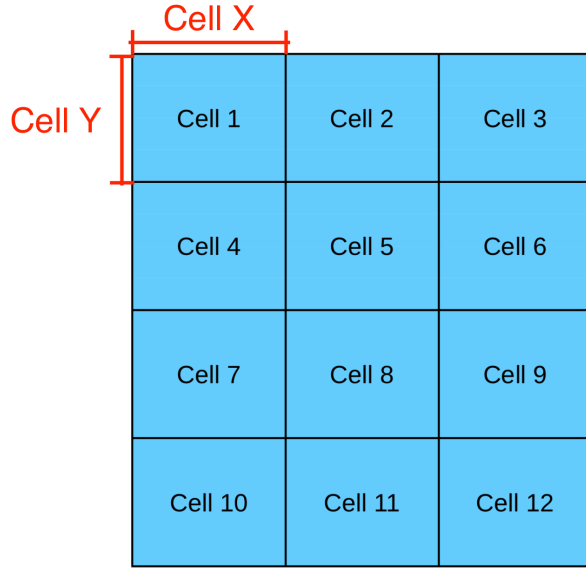
After downsample, the block of image will slide with the method of 5(a) and 5(b). Then for every block, partial it to some cell as 5(c). Finally statistic the histogram of gradiend in the way of 5(d) and 5(e). Connect all of them, then we can got the Hog feature of a block.



(a) Slide in X



(b) Slide in Y



**Block i**  
(c) Cells in a block

|    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
| 3  | 5  | 8  | 10 | 10 | 10 | 12 | 14 |
| 4  | 6  | 10 | 12 | 15 | 20 | 25 | 30 |
| 20 | 30 | 72 | 59 | 34 | 12 | 25 | 82 |
| 60 | 75 | 91 | 81 | 76 | 82 | 19 | 54 |
| 24 | 30 | 77 | 27 | 32 | 87 | 71 | 46 |
| 80 | 57 | 23 | 83 | 75 | 32 | 58 | 6  |
| 15 | 49 | 4  | 47 | 2  | 89 | 9  | 75 |
| 39 | 81 | 64 | 85 | 91 | 35 | 93 | 42 |

**Cell i**

(d) Cell and statistic

|          |        |        |        |   |       |        |     |
|----------|--------|--------|--------|---|-------|--------|-----|
| Angle    |        |        |        |   |       |        |     |
| Gradient | 3      | 8      | 10     |   | 30    | 12     |     |
|          | 5      | 10     | 10     |   |       | 25     |     |
|          | 10     | 14     | 12     |   |       |        |     |
|          |        | 4      | 20     |   |       |        |     |
|          |        | 6      |        |   |       |        |     |
|          |        | 15     |        |   |       |        |     |
| Sum      | 18     | 57     | 42     | 0 | 30    | 37     | 184 |
| Normal   | 0.0978 | 0.3098 | 0.2283 | 0 | 0.163 | 0.2011 |     |

(e) Statistic Data (Color means angle and value means gradient, statistic of white line area in 5(d))

Figure 5: Hog Block and Cells

After Hog description, we can classification feature with SVM. And finally we finished a pedestrian detection processing. On the one hand, this kind of Hog have to calculate feature of all block in the image. As most block have no human, the most source of calculation have been wasted. On the other hand, Although DNN method have better result and higher accuracy, the model of DNN will be too large for running in a small device which have just 5MB memory. So to solve this problem, we also changed DBSCAN and Hog with the characteristic of surveillance

camera image sequence.

### 3 Development of the method

#### 3.1 About DBSCAN

Typically, DBSCAN using the dataset of node. However, as image cluster just have two parameter,  $x$  and  $y$ . It is not necessary to save all vector which need cluster as a new dataset.

We can list all neighbourhood node as a table in the initial part of the algorithm. And just search the image from the table with DFS(Deep First Search). So the max time complex is  $\mathcal{O}(n \cdot m)$ , where  $n$  is total of node which need cluster and  $m$  is the size of table. It is faster than original method  $\mathcal{O}(n^2)$

#### 3.2 About Hog Descriptor

Typically, we need several times Hog network with different parameter to determine the location of person. [1] As we used the DBSCAN for object detection, we don't need the whole image description. We just need the descript for every region(Cluster) as a block. This can also make the model of SVM smaller.

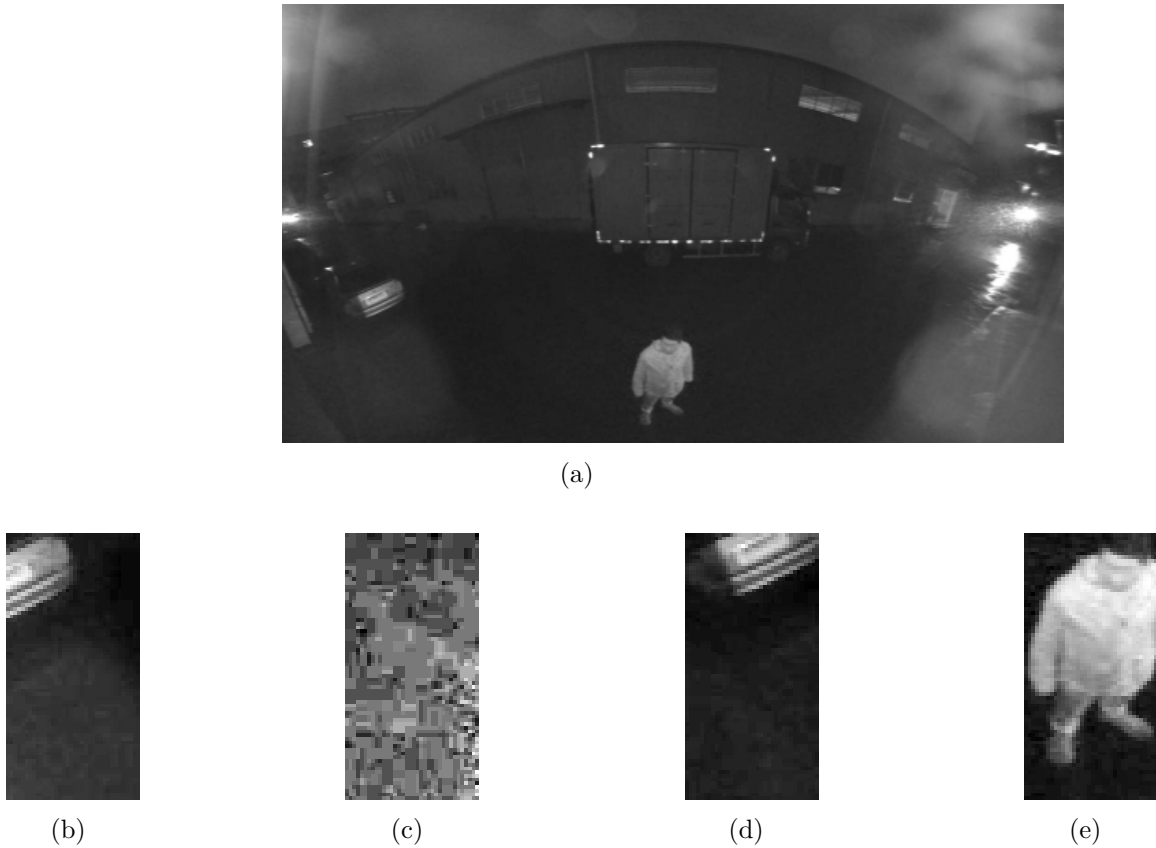


Figure 6: Image and Cluster image



Result:

a: 0.268041, 0.028966, 0.259261, ..., 0.004802, 0.104433,  $length = 144$

b: 0.436385, 0.008739, 0.054215, ..., 0.021862, 0.255745,  $length = 144$

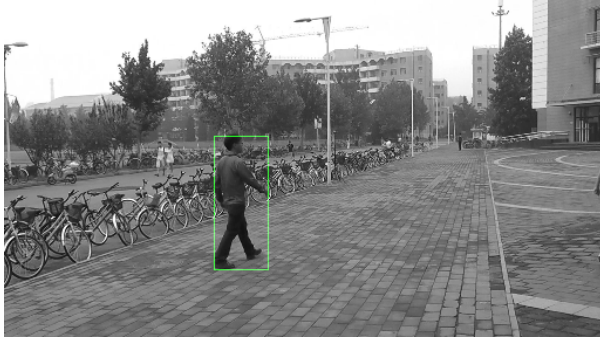
c: 0.259077, 0.057242, 0.074549, ..., 0.009185, 0.243987,  $length = 144$

d: 0.349008, 0.100296, 0.149541, ..., 0.007501, 0.184265,  $length = 144$

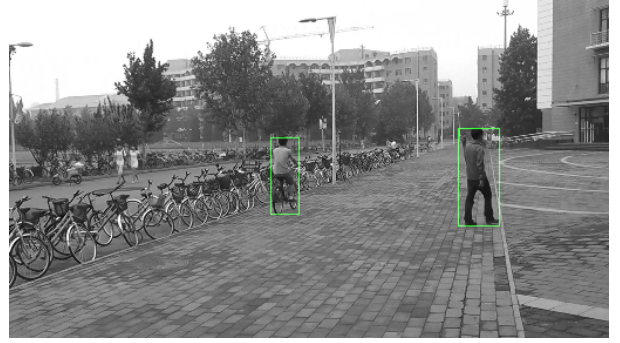
All Image:  $2.26329e^{-11}, 2.595552e^{-11}, 1.03369275e^{-11}, \dots, 2.8235041e^{-11}, 1.8345979e^{-10}$ ,  
 $length = 3888$

## 4 About IFECF Dataset(Infrared Fish Eye Camera Pedestrian Dataset)

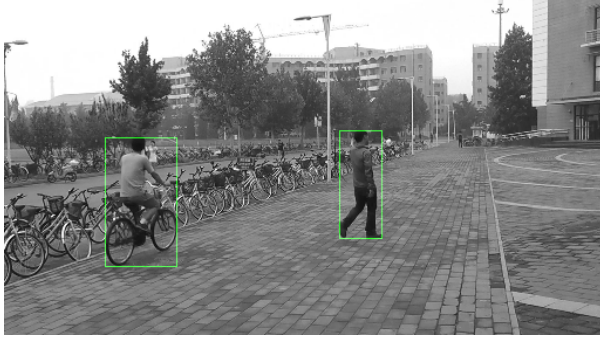
## 5 Result



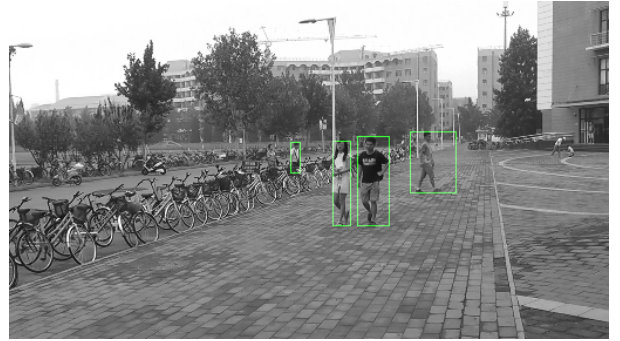
(a) Result 1



(c) Result 3



(b) Result 2



(d) Result 4

Figure 7: Result (PRW Dataset)

## 6 Acknowledge

## 7 Appendix

## References

- [1] Yuriy Lipetski and Oliver Sidla. A combined HOG and deep convolution network cascade for pedestrian detection. *Electronic Imaging*, 2017(4):11–17, jan 2017.
- [2] Duc Thanh Nguyen, Wanqing Li, and Philip O. Ogunbona. Human detection from images and videos: A survey. *Pattern Recognition*, 51:148–175, mar 2016.